UNIVERSITY OF MACEDONIA
DEPARTMENT OF APPLIED INFORMATICS
MSC IN ARTIFICIAL INTELLIGENCE AND DATA ANALYTICS

DETERMINING POSSIBLE GENDER DIFFERENCES IN UNDERGRADUATE INTRODUCTORY PROGRAMMING COURSES BY APPLYING DATA ANALYSIS TECHNIQUES IN THE RESULTS OF THE STUDENTS' ASSIGNMENTS, TO PROMOTE GENDER EQUALITY

A dissertation
by

Evangelos Dagklis

Thessaloniki, September 2023

DETERMINING POSSIBLE GENDER DIFFERENCES IN UNDERGRADUATE INTRODUCTORY PROGRAMMING COURSES BY APPLYING DATA ANALYSIS TECHNIQUES IN THE RESULTS OF THE STUDENTS’ ASSIGNMENTS, TO PROMOTE GENDER EQUALITY

Evangelos Dagklis – aid22012

Bachelor of Computer Science, University Of Macedonia, 2021

Dissertation

Submitted in partial fulfilment of the requirements for
THE MSC IN ARTIFICIAL INTELLIGENCE AND DATA ANALYTICS

Supervisor Professor
Maya Satratzemi

Approved by the three-member Examination Committee on 11/09/2023

Maya Satratzemi	Georgia Koloniari	Georgios Evangelidis
Professor	Assistant Professor	Professor

Evangelos Dagklis

....

Abstract

Over three quarters of a century have passed since women gained their right to work in a manner that permits their financial independence, yet the stereotypes regarding the kinds of professions they should prefer continue to exist. One of the domains where this observation still applies is Computer Science, a field traditionally considered to be aimed predominantly towards a male audience. But, does the real-world data provide evidence that could explain the perpetuation of this trend, or does it happen due to reasons of non-empirical nature?

The current study aims to contribute by examining the case of an introductory course on Data Structures from the second semester of the undergraduate studies program of the Department of Applied Informatics of University of Macedonia. It was conducted in the years 2021 and 2022 and the city of origin is Thessaloniki, Greece. The chosen course being one with mandatory programming assignments means that it is of very high importance and also one of a certain amount of student failure. Novice programmers are susceptible to facing difficulties with programming courses and these courses often have the highest dropout rates.

A per gender analysis of the compilation errors found in students' assignments along with their assignment grades and the final course grades is performed. Considering the numerical disparity between male and female students in the department, a better understanding of how the results compare between the two genders is interesting, on the basis of it constituting empirical evidence on how or even if a student's gender and their academic performance relate. The analysis of the collected data is done through learning analytics and more precisely through visualization and statistical analysis tests on the programming errors and grades per student. The aim is to monitor the performance per gender throughout the semester and determine possible differences that may exist. Additionally, association rule mining and clustering are applied in order to reveal potentially contributing factors for a student in passing the course, as well as whether the students can be divided into groups that share specific characteristics. The data is also used for creating models that attempt to predict whether the student will pass the course or not based solely on data from the assignment assessment files.

The findings suggest that the students' gender does not particularly change their performance characteristics, while the results of the two genders usually were deemed not significantly different. Interestingly, whenever any differences emerge, it is women that seem to be the dominant gender in terms of academic performance. Lastly, it turns out that it is possible to predict student success or failure on the course at a level that is fairly acceptable to decent by using exclusively each person’s number of errors and their assignment grade average. This can even be mostly done as early as by the first third of the semester or with higher success by its middle.

Keywords: Gender Gap, Data Structures, Learning Analytics, Machine Learning.

Foreword – Special thanks

First of all, I would like to thank Professor Mrs. Maya Satratzemi for the offer of this very interesting Dissertation subject, as well as for her cooperation, guidance, patience, quick and efficient problem solving and in general for the display of professional and human qualities in overseeing the study and everything related with it.

I would equally want to thank Assistant Professor Mrs. Georgia Koloniari, with whom I had already successfully cooperated with in the past, as well as Laboratory Professor Mr. Alexandros Karakasidis for the ideas they proposed for improving the study, for providing aid with the problem solving procedure when it was necessary, as well as for their excellent cooperation regarding the writing and correction of the abstracts and paper for the conference contribution.

Also I would like to thank Professor Mr. Georgios Evangelidis for the ideas he proposed for the improvement of the study and for aiding in problem solving whenever it was necessary.

I would like to give special thanks to my parents for the moral and financial support they provided.

I would also like to give special thanks to the Dean of the Department of Applied Informatics of University of Macedonia and former Artificial Intelligence and Data Analytics MSc Program Head Professor Mr. Ioannis Refanidis, as well as in general to the faculty of the Artificial Intelligence and Data Analytics MSc program for giving the students the opportunity to apply in practice the knowledge gained from the fields studied during their attendance of the MSc program’s courses, in a manner that prepares them for PhD studies.

In a more general note, I would like to thank the faculty of the Department of Applied Informatics of the University of Macedonia. The satisfaction from my experiences during its BSc studies program was one of the major factors that contributed in my choosing this MSc program over ones with similar subjects.

Finally I would like to thank the University of Macedonia Research Committee for providing additional financial support that covers a significant part of the expenses for presenting a paper based on work from the dissertation in the ICL2023 Conference. Some more details about this Contribution can be found in **Appendix C – Conference Contribution Paper**.

Table of contents

List of Figures

List of tables

Notations / Acronyms

Table 0-1: Acronyms and their meaning

Notation/Acronym	Full Name/Meaning
BSc	Bachelor of Sciences
MSc	Master of Sciences
PhD	Professional Degree
RQ	Research Question
Web: … (dd/mm/yy)	Website named: … visited in the indicated (day/month/year)
AI	Artificial Intelligence
ML	Machine Learning
LR	Logistic Regression
DT	Decision Tree
SVM	Support Vector Machine
NN	Neural Network
WEKA	Waikato Environment for Knowledge Analysis
LMS	Learning Management System
PCA	Principal Component Analysis
AUC	Area Under the Curve
VIF	Variance Inflation Factor
Sup.	Support
Conf.	Confidence
Cos.	Cosine Similarity
Jac.	Jaccard
C. Pass	Course Pass
TP	True Positive
FP	False Positive
FN	False Negative
TN	True Negative
LinearSVC	Linear Support Vector Classifier
MLP	Multi-Layer Perceptron

1 Introduction

1.1 Problem – Importance of the topic

Over the last three quarters of a century, numerous and significant steps were realized towards achieving true gender equality. Despite that, there is still progress to be made in order to definitively reach this goal. Regarding women’s working terms, tendencies like paying them less than male workers for the same jobs have not been completely eliminated yet, while the same also holds true for the stereotypes that dictate which working positions should be preferred by which genders (de Carvalho, C. V., Cerar, Š., Rugelj, J., et al. (2020)). For example, it is the norm to find more men than women in the decision-making management positions. The same applies for the opinion that women should favor humanitarian or theoretical studies while the more practical or mathematics-based ones should be conserved for a male audience.

The phenomenon where differences remain between the two genders favoring men, despite women being theoretically equal to them is named as the “gender gap” and it can be observed in a large variety of areas (Web: gender gap | European Institute for Gender Equality - europa.eu (04/02/2023)). One domain characterized by it is the aforementioned scientific studies, like the Information Technology (IT) sciences. Attempts to explain the gender gap in fields like programming support that women are more susceptible to facing greater anxiety, resulting in easier loss of motivation (de Carvalho, C. V., Cerar, Š., Rugelj, J., et al. (2020); Forrester, C., Schwikert, S., Foster, J,. et al. (2022)).

Then, on the subject of Computer Science, its essence can be described by the combination of Programming and Data Structures. While the courses instructing them are crucial for introducing and developing the core concepts required for understanding each different programming paradigm, as other studies have corroborated, they also tend to cause particular difficulties for students and are characterized by some of the highest failure rates among all types of university courses (Werth, L. H. (1986); Pillay, N., & Jugoo, V. R. (2005); Watson, C., & Li, F. W. (2014, June); Höök, L. J., & Eckerdal, A. (2015, April)). This can be observed by studying the advancement of their errors and academic performance alongside that of the instruction sessions.

It is true that utilization of machine learning techniques allows for the analysis of the corresponding student data and the better understanding of the situation from multiple angles. This could involve uncovering hidden useful patterns and relationships within the course's attributes affecting the students' learning process, making predictions or even revising aspects of the instruction procedure, if proven necessary.

Another noteworthy thing is that, research on the topics of Computer Science and Gender concurrently has only recently started gathering more attention (de Carvalho, C. V., Cerar, Š., Rugelj, J., et al. (2020); Forrester, C., Schwikert, S., Foster, J., et al. (2022)). Even among more recent studies regarding the usage of more advanced analysis or analytics techniques on data from programming courses, relatively few take into account the student's gender as a part of their examined factors (Kostopoulos, G., Lipitakis, A. D., Kotsiantis, S., et al. (2017); Bucos, M., & Drăgulescu, B. (2018); Forrester, C., Schwikert, S., Foster, J., et al. (2022)) and even fewer elaborate on it as their core subject (Forrester, C., Schwikert, S., Foster, J., et al. (2022)).

With the above in mind, the analysis of student results from an introductory programming course with an emphasis on per gender comparisons should be desirable on the basis of it constituting additional empirical evidence on the subject of possible differences between the two genders' programming course results, if these exist. Since the examination of this subject from the point of view of the comparison per gender is relatively new, supplementary empirical evidence could contribute towards its better comprehension. On that note though, for the purposes of this research, the genders are considered to be binary.

1.2 Aim – Objectives

This study aims to contribute by examining the introductory course on Data Structures with weekly programming assignments from the 2nd semester of the undergraduate studies program of University of Macedonia's Department of Applied Informatics, during the years 2021 and 2022. The city of origin is Thessaloniki, Greece. It should be noted that the reputation of the chosen course is that of neither being among the department's hardest ones nor its easiest. Its collected data will be used for learning analytics procedures and more precisely, for visualizations, association rule mining and clustering in order to gain a first understanding of the data, as well as for statistical tests to find out if differences exist based on a specific factor. Machine learning models will be trained afterwards, to find out up to which point it is possible to predict whether a student will pass the course or fail to do so by using only information available in their assignment submissions.

1.3 Research Questions

The aim of the study can be outlined by the following three Research Questions:

RQ1: *Do students' errors and academic performance in programming differ based on gender and if so, how?*

In order to answer the above question, the students' data will be used in two ways; for the required visualizations as well as for conducting statistical tests to determine if the two genders' population is homogenous or not. This should give a first idea of whether statistically significant differences exist between the two genders and which group is performing better, in case there is one.

RQ2: *Can the association rules and clustering aid in better understanding the relationships formed by attributes within the course?*

Using the student-level data, association rules are mined in order to better comprehend the factors that aid a person in passing the course or in not managing to achieve this goal. The data also undergoes through clustering, so as to reveal how many clusters are created, as well as whether they reflect elements of the existing student information.

RQ3: *Can a student's final grade category be predicted before the exams or, preferably even earlier in the semester by using only the number of their compilation errors and their average assignment grade?*

By utilizing the extracted information of the factors mentioned, machine learning models are trained, evaluated and tested in the process of predicting whether a student will pass the course or in not, firstly by using the data from all the available assignments per year and then only up to certain rather early ones.

1.4 Contribution

1. The data analyzed is primarily comprised of the students' errors during the compilation of their assignment exercise code, their assignment grades as well as their final course grades.
2. Some additional information is also used, regarding either a student's course status, like whether they have already been enrolled in it during previous years, or

some very general information about their current status in the department, like which is their current semester at the time of the study.

3. Information regarding other factors that were present in many other studies, like their parents’ job, the families’ incomes, whether they live in the city the university is situated or not, their high-school degrees, their presence in the instruction sessions, or even other personal information like whether they work alongside doing their studies is not present (Werth, L. H. (1986); Islam, N., Shafi Sheikh, G., Fatima, R., et al. (2019); ElGamal, A. F. (2013); Koprinska, I., Stretton, J., & Yacef, K. (2015); Okubo, F., Yamashita, T., Shimada, A., et al. (2017, March)) neither is information regarding how quickly the student writes their code or the code’s other characteristics, apart from the errors found (Pillay, N., & Jugoo, V. R. (2005); Watson, C., & Li, F. W. (2014, June); Höök, L. J., & Eckerdal, A. (2015, April)).
4. The analysis is done in a per gender manner to find out if there are any differences regarding the amount of errors per assignment, their assignment grades and their final grades. This is realized through the appropriate visualizations as well as by performing statistical tests to find whether any differences exist between the two genders and if so, where they are located.
5. As a means to further explore the reasons that contribute to the students’ success or not in passing the course as well as whether gender has any role in it, clustering is applied and association rules are mined from the extracted data.
6. A part of the collected data is also used for training Machine Learning models so as to predict whether a student will be able to pass or not the course, preferably as early in the semester as possible.

1.5 Basic terminology

The following definitions concern subjects that will be brought up in the remaining of the work and function as a basic introduction to these subjects.

Gender Gap refers to the phenomenon where, although men and women are considered equal, differences and inequalities remain between them favoring the former. It can be observed in a variety of different forms in numerous areas, like access to working positions or to education (Web: gender gap | European Institute for Gender Equality - europa.eu (04/02/2023)).

Data Analytics is a term describing the actions that can be made in order to find and extract hidden useful information contained in some data, in order to inform the next action taken to achieve something. This process usually entails multiple steps: the in-depth consideration of what kind of data is needed and what are the possible data sources, the extraction of raw data from its source, the type of mediums required to store and access the data, the conversion of the data from different sources into a more uniform shape if proven necessary, one or more steps of preprocessing so as to handle missing values or erroneous records while also bringing it to a form suitable for usage with data analysis algorithms, the application of these algorithms, the post-processing of the "raw" results, in a manner that filters the produced information deemed not very useful and/or that enhances the presentation of the final results and lastly their utilization for decision-making purposes. Analytics can differ from simple analysis in that there is consideration of what type of data will be needed, before being collected as well as the interpretation of the results is even more thorough and structured, since it will be used for improving decision-making, that the data may be used for forecasting and predictions, not just for better understanding the current situation and also in that the process may involve the building of the foundation/codebase for the analytics to be possible. In this sense, it could be said that Data Analytics is the superset the Data Analysis belongs in. This process can be applied in all different data types coming from a variety of different areas. It can be generally agreed that there are 4 types of Analytics, even though they may also be considered as different types of Analysis instead of Analytics: Descriptive, Diagnostic, Predictive and Prescriptive. Descriptive and Diagnostic Analytics both concern events that have already occurred in the past. Descriptive mainly aims to indicate what occurred in the past in a more general manner, while Diagnostic is more concerned about the reasons of the occurrence of particular past events. Similarly, Predictive and Prescriptive Analytics are more preoccupied on the subject of what can occur in the future, based on the current circumstances. As its name implies, Predictive Analytics' aim is attempting to make correct guesses about the advancement of particular situations, while Prescriptive Analytics is more preoccupied with what plans will be preferable in future circumstances (Web: Data Analysis vs. Data Analytics: Definition and Types – Indeed (04/06/2023); Web: What is Data Analytics? - Definition from WhatIs.com (03/06/2023); Web: Analysis vs. Analytics: How Are They Different? (04/06/2023); Web: Data Analysis vs.

Data Analytics: 5 Key Differences – Upwork (04/06/2023); Web: What is Data Analytics? (28/06/2023); Bruha, I., & Famili, A. (2000)).

Data Analysis refers to the specific process of Data Analytics concerning the application of specialized algorithms on defined data in order to extract useful information from it. In more detail, it encompasses the steps of: data collecting, data preprocessing and transformation, its usage for a deeper analysis through the appropriate algorithm implementations, maybe some filtering of the not so useful produced information and lastly a rudimentary presentation of the results. The consideration of the type of data required to solve the problem could be included in this procedure in a more basic form compared to Data Analytics, being more about the data sources available for using as-is at that point, not so much about the type of data source that will be required for the purpose more generally and how that can be collected, while attention regarding the building of the environment the analysis will be conducted or more in-depth consideration for the storage of the data, alongside advanced interpretation or usage of the results produced for "decision-making" are not involved. These types of factors differentiate "Data Analysis" from "Data Analytics", rendering the former a subset of the latter (Web: Data Analysis vs. Data Analytics: Definition and Types – Indeed (04/06/2023); Web: What is Data Analytics? - Definition from WhatIs.com (03/06/2023); Web: Analysis vs. Analytics: How Are They Different? (04/06/2023); Web: Data Analysis vs. Data Analytics: 5 Key Differences – Upwork (04/06/2023); Web: What is Data Analytics? (28/06/2023)).

Educational Data Mining refers to the applications of new or properly "adapted" existing methods belonging in the fields of data analysis, machine learning and statistics, in order to extract useful information from data created in educational activities and environments. Alternatively, it can also be defined as the deployment or tailoring of Data Analysis techniques on environments where educational activities take place (Şahİn, M., & Yurdugül, H. (2020); Liñán, L. C., & Pérez, Á. A. J. (2015); Romero, C., & Ventura, S. (2020)).

Learning Analytics refers to the utilization of the information gathered from Educational Data Mining and Data Analytics techniques in order to aid the decision-making process on educational matters, provide the appropriate "feedback" to the learners or improve aspects of the educational or instruction procedures and pipeline

(Şahİn, M., & Yurdugül, H. (2020); Liñán, L. C., & Pérez, Á. A. J. (2015); Romero, C., & Ventura, S. (2020)).

Artificial Intelligence (AI) is the research field on the topic of enabling technological creations to do things and perform functions that are most often associated with more human-like intelligence, like learning from their "environment" and/or their mistakes, thinking, problem solving, and expressing emotions. There are two primary considerations about how much can be achieved on a theoretical level on this field, which are the Strong AI and Weak AI. In Weak AI, it is considered that the AI agents can only emulate the human cognitive capabilities up to a certain point, while Strong AI considers that these agents should be capable of being at the same level as the human intelligence (Russel, J. S., Norvig P. & Ρεφανίδης I. (2005) – p.32; Flowers, J. C. (2019, March); Popenici, S. A., & Kerr, S. (2017)).

Machine Learning refers to the act of enabling computer systems to become proficient in successfully completing a specific task by attempting it multiple times and adapting according to the mistakes they made in each attempt so that they can improve on the next, in a manner that permits them to gradually improve in handling said task. The main types of Machine Learning are three; Supervised, Unsupervised and Semi-Supervised Learning. In Supervised Learning, the model is given examples accompanied by their corresponding labels, so as to recognize and learn the examples' patterns and properties in a manner allowing the correct guessing of said label when presented with unlabeled examples. In Unsupervised Learning, the model does not have a reference point for getting to know how to aggregate the given data, thus in order to group it accordingly, hidden relationships existing in it are found and integrated. In Semi-Supervised Learning, the model is trained with examples where only a small fraction of them are accompanied by their label, but information from the unlabeled part may also be used to improve the training results, in a manner allowing easier and improved extrapolation or generalization of the findings compared to the application of supervised learning algorithms or methods on the same data (Alzubi, J., Nayyar, A., & Kumar, A. (2018, November); Teng, X., & Gong, Y. (2018, July); Web: What Is Semi-Supervised Learning - Machine Learning Mastery (24/07/2023)).

Classification is the process of separating the given data into two or more unique "categories", rendering its type "binary" or "multiclass" respectively, based on relationship patterns recognized in each observation's attributes or characteristics and the target “category” the observation belongs in (Alzubi, J., Nayyar, A., & Kumar, A. (2018, November); Teng, X., & Gong, Y. (2018, July)).

Logistic Regression is the type of regression where the linear manner the predictive variables are expressed corresponds to the quantity of the “natural logarithm” of the ratio with numerator the probability of something happening to or for an example belonging in a class and denominator the probability of that something happening if the example does not belong in said class. This probability ratio is named as the odds ratio and if a predictive variable’s odds ratio has a value exceeding 1, said odds ratio indicates how much more likely this something is to happen to or for the example if it belongs in the class, while if a predictive variable’s odds ratio has a value less than 1, it indicates how much less likely this something is to happen to or for the example if it belongs in the class. It should be noted that, since the predictive variables correspond to the logarithm of base e of the odds ratio, the relationship between the predictive variables and the (non-logarithmic) odds ratio itself is not of linear nature (Pedro, M. O., Baker, R., Bowers, A., et al. (2013, July); Web: What is Logistic regression? – IBM (27/06/2023); Web: Logistic Regression: Understanding odds and log-odds (27/06/2023)).

Decision Tree is a term referring to a family of supervised machine learning algorithms capable of modeling through a set of if-else rules how the different values from a dataset’s variables, excluding one which functions as the target variable, can lead to said target variable’s different values. More precisely, in each step, the variables of the data, excluding the target variable, are examined and, according to which one is considered the most appropriate from the chosen measure to divide the data in the most well-defined polar opposite manner by the different values of the target variable, the desired variable’s different values if it is categorical or a splitting on them if it is continuous are used as a “condition” of “binary” or “multi-way” nature in order to lead the input examples to different paths. Each of these paths either leads to a new condition, where the above steps are repeated, or to a specific value of the target variable. The process is usually repeated until the entirety of the input data is modeled or a certain threshold condition is satisfied. These steps can be logically represented by a tree-like structure where each “condition” on the chosen variable’s values is a node and the results of the “condition” are the branches. The node the rest of the tree stems from is called the root node, while in case a node does not lead to any other branches, meaning it represents a value of the target variable, it is named as leaf node. Each example starts from the root node and carves its path by following the branches that indicate how it fares in each node's "condition", which is determined by the result of the example's corresponding variable tested in the current node-"condition" (Alzubi, J., Nayyar, A., & Kumar, A. (2018, November); Teng, X., & Gong, Y. (2018, July)).

Support Vector Machine refers to a machine learning method belonging in the supervised category which, at its most rudimentary form, represents a manner of separating two types of observations by defining a line that dichotomizes the observation space and is characterized by the maximum distance from the closest observations of both different types. In order to determine the final line or support vector, knowledge of only the observations closest to it is required. Higher feature dimensions are supported, in which case the support vector becomes a surface, while if multiple different types of observations exist, more support vectors would be required (Alzubi, J., Nayyar, A., & Kumar, A. (2018, November); Guenther, N., & Schonlau, M. (2016); Suthaharan, S., & Suthaharan, S. (2016) pp.39-40, 42).

Neural Network can be defined as a set of data processing units that are interconnected, with these connections among them having a weight uniquely associated with them. Each network has two visible layers, the input and output ones, as well as one or more hidden layers between the visible ones. Said connections are exclusively between processing units of different layers, not of units belonging in the same layer or intra-layer and inter-layer simultaneously. During the training process, the values of these weights are changing in a manner that permits the network to learn by uncovering and recognizing hidden and complex patterns in its input data that will be used during its deployment on unknown data. The more the network can output the desired result when given unknown data, the more its training is considered to be successful (Haykin, S. & E. Γκαγκάτσιου (2010) – p.2; Alzubi, J., Nayyar, A., & Kumar, A. (2018, November); Baradwaj, B. K., & Pal, S. (2012); Teng, X., & Gong, Y. (2018, July)).

Clustering is an unsupervised learning method referring to the process of grouping observations or records of a dataset together, forming teams, groups or sets of items that are considered similar. This data similarity is a combination of the data's inherent characteristics as well as their interpretation by the chosen algorithm and similarity metric. Unlike in the case of classification, whose algorithms belong in the supervised or at least in the semi-supervised learning category, clustering is unsupervised as the "ground-truth" of the groups the data should be arranged in is either not known or is not used in the procedure (Teng, X., & Gong, Y. (2018, July); Alzubi, J., Nayyar, A., & Kumar, A. (2018, November))

Association Rules are a data mining technique capable of revealing hidden patterns and relationships existing in the data, by finding the frequent co-occurrences that happen to characterize some attributes of its records (Damaševičius, R. (2010); Ayub, M., Toba, H., Yong, S., et al. (2017); Matetic, M., Bakaric, M. B., & Sisovic, S. (2015, June)).

1.6 Structure of the study

The 2nd chapter reviews the goals and findings of other studies that have broadly similar subjects to the current one. The 3rd chapter discusses multiple points regarding the realization of the study. In greater detail, it is used to acquaint the readers with the data used, its extraction process and the tests it underwent through, in order to better comprehend its characteristics. It also describes the process of its usage for the data analysis procedures and presents the produced results. A review and short discussion on the findings, as well as an outline of the limitations of the study, alongside potential future additions are provided in the 4th chapter.

2 Literature review – Theoretical background

While the relationship between a person's gender and its role in passing an introductory programming course is usually not on its own the main subject of most published studies, the attempt to predict whether the student will pass one such course and the finding of the factors that either aid them in passing the course or prevent them from doing so has been an important topic for decades (Werth, L. H. (1986)). From the available methods for conducting this analysis, the usage of machine learning algorithms for making predictions and the extraction of association rules are the most common, although other methods like questionnaires are also seen.

Beginning with the usage of questionnaires, this method has been used in combination with other student data in (Pillay, N., & Jugoo, V. R. (2005); Höök, L. J., & Eckerdal, A. (2015, April)) and (Islam, N., Shafi Sheikh, G., Fatima, R., et al. (2019)).

From the cited three studies that used questionnaires (Pillay, N., & Jugoo, V. R. (2005); Höök, L. J., & Eckerdal, A. (2015, April); Islam, N., Shafi Sheikh, G., Fatima, R., et al. (2019)) the first one (Pillay, N., & Jugoo, V. R. (2005)) was more about the factors that "influenced" the students' performance in an introductory programming course, while the other two (Höök, L. J., & Eckerdal, A. (2015, April); Islam, N., Shafi Sheikh, G., Fatima, R., et al. (2019)) were more about the difficulties the students faced during said course. In (Pillay, N., & Jugoo, V. R. (2005)) the data collected concerned the students' personal characteristics that could potentially be associated with their grades and more precisely their maternal language, existing capability in using computers, learning style, aptitude in problem-solving and their gender, in (Höök, L. J., & Eckerdal, A. (2015, April)) it was about their study habits and in (Islam, N., Shafi Sheikh, G., Fatima, R., et al. (2019)) it was about the parts of the course they found the most difficult. In (Pillay, N., & Jugoo, V. R. (2005)) the researchers concluded that the maternal language and the person's own problem solving abilities were the most influential factors, while the difference between male and female students was not statistically significant. In (Höök, L. J., & Eckerdal, A. (2015, April)) they found that the students who failed did not actively engage with the course for long enough, although they did so passively for significantly more when compared to the students who passed, alongside that solo work for the course was more proficient than of small teams and in (Islam, N., Shafi Sheikh, G., Fatima, R., et al. (2019)), by applying association rules and clustering on the answers, they found specific parts that were considered difficult, as well as the mediums the students seem to more easily learn from.

The attempt to predict the students' grades, preferably early enough in the semester seems the most common practice with the usage of data mining techniques on information from introductory programming courses (ElGamal, A. F. (2013); Mueen, A., Zafar, B., & Manzoor, U. (2016); Bucos, M., & Drăgulescu, B. (2018); Koprinska, I., Stretton, J., & Yacef, K. (2015); Kostopoulos, G., Lipitakis, A. D., Kotsiantis, S., et al. (2017); Okubo, F., Yamashita, T., Shimada, A., et al. (2017, March); Figueiredo, J., Lopes, N., & García-Peñalvo, F. J. (2019, October)).

In (ElGamal, A. F. (2013)) the data collected was from the Learning Management System (LMS), as well as some additional data describing the student's gender, mathematical performance and existing programming capacity. In (Mueen, A., Zafar, B., & Manzoor, U. (2016)), the collected data was similar, but the students' additional data was much more detailed, despite not involving high school grades or other experiences, like in (ElGamal, A. F. (2013)). In (Kostopoulos, G., Lipitakis, A. D., Kotsiantis, S., et al. (2017)), the data used was similar to the previous two studies, but the students' demographic characteristics concentrated on their marital, family and professional status, indicating an older target audience. Unlike the previous two similar studies and especially the second one, in which there was a possibility the studied courses might have demanded, at least to a certain degree, physical presence, this one concerns a bachelor course instructed remotely, though four optional "contact sessions" exist. Moreover, the active learning paradigm was used, along with a gradual deployment of the data. The data used in (Bucos, M., & Drăgulescu, B. (2018)), was again similar, but unlike some of the previous studies, the timeframes that characterized the student's activities were not included. It is noted that the data was collected exclusively from traditional learning environments, although it is similar to the ones from the previous studies, being course report data in conjunction with some additional student information.

While in most cases, the data collected involved additional participants' information, in some others the researchers refrained from using any form of demographic or social characteristics. In (Koprinska, I., Stretton, J., & Yacef, K. (2015)) the data collected concerns only the students' usage of the course's e-resources, their forum activity and their grades, like in the studies mentioned in the previous paragraph

(ElGamal, A. F. (2013); Mueen, A., Zafar, B., & Manzoor, U. (2016); Bucos, M., & Drăgulescu, B. (2018)) but no other external data. In (Okubo, F., Yamashita, T., Shimada, A., et al. (2017, March)) the data was also very similar to (Koprinska, I., Stretton, J., & Yacef, K. (2015)), although instead of discussion activity, it involved the presence in the instruction sessions. Also unlike that case, here the data is handled like a time-series, when multiple weeks of training data are involved.

There is also the case of (Figueiredo, J., Lopes, N., & García-Peñalvo, F. J. (2019, October), where the students' data was created during the instruction sessions, by the "monitoring and evaluation" of their "activities", resulting in somewhat more abstract indicators like "curiosity", "perfectionism" and the ability to successfully utilize in practice the do-while concept. These indicators seem to characterize much more the person's learning habits and attitude during the course, instead of just the time spent with its activities. Additional data includes the students' participation and their performance in specific activities. Others types of interaction with the course's learning materials are not monitored or used.

From the studies mentioned above, information regarding the students' gender was available in (ElGamal, A. F. (2013); Kostopoulos, G., Lipitakis, A. D., Kotsiantis, S., et al. (2017)) and (Bucos, M., & Drăgulescu, B. (2018)) but it was used as a predictive factor in just the first two cases.

Beginning with the studies' results, in (ElGamal, A. F. (2013)), the Decision Tree C4.5 algorithm was used to produce if-then rules that would result in the student passing the course with different levels of ease. The aim is to indicate the usefulness of this method in the examined situation. It is emphasized that in the created rules, the left hand side part would always result in a specific grade category as the right hand side with "100%" certainty. Factors like the "High School Mathematics grade" and "Programming Aptitude" were found to indeed be useful.

The (Koprinska, I., Stretton, J., & Yacef, K. (2015)) also used decision trees exclusively, but this time the goal was the prediction of the students' course grade category preferably early enough, not as much the creation of student profiles, like in (ElGamal, A. F. (2013)). This was managed with 72.69% accuracy with the data of all 13 weeks, or with 66.52% accuracy with the data up to the 7th week.

In (Mueen, A., Zafar, B., & Manzoor, U. (2016)) Decision Trees were also used for the prediction of the students' grades, like in (Koprinska, I., Stretton, J., & Yacef, K.

(2015)), but in conjunction with Multilayer Perceptron and Naïve Bayes. The results showed that Naïve Bayes had the best results of the three, achieving an accuracy of 86% with all the available data features, or 85.7% with the 6 best ones.

In (Kostopoulos, G., Lipitakis, A. D., Kotsiantis, S., et al. (2017)), algorithms of the same categories with (Mueen, A., Zafar, B., & Manzoor, U. (2016)) were used, and more precisely the J48 and JRip Decision Trees, the Naïve Bayes, the Multilayer Perceptron as well as Logistic Regression and Sequential Minimal Optimization from WEKA in order to predict only whether the student would fail or pass the course, not the category their grade would belong to, like in (ElGamal, A. F. (2013)) and (Koprinska, I., Stretton, J., & Yacef, K. (2015)). The Sequential Minimal Optimization method had the best result, with its accuracy being 64.61% by using only the pre-university student data, 75.54% with the data up to around the "middle" of the examined period and 80.82% before the exams.

In (Okubo, F., Yamashita, T., Shimada, A., et al. (2017, March)), the aim was once again the prediction of the students' grades, like in (Koprinska, I., Stretton, J., & Yacef, K. (2015)) and (Mueen, A., Zafar, B., & Manzoor, U. (2016)), but this time with the usage of a Long Short Term Memory Neural Network, a Recursive, thus more advanced and specialized Neural Network variant than the non-Recursive Multilayer Perceptron, used in (Mueen, A., Zafar, B., & Manzoor, U. (2016)) and (Kostopoulos, G., Lipitakis, A. D., Kotsiantis, S., et al. (2017)), capable of "remembering" long term relationships (Hochreiter, S., & Schmidhuber, J. (1997)). As a point of reference, they also used Multiple Regression Analysis, a method not seen in the previously cited studies (ElGamal, A. F. (2013); Mueen, A., Zafar, B., & Manzoor, U. (2016); Koprinska, I., Stretton, J., & Yacef, K. (2015); Kostopoulos, G., Lipitakis, A. D., Kotsiantis, S., et al. (2017)). The Neural Network achieved significantly better performance and by the 6th week from a total of 15, it had 93% accuracy.

In (Bucos, M., & Drăgulescu, B. (2018)), the aim is only the prediction of whether a student would pass or fail, like in (Kostopoulos, G., Lipitakis, A. D., Kotsiantis, S., et al. (2017)), not a more detailed grade like in (ElGamal, A. F. (2013); Koprinska, I., Stretton, J., & Yacef, K. (2015)) or (Okubo, F., Yamashita, T., Shimada, A., & Ogata, H. (2017, March)). The available demographic data, like the person's gender was not used. The methods used were Python Scikit Learn's Classification And Regression Tree, Extra Trees Classifier, Random Forest Classifier, Logistic Regression

Classifier and C-Support Vector Classification. The Random Forest, Logistic Regressions and Support Vector Classifiers managed the overall best results, according to the authors, with an accuracy of 86% in determining the students that pass the course, with data up to week 8 from 12.

In (Figueiredo, J., Lopes, N., & García-Peñalvo, F. J. (2019, October)) the aim is to determine whether the students pass the course or not, like in (Kostopoulos, G., Lipitakis, A. D., Kotsiantis, S., et al. (2017)) and (Bucos, M., & Drăgulescu, B. (2018)), but with a greater emphasis on finding the ones who fail. The only method they used was the Neural Network, like in (Okubo, F., Yamashita, T., Shimada, A., et al. (2017, March)) where it was the main prediction method, as well as in (Mueen, A., Zafar, B., & Manzoor, U. (2016)) and (Kostopoulos, G., Lipitakis, A. D., Kotsiantis, S., et al. (2017)) where it was among the methods used. The chosen model is a Multiple Back-Propagation Neural Network with an achieved accuracy over 94% and an F1 score over 96.5%, while the students it misclassified were only 3 from 85 and from which only 1 was considered successful when they were actually not.

As it can be seen, most of the studies that attempted to predict whether a person will successfully pass an introductory programming course used a considerable amount of data describing the students' situation in the course, while even in these cases, the student's gender was rarely a factor.

Another interesting analysis is that of (Alzahrani, N., Vahid, F., Edgcomb, A. D., et al. (2018, June)), which uses the students' coding data, like the time and attempts needed, from a total of 80 students in 78 short questions in a C++ introductory course from spring of 2017. With this data, a struggle metric was proposed and the activities that caused the most struggle as well as the most common compilation errors in them were determined. The aim is to indicate which course parts can cause the highest struggle, so that the instructors would make the necessary changes to reduce it. As a more basic method to deal with this, the authors briefly describe the implementation of a hint system. The struggle results before and after this change were compared, indicating a somewhat reduced struggle level after the method's application.

Clustering is another method that has been used in order to analyze student data from courses belonging in the IT or CS departments.

In (Cobo, G., García-Solórzano, D., Santamaría, E., et al. (2010, June)) the forum data of 55 "online" students in a "virtual" "Electronic Circuits Theory" course was used for agglomerative hierarchical clustering. The authors followed what they refer to as a “five-step strategy” where firstly, it is accepted that the main activities a student can do in a forum is to either read a post or create one and then, that a sequence of these activities can be logically conceived as a time series. The data concerned 119 days and the actions performed in them were 14142 reads and 369 posts. In (Bian, H. (2011, April)), the “SimpleKMeans” and Subspace Clustering methods are applied in a very different kind of dataset to that of (Cobo, G., García-Solórzano, D., Santamaría, E., et al. (2010, June)), which is the grades of 16 “activities” divided in 12 “in-class labs”, 2 larger projects and 2 exams, as well as the final course grade from 30 students in a “Computer Science Service” course. The study of (Lopez, M. I., Luna, J. M., Romero, C., et al. (2012)), also used forum data, like (Cobo, G., García-Solórzano, D., Santamaría, E., et al. (2010, June)), but both the course and the kind of data used are different. While (Cobo, G., García-Solórzano, D., Santamaría, E., et al. (2010, June)) used only read and post actions represented through time series, (Lopez, M. I., Luna, J. M., Romero, C., et al. (2012)) uses indicators describing the characteristics of each student’s discussion environment usage, like the “number of messages” they sent or the “number of sentences” they wrote, alongside the two network analysis measures “Centrality” and “Prestige”. Like (Bian, H. (2011, April)), the students’ final grade is also available. (Lopez, M. I., Luna, J. M., Romero, C., et al. (2012)) does not exclusively use clustering, like the previous two works mentioned, but also uses classification. The clustering algorithms are WEKA’s: “EM”, “FarthestFirst”, “HierarchicalClusterer”, “sIB”, “SimpleKMeans” and XMeans” and the classification ones are WEKA’s: “DTNB”, “JRip”, “NNge” and “Ridor”, “ADTree”, “J48”, “LADTree”, “RandomForest”, “Logistic Regression”, “Multi-layer Perceptron”, “RBFNetwork”, “SMO”, “BayesNet” and “NaiveBayesSimple”.

Elaborating on the goals of each study, (Cobo, G., García-Solórzano, D., Santamaría, E., et al. (2010, June)) aims to indicate a new methodology for examining and better comprehending student activity in online “discussion” environments, as well as dividing said students into subsets based on the examined forum factors, (Bian, H. (2011, April)) aims to use clustering and especially subspace clustering in order to determine different useful student subsets, like for example to identify those that are in danger of failing the course early enough in the examined period, while (Lopez, M. I., Luna, J. M., Romero, C., et al. (2012)) investigates if it is possible to predict a student’s final course grade with data exclusively from their online discussion activity, alongside if it is possible to achieve classification of successful and unsuccessful students through clustering. The study's second goal is similar to (Bian, H. (2011, April)), although without an additional focus on student failure. It should be noted that the final course grades in both (Bian, H. (2011, April)) and (Lopez, M. I., Luna, J. M., Romero, C., et al. (2012)) are binary. From the results of (Cobo, G., García-Solórzano, D., Santamaría, E., et al. (2010, June)), it is indicated that the desired clustering is feasible, although the "natural" clusters will have to be extracted by separating with "inconsistency thresholds" and not with the more usual "distance". In (Bian, H. (2011, April)), the clustering methods deployed, including the more usual algorithms, especially "SimpleKMeans", used firstly for students and then, by transposing the dataset, for the activities, as well as the Subspace Clustering revealed that some activities function well for dividing the students according to whether they pass the course, while others were definitely not as they resided on the two extremes of the difficulty spectrum. The "turning point" for a person's performance was indicated to be activity 6 "loops" and then the activities 8 and 9, corresponding to "Classes and Objects", while 10 and 7 are also mentioned. Finally, (Lopez, M. I., Luna, J. M., Romero, C., et al. (2012)) indicated that it is feasible to both use only forum-based data to predict the students' final course grade and that, clustering algorithms can offer the same levels of accuracy as the more usual classification algorithms, which is what happened with EM, especially so after testing with only the better dividing features of the dataset.

Another method for analyzing data coming from introductory programming courses, used by the researchers, is the application of association rules.

In (Damaševičius, R. (2010)) the association rule extraction was utilized for determining the relationship between a student's failure in at least some of the "object oriented programming" course's labs, exercises and theory exam with the failure in passing the course, entailing the achievement of a grade less than 5. The most interesting rule ended up being the failure in the lab regarding the Linear Dynamic List subject leading to failure in passing the course which dictates the need to restructure the corresponding instruction session, concerning the Dynamic Lists.

In (Matetic, M., Bakaric, M. B., & Sisovic, S. (2015, June)), the method was used on the log data from a blended course, the second introductory programming one named "Programming 2". The data involves the points and grades the students got in the assessments, like in (Damaševičius, R. (2010)), but also their active involvement with the available resources. It turned out that, on the one hand, the students who failed were uninvolved with both the self-assessments and the lecture discussions and that on the other hand, engagement with video lectures was positively associated with passing the course. It is concluded that some changes to the course will be beneficial, like the creation of video lectures for more units.

In (Ayub, M., Toba, H., Yong, S., et al. (2017)), the association rule mining was used on the activity log file and grade data from the students of two undergraduate courses “Introductory Programming” and “Algorithm and Data Structures”, with the aim of determining in greater detail the characteristics of the students’ engagement with the course and of improving the used Learning Management System. The data used was quite similar to that of (Matetic, M., Bakaric, M. B., & Sisovic, S. (2015, June)) although, unlike it and (Damaševičius, R. (2010)), the data concerns two courses where the first is a “prerequisite” for the second. The findings suggest that students’ educative material access time and their level of engagement with the Learning Management System are strongly correlated with their final grade, while as a means to make using the LMS more desirable, the authors consider the addition of gamification elements.

In (Caton, S., Russell, S., & Becker, B. A. (2022, February)), the association rules method, in conjunction with the markov chain analysis, were used on the log data from Java exercises derived from 4 introductory programming courses during 2 years. The data is comprised of the program’s output to the student’s code, from which the compilation errors are extracted, as well as the person’s email whose hash works as a session identifier. The markov chain method is used to represent the compilation errors and the system’s response to the student code, describing the completeness or correctness of the said submission, creating a model from which the association rules are extracted. The aim is to find students’ patterns of commonly repeated errors so as to provide suggestions for the adequate management of the situation by the instructors. Their conclusion is that specific coding creation process patterns observed exist, which can result in hard to comprehend compilation errors that the teaching materials and the instructors should warn the students about.

The above constituted the main comparison of the Literature Review’s works. Two tables summarizing the results in chronological order can be found in **Appendix A – Dissertation Literature Review Tables**.

3 Methodology

In this part, the details of the empirical study conducted in this dissertation are explained. This includes in-depth descriptions for the experiments' data characteristics along with everything related to the analysis methods followed and the procedure for making predictions.

3.1 Course Outline

The course the data comes from is "Data Structures", which is taught in the second semester according to the department's undergraduate studies program. Its subjects are: 1) Sets, 2) Stacks, 3) Queues, 4) Lists – Dynamic Stacks & Queues, 5) Binary Search Trees and 6) Hashing. As homework students are asked to solve weekly programming assignments. Each of them usually consists of two to three small to medium size programming exercises, using the C programming language.

3.2 Participants

The participants in the study are the students that attended the course in the years 2021 and 2022. In the first year, 34% of the attendants were of female gender and 46% of the total students had already been enrolled in it in a previous year, while in the second year, 31% of the students were women and 19% of the total attendants had been enrolled in it before. It should be also noted that, in 2021 the course was conducted remotely due to the COVID-19 pandemic, while in 2022 it was in person. Additionally, the total amount of unique students found in all the examined data is 1327. But, since 4 students, 1 from 2021 and 3 from 2022, do not appear in either year's supplementary data files, they were omitted, making the examined students 1323. Their lack of such data could be a result of not enrolling in the course in the allotted timeframe despite them having started attending it. From these 1323 unique student, 924 were from 2021 and 693 from 2022, though overlapping students exist between the two years.

3.3 Data Files and Analysis

The data used for the analysis done in this work comes from two sources: the output files of the automatic assessment tool of the students' exercise code, available for each exercise as well as spreadsheet files with additional student data describing their status in the course and in the department available for each year.

3.3.1 Experiment's Hardware and Software

It should be noted that every part of the empirical analysis was conducted using locally installed Python on a pc with a 6-core/12-thread Intel CPU running at ~4Ghz, 16GB of RAM and an Nvidia GPU featuring 2048 cuda cores with 8GB of Video RAM. Regarding the software, the entirety of the experiments were conducted using Python programming language and the Spyder IDE, as they constitute software that does not require a financial fee to be used, they are quite well suited for the purposes of data analytics as well as, since in 6 out of 8 courses (or out of 7 if only the courses with programming assignments are counted) in the MSc's syllabus, this was the preferred programming language to use.

3.3.2 Exercise Assessment Files

These files contain the assessment of the solutions from the programming exercises each of the students submitted as a part of the weekly assignment they are asked to solve. They are created by the program named "Diorthotis", whose creation and demonstration is the work of Laboratory Professor Mr. Karakasidis (Karakasidis, A. (2023)). Each such assignment is usually comprised of two or three small to medium size programming exercises, and whenever this is the case, the multiple submitted code files of each person are examined serially.

Their typical file form is:

```
student_id,total_assignment_grade,current_exercise_name

=== COMPILATION ===
compilation_status_message
compilation_errors_notes_and_warnings

=== EXECUTION. TEST #XY ===
code_execution_result

=== EXECUTION RESULT: ===
code_execution_result_status_message
GRADE:current_exercise_grade
```

For each programming exercise of an assignment submitted by a student, the data written are: a comma separated triplet with the student's id, the grade they achieved in the assignment and the name of the exercise being examined, the compilation field with the code's compilation status and the errors that occurred alongside where they occurred, if there were any, one or more execution test fields where the code is being tested with input data chosen by the instructors and finally an execution result field with the result of the code's testing and the grade the student achieved in this question.

There are some details that need to be known though in order for the parsing to be realized correctly:

- The number of empty lines separating the different parts described is not completely fixed. There can be more than one such lines or none.
- In the first data triplet seen, which can be considered as the id of each assessed exercise code, instead of a comma (,) a vertical line (|) may have its place.
- If an assignment is comprised of more than one exercises, which is what usually happens, in each consecutive exercise code file after the first one submitted by a student as a part of the examined assignment, the triplet "student_id,total_assignment_grade,current_exercise_name" becomes "</pre>,current_exercise_name". Its first two parts are replaced with a "</pre>" label, signifying that they are the same with the latest mentioned.
- The assessments concern only the exercises of an assignment submitted by a student, not the entirety of the assignment's exercises the student is asked to solve. So, there is no explicit mentioned information about whether a student has submitted all of the exercises contained in an assignment or only some of them. This has to be inferred by searching through the students' submissions in the corresponding assessment files.
- While there are no duplicate students in an assignment assessment file on its own, if two of these files exist for one assignment, in case, for example one of them is for the students who worked with teams and the other for those working solo, duplicate students can exist.
- An exercise of an assignment can be considered a copy if the "FULL_COPY" label is found. In this case, even if an exercise has been graded, its actual grade is considered to be zero.

- If during the runtime of an exercise code, an infinite loop occurs, it is possible for the entire execution result field to be omitted, in which case the exercise is again graded with zero.
- Finally, if an exercise was not solved with the usage of the appropriate function of the required library, after its introductory triplet a message is shown indicating the function's incorrect usage and the exercise is graded with zero.

The knowledge of these small details is required in order for the parsing of these files to be met with success.

3.3.3 Student Information Files

These are spreadsheet files containing supplementary information for each year's students that are enrolled in the course. This information shows a few more details regarding the student's situation in the course as well as in the department in general. The names of the files' data columns are: student id, name and surname, father's name, status ("active", "deleted", "achieved bachelor", "paused the studies"), department entry year, course enrollment semester, first enrollment in the course, grade (meaning the course's final overall grade), number of absences during the instruction sessions, class, state of application (is always "normal"), email and gender. The column-features used in the analysis are: first enrollment, enrollment semester, final overall grade and gender. The student id is also used but only for gathering the student's data and not as an element to be analyzed. The number of absences could have been used but since they are not counted for this course, they are exclusively "0". Though, it should be noted that, these features indicate the contents of the files in their complete, initial form. Before being locally stored, they were anonymized so as to avoid having access to personal information of the course's students.

In order to begin the data analysis procedure, the useful data from the assignment evaluation files is extracted through the usage of a custom parser. On the field of the compilation imperfections, the errors, warnings and notes contained are also parsed, through regular expression patterns. For the remainder of the study, all the types of compilation imperfections will be called simply as compilation errors. An error is considered to be made each time a pattern is activated, as many times as it happens in each exercise in a person's assignment. The number of unique error categories recognized was initially 109, but since a substantial number of these were not considered frequent enough, only 46 error categories remained, which are the ones used in the rest of the study. An error category is considered frequent if it appears at least 12 times among all the studied assignment assessment files from all years. The plots showing said recognized errors, their frequencies and the percentages they were encountered by each gender are shown in **Appendix B – Errors and their Frequencies**. From this procedure, the extracted information concerns:

- the errors found and more specifically the categories they belong in, how many times they were made and by whom,
- the students' separate exercise grades as well as
- their assignment grades.

It should be noted that the amount of assignments done in each year is not the same; there are 12 assignments in 2021 and 11 in 2022, and neither are the exercises they contain despite coming from the same assignment sheets. In more detail, the original assignments of the two years are the following:

These are the contents of 2021's assignments.

Table 3-1: Assignment details from 2021

Assignment Number	**Exercises**	**Subject**
1	a1f1, a5f1	Sets
2	a5f2, a16f2	Stack
3	a8f2, a17f2	Stack
4	a6f3, a12f3	Queue
5	a1f4, a30f4	List, Dynamic Stack, Queue
6	a2cf4, a2jf4, a2rf4	List, Dynamic Stack, Queue
7	a9f4, a10f4, a16f4	List, Dynamic Stack, Queue
8	a11f5, a29f5	Binary Search Tree
9	a25f5, a26f5	Binary Search Tree
10	a30f5	Binary Search Tree
11	a7f6	Hashing
12	a4f6	Hashing

And these are the contents of 2022's assignments.

Table 3-2: Assignment details from 2022

Assignment Number	**Exercises**	**Subject**

1	a2f1, a6f1	Sets
2	a7f2, a18f2	Stack
3	a1f3, a13f3	Queue
4	a23f4, a25f4	List, Dynamic Stack, Queue
5	a31f4, a32f4	List, Dynamic Stack, Queue
6	a2if4, a2gf4, a2qf4	List, Dynamic Stack, Queue
7	a11f4, a15f4	List, Dynamic Stack, Queue
8	a5f5, a31f5	Binary Search Tree
9	a8f5, a32f5	Binary Search Tree
10	a5f6	Hashing
11	a10f6	Hashing

The subjects of both years are the same but differences in the number of assignments and the exercises they contain exist. In order to more easily compare the results between the two years, it is considered that the completed assignments in each of them are just 10, something achieved by merging the data from specific consecutive assignments that come from the same original assignment sheet. Below, the new assignments and their contents are shown.

Table 3-3: The 10 new assignments for both years and the process of their creation

New Assignment Number	**2021's Assignment Numbers**	**2021's Contained Exercises**	**2022's Assignment Numbers**	**2022's Contained Exercises**	**Assignment Subjects**
1	01	a1f1, a5f1	01	a2f1, a6f1	Sets
2	02, 03	a5f2, a16f2, a8f2, a17f2	02	a7f2, a18f2	Stack
3	04	a6f3, a12f3	03	a1f3, a13f3	Queue
4	05	a1f4, a30f4	04, 05	a23f4, a25f4, a31f4, a32f4	List, Dynamic Stack, Queue
5	06	a2cf4, a2jf4, a2rf4	06	a2if4, a2gf4, a2qf4	List, Dynamic Stack, Queue
6	07	a9f4, a10f4, a16f4	07	a11f4, a15f4	List, Dynamic Stack, Queue
7	08	a11f5, a29f5	08	a5f5, a31f5	Binary Search Tree
8	09, 10	a25f5, a26f5, a30f5	09	a8f5, a32f5	Binary Search Tree

9	11	a7f6	10	a5f6	Hashing
10	12	a4f6	11	a10f6	Hashing

This data, along with the students' overall course grades, is used for the creation of the following plots.

3.4 Visualizations

This section demonstrates potentially interesting visualizations and how the results they indicate can be explained so as to better understand the characteristics of the used data.

3.4.1 Errors per Student

Figure 1 concerns the percentage of errors made by students from each of the two genders from the total amount of errors found in each of the assignments during the two examined years.

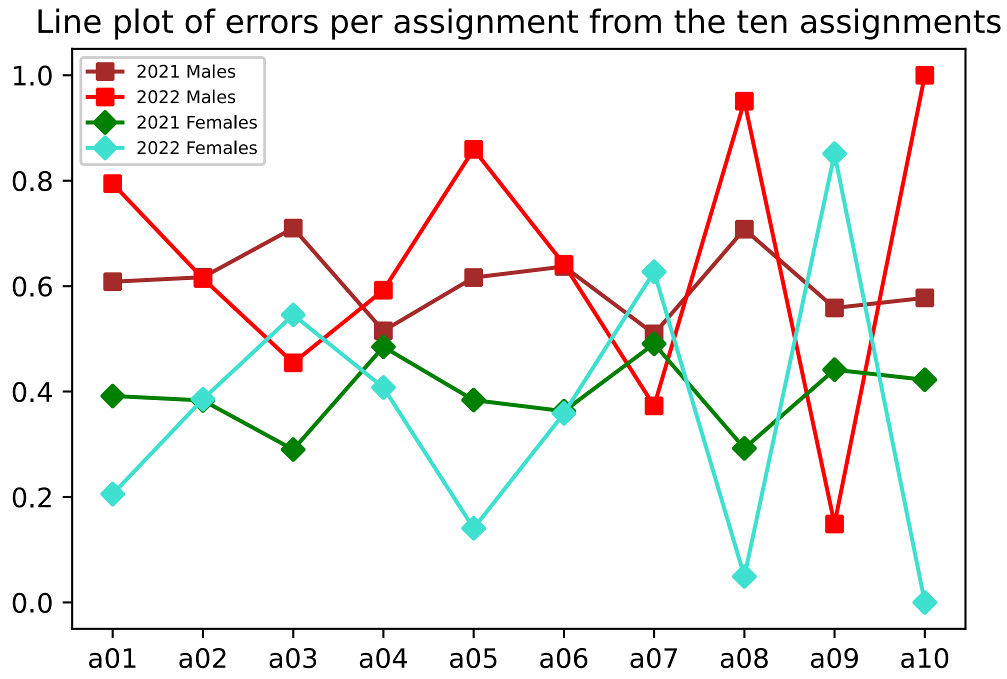


Figure 3-1: Line plot of the percentage of errors done by the two genders in each assignment from the two years

At first, it seems that the male students are much more prone to making errors than the female ones, as their percentages are always higher in the year 2021 and almost always higher in 2022, excluding the assignments 3, 7 and 9. It also seems that the percentages of the two genders are much closer in 2021, where the largest difference between them is around 40% in the 3rd exercise, with the 8th being similar enough, than in 2022, where at least five exercises, the 1st, 5th, 8th, 9th and 10th, have differences of at least 60%. This either suggests that 2021’s two genders’ students are closer in terms of performance than those of 2022 or that the student profiles between the two years are different.

In 2021, the course was conducted remotely, due to the pandemic, while in 2022, it was in person. By examining the students’ extra data, it turns out that in 2021, around 46% of students had already been enrolled in the course at least one time before, while in 2022, this percentage was only 19%. This means that in 2021, many older students ceased the opportunity to enroll in the course and pass it, while in 2022 their number probably seems to be more in line with what used to happen in the years before the pandemic.

But a very important thing to be noted is the fact that the male students are significantly more numerous than the female ones each year, who only constitute one third of the course’s active students. Due to the imbalance between the classes, it is not possible to safely make any assumptions about the amount of errors done by the two genders just from this plot.

3.4.2 Weighted Errors per Student

Figure 2 depicts the errors per student by gender weighted by the amount of exercises submitted by students of the examined gender as a part of each examined assignment.

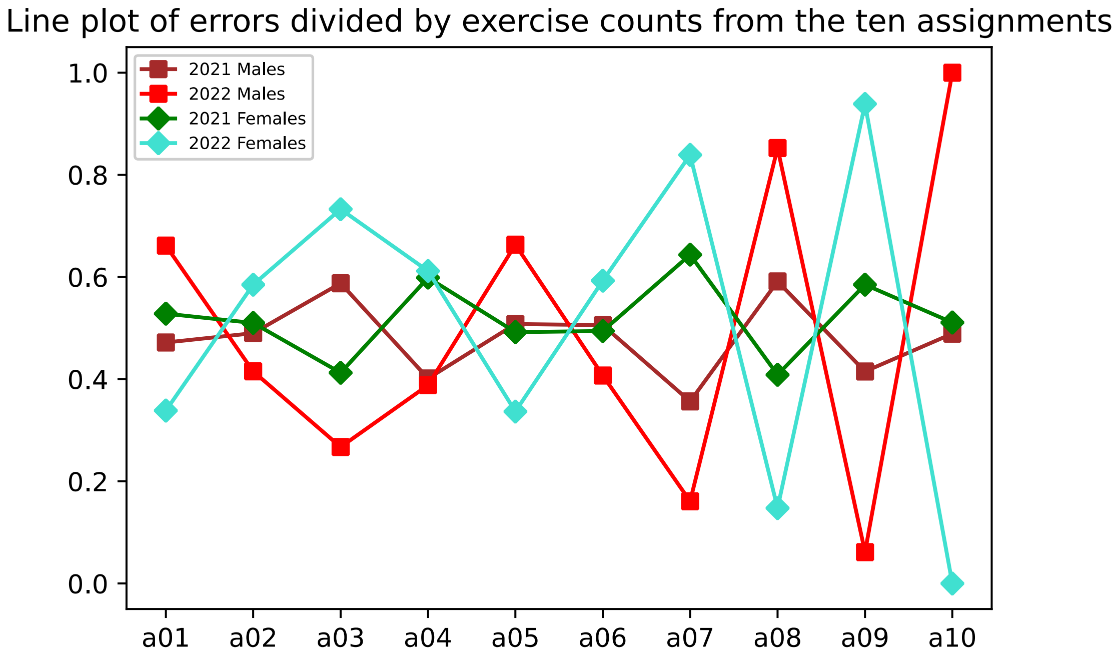


Figure 3-2: Line plot of the errors by gender divided by the exercises contained in each assignment and submitted by students of that gender that have submitted each examined assignment

The weighting of the errors per student from each gender's students was deemed necessary in order to minimize the effect of the imbalanced nature of the two genders' classes. By doing so, though, the perception about the percentages of errors per gender changes drastically, as now it seems that female students are just as likely to make errors as the male students, if not even more so. This can be inferred since in 2021 and 2022, more errors were found by women in the 6 out of 10 assignments, the 1st, 2nd, 4th, 7th, 9th and 10th in the first case, the 2nd, 3rd, 4th, 6th, 7th and 9th in the second case.

Regarding the differences per gender in both years, they seem to be lesser in 2021 than in 2022 just like the previous plot, and while the largest difference in 2021 remains around 40%, it is found in the year's 7th assignment not its 3rd as before. For 2022, just like the first plot, the 8th, 9th and 10th assignments have a difference of at least 60% between the genders, but the 1st and 5th assignments do not. On the contrary, the 7th is now characterized by such a difference, while the difference between genders in the 3rd assignment is also quite close to the aforementioned 60% threshold. The dominance of relatively smaller differences in 2021 and of more extreme ones in 2022 indicates that students of the former year have closer error rates between genders than those of the latter.

3.4.3 Copied Exercises per Assignment

Figure 3 shows the percentages of the submitted exercises that were deemed as copies from each year's ten assignments.

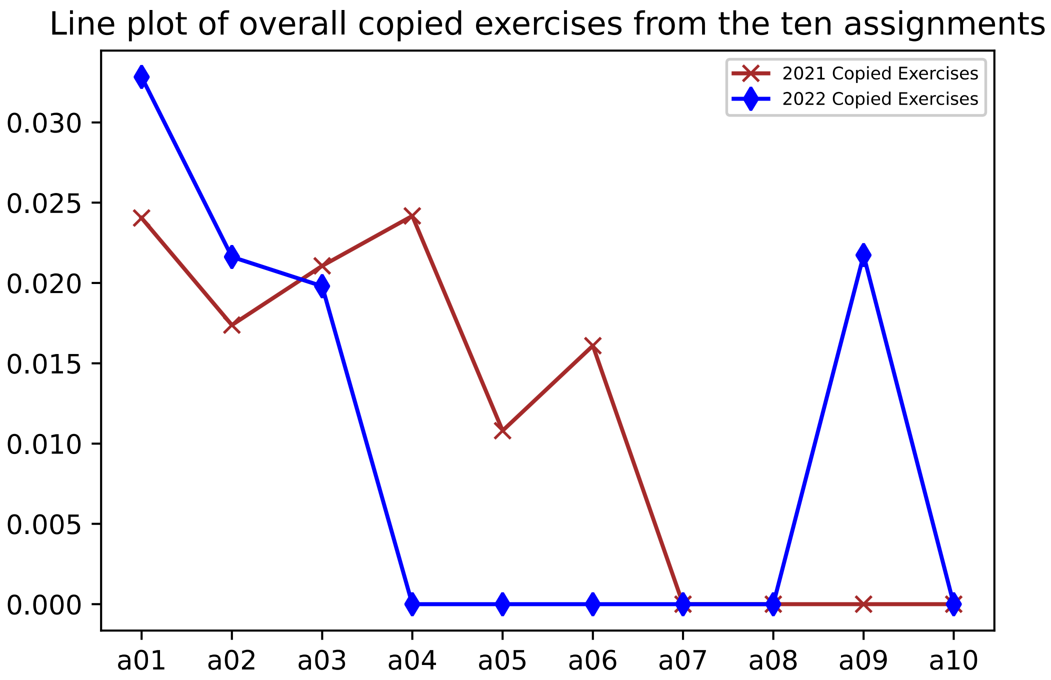


Figure 3-3: Line plot of the percentages of exercises considered to be copied in each of the two year's assignments

It is encouraging that these percentages appear to be quite low, never surpassing the 3.5% threshold in any year's assignments. It seems that there were more assignments in 2021 where copied exercises were found, with these being the first 6 of them, while in 2022, copied exercises were only found in the year's first 3 assignments as well as in its 9th one. The more usual variance of percentages for the copied exercises is higher in 2021, with it ranging from 1% to 2.5% than in 2022, where it usually is around 2% to 2.4% except for that year's 1st assignment, where this percentage rises to almost 3.5%. This means that the actual variance of these percentages, at least when they are non-zero, is 1.5% in both years, 1% to 2.5% in 2021 and 2% to 3.5% in 2022.

3.4.4 Copied Exercises per Gender per Assignment

Figure 4 depicts the percentages of copied exercises from each of the two years' ten assignments made by each gender.

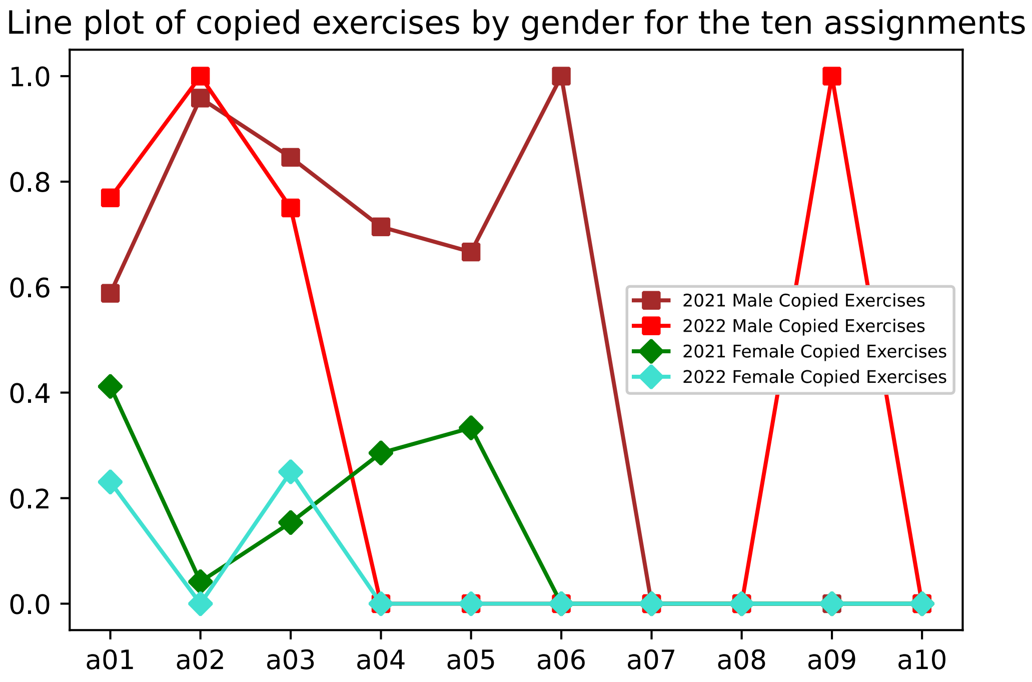


Figure 3-4: Line plot of percentages of copied exercises by gender in each of the two years' assignments

In all the assignments, the percentages of copied exercises are higher for the male students than for the female ones, something expected as women constitute only a third of the students' total population. The least difference between the two genders is found in the 1st assignment of 2021, being around 16% (42% from women, 58% from men), while in all the other cases, the differences are higher than 20% (at least 40% women and 60% men). Especially in 2022, the differences in all the cases are at least 40%, with 30% of the copies at most being attributed to women and 70% to men. Something else that can be noted is that women do not only tend to have lower percentages of copied exercises in comparison to men, there are fewer cases where they have cheated in general, as they do not contribute in the percentages of the 6th assignment of 2021 or the assignments 2 and 9 of 2022. With the above data, women students seemingly have less of a tendency to cheat in their exercises in comparison to their male colleagues.

3.4.5 Per Gender Average Exercise Grades

The next Figure, the 5th, is of the per gender average assignment grades from each of the ten exercises of the two years.

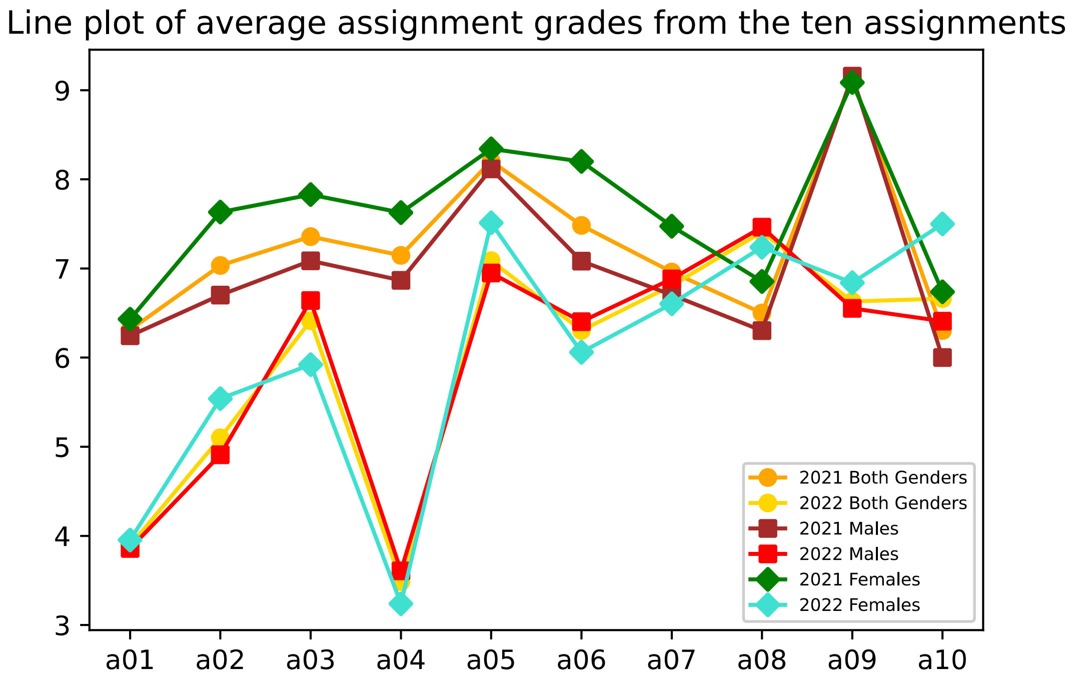


Figure 3-5: Line plot of per gender average grades the students achieved in each of the ten assignments

Starting with the per gender comparison, it can be seen that in 2021, the averages of the female students surpass the male ones in all the assignments but the 9th, while in 2022, they only surpass the male students in half of them, in the 1st, 2nd, 5th, 9th and 10th. Since in both years the male candidates are significantly more numerous than the female ones, the averages from all the students are closer to the averages of the male students. The range of grades is smaller in 2021 than in 2022, as in the first case it is between around 6 and 9, making it 3 grades, while in the second case it is from a little bit over 3 till around 7.5, making it almost 4.5 grades. Interestingly, in all the assignments excluding the 7th, 8th and 10th, the lowest 2021 per gender average grade is visibly higher than the highest 2022 per gender average. The overall trend formed indicates that 2021's students tend to achieve higher performance than those of 2022, as well as that, women are higher achievers than men in 2021 even though the two gender's performance is much closer in 2022.

3.4.6 Per Gender Final Course Grades

This 6th Figure represents the average final overall course grades per gender as seen from the students that submitted each of the ten assignments.

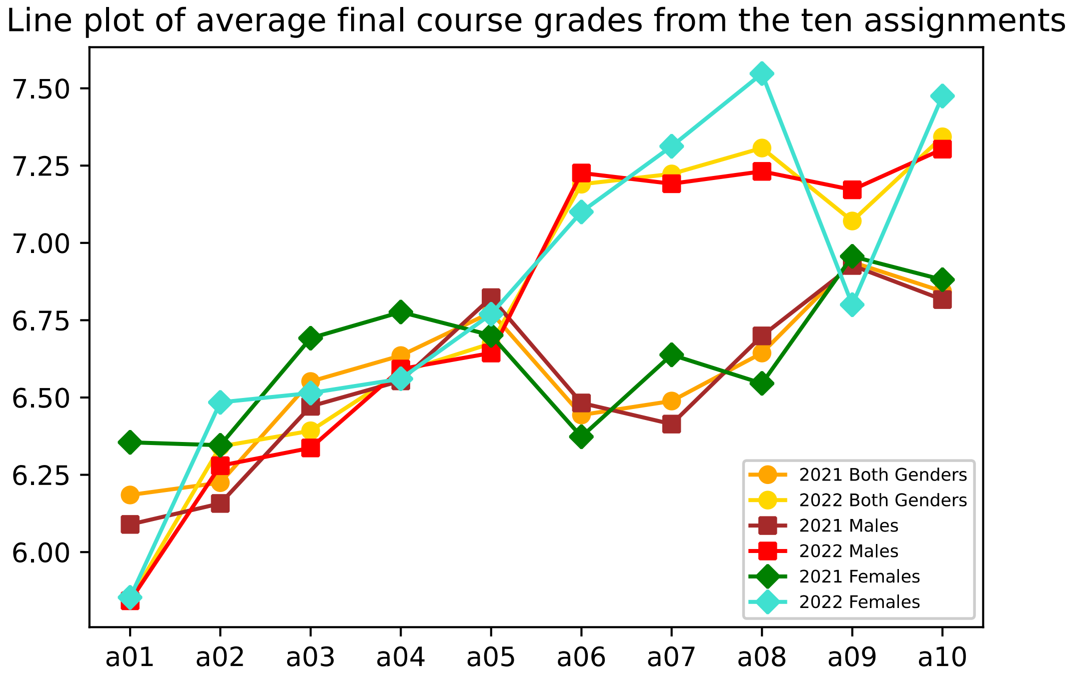


Figure 3-6: Line plot of per gender average final course grades the students that submitted each of the ten assignments achieved

Here, the average grades for the female students remain higher than their male counterparts' for seven assignments per year, with the only exceptions being the 5th, 6th and 8th ones from 2021 and the 4th, 6th and 9th of 2022. The range of grades in 2021 is once again narrower than in 2022, with it being from 6 to 7, less than 1 grade in the first case and from around 5.6 to 7.6, around 2 grades in the second case. Unlike the average assignment grades, though, here there is a clear difference in the grades per year only for the 1st, 6th, 7th, 8th and 10th assignments, from which only in the 1st the grades are unequivocally better in 2021. It seems that, here, the highest averages were achieved by the 2022 students who submitted the assignments from the sixth and onwards, with the only exception to the rule being the women who submitted the 9th assignment of 2022. Still, regarding the performance per gender, women seem to retain an overall higher level of achievement in both years.

The fact that 2021's students managed better assignment average grades but 2022's better final grades may show that in 2021, there were comparatively more students who may have already had the knowledge required to solve the assignment exercises from having participated but abandoned the course in previous years or from attending other similar courses in the department or, alternatively, that they were either only concerned with passing the course, not with achieving a high grade, or lacked the time to study properly for the final exams, due to participating in too many courses.

3.5 Statistical Tests

The aim of the conducted tests is to investigate if any statistically significant differences emerge between the two genders and if so in which cases these are found. These tests were conducted for:

- the number of errors per person,
- the exercise grades of each person as well as
- their final grades.

The data used is at the level of:

- each of the years' assignments, so for 12 in 2021 and 11 in 2022,
- all the available data per year, so for 2021 separately and for 2022 separately and
- all the available data, so for the data of the two years together.

The examined confidence levels are 90%, 95% and 99%. The tests able to determine the said differences between genders are two: the t-test for the similarity of the population's means and the Mann-Whitney population homogeneity test. As the first test requires the data from both of the desired populations to follow normal distribution, in case this does not happen, the second one will be applied.

3.5.1 Test Descriptions

Before applying the tests, a small description of each one is necessary so as to understand their goal.

Kolmogorov-Smirnov Test: A test that determines if a population follows a specific distribution. In this case, the desired distribution is the normal one and it is used if the number of observations is over 50. Its two hypotheses (Web: scipy.stats.kstest — SciPy v1.11.2 Manual (03/08/2023); Patrício, M., Ferreira, F., Oliveiros, B., et al. (2017); Razali, N. M., & Wah, Y. B. (2011)) are:

H_0: The Population follows the defined Distribution

H_1: The Population does not follow the defined Distribution

Since, the defined distribution is Normal in this case, the hypotheses (Razali, N. M., & Wah, Y. B. (2011); Oppong, F. B., & Agbedra, S. Y. (2016)) become:

H_0: The Population follows Normal Distribution

H_1: The Population does not follow Normal Distribution

If H_0 cannot be rejected, the population is considered to follow the tested distribution, which is the normal in this case, otherwise it is considered not to.

Shapiro-Wilk Test: A normality test for populations that originally had fewer than 50 observations (Patrício, M., Ferreira, F., Oliveiros, B., et al. (2017); Razali, N. M., & Wah, Y. B. (2011)) which is the threshold that is used in this case. Its two hypotheses (Web: scipy.stats.shapiro — SciPy v1.11.2 Manual (03/08/2023); Patrício, M., Ferreira, F., Oliveiros, B., et al. (2017); Oppong, F. B., & Agbedra, S. Y. (2016)) are:

H_0: The Population follows Normal Distribution

H_1: The Population does not follow Normal Distribution

If H_0 cannot be rejected, the population is considered to follow the normal distribution, otherwise it is considered not to. Its hypotheses and their meaning are essentially the same with the Kolmogorov-Smirnov Test, when the desired distribution is the Normal one.

T-Test: This test attempts to determine if the means of the two compared populations are the same. The two hypotheses of the T-test (Web: scipy.stats.ttest_ind — SciPy v1.11.2 Manual (07/08/2023); McGee, M. (2018)) are:

H_0: The Means of the Two Populations are Statistically the Same

H_1: The Means of the Two Populations are Not Statistically the Same

Whenever the H_0 cannot be rejected, which signifies that the two populations' means are the same, there are no statistically significant differences between the two compared populations. If the said hypothesis is rejected however, the population's statistical difference in their means entails that statistically significant differences exist between them.

Mann-Whitney Test: The two hypotheses of the Mann-Whitney Population Homogeneity test (Web: scipy.stats.mannwhitneyu — SciPy v1.11.2 Manual (08/08/2023); Milenović, Ž. (2011)) are:

H_0: The Two Populations have the same underlying Distribution

H_1: The Two Populations do not have the same underlying Distribution

An alternative form to write the two hypotheses would be:

H_0: The Two Populations are Homogenous

H_1: The Two Populations are Not Homogenous

which is how the test’s results are interpreted.

Whenever the H_0 cannot be rejected, signifying the homogeneity of the population, there are no statistically significant differences between the two compared populations, whereas if said hypothesis is rejected, the population’s non-homogeneity means that statistically significant differences exist between them.

3.5.2 Normality Tests

In order to continue with either of the tests for finding if there are any differences between the two genders, a normality test is being conducted on both of the populations. Depending on the number of instances found in each of them, the possible normality tests are either Kolmogorov-Smirnov or Shapiro-Wilk; the first one being applicable if the number of observations from the examined class is at least 50, the second one otherwise. It turned out that, in all the examined cases, there were none where the populations of both genders followed the normal distribution at the same time in any confidence level, so all the tests done were Mann-Whitney.

3.5.3 Population Homogeneity Tests

The two hypotheses of the Mann-Whitney Population Homogeneity test, in the used case are:

H_0: The Populations of the Two Genders (Men and Women) are Homogenous

H_1: The Populations of the Two Genders (Men and Women) are Not Homogenous

Whenever the H_0 cannot be rejected, the population is homogenous and there are no statistically significant differences between the two genders. If the said hypothesis is rejected, the population’s non-homogeneity means that statistically significant differences exist between them.

Regarding the errors made by each student, it turns out that, for the 90% and 95% confidence levels, the only case with a statistically significant difference between the two genders is the 11th assignment of 2022. In that though, the only errors found were made exclusively by men, so the result may not be particularly unexpected. This finding, when combined with the results of the corresponding plots, suggests that the student’s gender does not play an important role regarding how susceptible a person is in making errors.

Only one difference was found in the case of the per gender examination of the assignment grades, in the 95% confidence level, which came from the 7th assignment from 2021. In this case though, seeing that the female assignment average is around 8.2 and the male one is 7.1, it seems the most probable that women achieved a statistically higher average grade than their male counterparts. When examining for the 90% confidence level, apart from the 7th assignment of 2021, differences were also found in that year’s 3rd, 4th and 5th assignments. For these three, although the averages per gender had a difference of less than 1 grade, women always had the leading performance, leading to the same conclusion as in the 7th assignment. It should also be noted that women’s 25% percentile grades were higher than those of their male counterparts with the differences being usually quite noticeable. So, it seems that women do have the higher academic achievement than men, whenever differences were observed.

In the case of the students’ final grades, no differences were found in any of the examined cases for any confidence level.

For the 99% confidence level, no statistically significant differences were found between the genders in any case.

In the end, the only differences per gender found were two for the 95% confidence level and five for the 90%. In 95%, one is in the errors per individual and the other one in each person’s assignment grades. In 90%, one is in the errors per individual and four are in each person’s assignment grades. When compared to the overall number of tests made, which are 78, as well as that with each higher confidence level the number of differences shrinks, the student’s gender may not be a factor that heavily modifies neither their amount of errors nor their academic performance. When studied along with the three plots of the visualizations section, it seems that the student being of female gender seems to be associated with a slightly higher level of academic performance.

3.6 Association Rule Mining

The student-level data used in the previous parts is undergoing the appropriate transformation in order to be used for the creation of association rules. Their mining aims to determine the factors that contribute in students’ passing the course or if there are any preventing them from achieving this goal. The attributes used are the following:

- **Male/Female**: if the student is of male or female gender (only one value can be true between them).

- **Error_01,…,Error_46**: if the student has done any of the examined compilation errors, notes or warnings in any of their submitted exercises (multiple values can be true among them).
- **First_Enrollment/NOT_First_Enrollment**: whether the student has not been enrolled in the course in a previous year, or differently if this is their first time participating in the course (only one value can be true between them).
- **Took_Course_in_its_Semester/NOT_Took_Course_in_its_Semester**: whether the student is enrolled in the course during their second semester in the department, or in other words if they are participating in the course the semester it is being instructed according to the department's program (only one value can be true between them).
- **Submitted_Assignment_01,…,Submitted_Assignment_10**: if the student has submitted each of the assignments of each year, based on the consideration that these assignments are assumed to be 10 and that in order for an assignment to be considered as submitted, the student must submit at least one of its exercises (multiple values can be true among them).
- **Assignment_01_over_Base,…,Assignment_10_over_Base**: with the same consideration if the student managed a grade over base, at least 5/10 in each of the year's exercises (multiple values can be true among them).
- **Passes_the_Course/NOT_Passes_the_Course**: whether the student has managed to gain an overall course grade of at least 5 out of 10 (only one value can be true between them).

The analysis done in this part concerns mainly whether the student has passed the course or not and the role other factors, like their gender, have in this procedure. The algorithm chosen is Apriori with minimum support threshold 0.1 and minimum confidence threshold 0.01. The metrics used for the rules' evaluation are the subsequent:

- **Support**: the ratio of how many times both of the rules' parts appear concurrently to the total number of transactions:

$$\frac{Number\ of\ actions\ including\ subactions\ A\ and\ B\ concurrently}{Total\ number\ of\ actions}$$

It is also known as "Coverage", and it is "symmetric" (Merceron, A., & Yacef, K. (2008, June)).

- **Confidence**: the ratio of how many times both the left and right part of the rule appear to only its left part or the ratio of the antecedent and consequent to only the antecedent:

$$\frac{Number\ of\ actions\ including\ subactions\ A\ and\ B\ concurrently}{Number\ of\ actions\ including\ subaction\ A}$$

  It is also known as "Accuracy" and unlike Support, it is not symmetric, meaning Confidence(A→B) != Confidence(B→A) (Merceron, A., & Yacef, K. (2008, June)).

- **Lift**: describes the ratio of how much more or less common is the appearance of both parts of the rule to just its right part or its consequent (Merceron, A., & Yacef, K. (2008, June)):

$$\frac{Confidence(A \rightarrow B)}{Support(B)}$$

- **Cosine Similarity**: expresses how common it is to encounter both the antecedent and the consequent of a transaction concurrently to finding at least one of them (Merceron, A., & Yacef, K. (2008, June)):

$$\frac{Support(A\ and\ B)}{\sqrt{Support(A) * Support(B)}}$$

- **Jaccard Similarity**: evaluates the ratio of how much more or less common it is to find both parts of a transaction to finding only one of these parts but not both concurrently (Abdullah, Z., Herawan, T., Ahmad, N., et al. (2011); Lang, Q., Zhang, C., Qi, H., et al. (2023)):

$$\frac{Support(A\ and\ B)}{Support(A) + Support(B) - Support(A\ and\ B)}$$

In the following table, some of the rules that have as a consequent that the student passes the course, ordered by descending cosine similarity metric, are shown.

Table 3-4: A very small subset of potentially interesting association rules that show how the above factors contribute in someone's passing the course, ordered by descending cosine metric

Rule	**Sup.**	**Conf.**	**Cos.**	**Lift**	**Jac.**
{submitted_assignment_02}→{passes_the_course}	0.35	0.93	0.78	1.70	0.62
{submitted_assignment_03}→{passes_the_course}	0.34	0.93	0.76	1.70	0.59
{assignment_02_over_base}→{passes_the_course}	0.30	0.96	0.73	1.76	0.54

{submitted_assignment_02, assignment_02_over_base}→{passes_the_course}	0.30	0.96	0.73	1.76	0.54
{submitted_assignment_03, submitted_assignment_02}→{passes_the_course}	0.31	0.96	0.73	1.75	0.54
{submitted_assignment_04}→{passes_the_course}	0.31	0.95	0.73	1.74	0.54
{submitted_assignment_01}→{passes_the_course}	0.32	0.92	0.73	1.68	0.55
{assignment_03_over_base}→{passes_the_course}	0.30	0.94	0.71	1.71	0.52
{submitted_assignment_05}→{passes_the_course}	0.26	0.96	0.67	1.75	0.47
{first_enrollment, submitted_assignment_02}→{passes_the_course}	0.25	0.96	0.66	1.75	0.44
{took_course_in_its_semester, submitted_assignment_02}→{passes_the_course}	0.24	0.96	0.65	1.75	0.44
{submitted_assignment_03, submitted_assignment_01, submitted_assignment_02, submitted_assignment_04}→{passes_the_course}	0.23	0.97	0.64	1.78	0.41
{took_course_in_its_semester}→{passes_the_course}	0.32	0.70	0.64	1.27	0.47
{took_course_in_its_semester, first_enrollment}→{passes_the_course}	0.32	0.70	0.64	1.27	0.47
{first_enrollment}→{passes_the_course}	0.33	0.69	0.64	1.26	0.47
{male, submitted_assignment_02}→{passes_the_course}	0.23	0.93	0.62	1.70	0.40
{male, submitted_assignment_03}→{passes_the_course}	0.22	0.91	0.60	1.67	0.38
{male}→{passes_the_course}	0.38	0.51	0.59	0.93	0.41
{error_43 (warning: the ‘gets’ function is dangerous and should not be used)}→{passes_the_course}	0.17	0.94	0.55	1.72	0.31
{male, took_course_in_its_semester}→ →{passes_the_course}	0.21	0.67	0.51	1.23	0.32
{male, first_enrollment, took_course_in_its_semester}→{passes_the_course}	0.21	0.67	0.51	1.23	0.32
{male, first_enrollment}→{passes_the_course}	0.21	0.66	0.51	1.21	0.32
{submitted_assignment_03, female}→{passes_the_course}	0.12	0.95	0.46	1.74	0.22
{submitted_assignment_02,	0.13	0.93	0.46	1.70	0.23

female}→{passes_the_course}					
{female}→{passes_the_course}	0.17	0.66	0.45	1.21	0.27
{submitted_assignment_02, NOT_took_course_in_its_semester}→ →{passes_the_course}	0.11	0.87	0.42	1.59	0.19
{NOT_took_course_in_its_semester}→ →{passes_the_course}	0.23	0.42	0.42	0.77	0.26
{submitted_assignment_02, NOT_first_enrollment}→{passes_the_course}	0.11	0.87	0.41	1.58	0.19
{submitted_assignment_02, NOT_first_enrollment, NOT_took_course_in_its_semester}→ →{passes_the_course}	0.11	0.87	0.41	1.58	0.19
{NOT_first_enrollment}→{passes_the_course}	0.22	0.42	0.41	0.77	0.26
{NOT_first_enrollment, NOT_took_course_in_its_semester}→ →{passes_the_course}	0.22	0.42	0.41	0.77	0.26
{took_course_in_its_semester, female}→{passes_the_course}	0.11	0.74	0.39	1.36	0.19
{took_course_in_its_semester, first_enrollment, female}→{passes_the_course}	0.11	0.74	0.39	1.36	0.19
{first_enrollment, female}→{passes_the_course}	0.11	0.74	0.39	1.35	0.19

It turns out that most of the rules that end up with the student passing the course are of the form: {submitted_assignment_XY}, {assignment_XY_over_base}, {submitted_assignment_XY, assignment_XY_over_base} or any combination of them, with XY being any number between 01 and 10. In more detail, the assignment subjects whose knowledge seems the most valuable in passing the course are the 2nd, 3rd, 4th, 1st and 5th, which cover the first four of the course's subjects in the order of: 2)Stack, 3)Queue, 4)List – Dynamic Stack – Queue, 1)Sets. It also seems that if a student is participating in the course for the first time or the participation is done during the semester the department's program dictates, this person has a higher chance of passing the course.

Regarding the role the gender of the student has in this situation, the results are not entirely clear. By assessing the produced rules with the lift and cosine similarity metrics, as a means for determining their interestingness, a measure proposed by

(Merceron, A., & Yacef, K. (2008, June)), it seems that the male students who belong in the categories mentioned above, meaning those who complete these assignment and/or either participate in the course for their first time or are participating in it in the course's semester, are greatly associated with passing the course. Their female counterparts have similarly high lift values but much smaller cosine similarity values. What is interesting though is that, as seen above, the factor of someone being of female gender on its own is positively associated with passing the course, which is not the case for the male students as that rule's lift is less than 1.

Table 3-5: The entirety of association rules with a lift value at least 1 that result in a person not passing the course, ordered by descending cosine metric

Rule	Sup.	Conf.	Cos.	Lift	Jac.
{NOT_took_course_in_its_semester}→ →{NOT_passes_the_course}	0.31	0.58	0.63	1.28	0.46
{male}→{NOT_passes_the_course}	0.37	0.49	0.63	1.09	0.44
{NOT_first_enrollment}→{NOT_passes_the_course}	0.30	0.58	0.62	1.28	0.45
{NOT_first_enrollment, NOT_took_course_in_its_semester}→ →{NOT_passes_the_course}	0.30	0.58	0.62	1.28	0.45
{male, NOT_took_course_in_its_semester}→ →{NOT_passes_the_course}	0.26	0.61	0.60	1.36	0.43
{male, NOT_first_enrollment}→{NOT_passes_the_course}	0.26	0.61	0.59	1.35	0.42
{male, NOT_took_course_in_its_semester, NOT_first_enrollment}→{NOT_passes_the_course}	0.26	0.61	0.59	1.35	0.42

From all the rules produced, the only ones that result in the student not passing the course and have a lift value over 1 are the seven seen above. It is of interest that, according to the rule with the highest cosine value, a student being male is a rather important factor for them not successfully completing the course, although more important roles arguably have either the student not participating in the course for the first time or not participating in it in the course's semester, judging by their lift values. The combination of at least one of these two factors with the person being male is arguably the biggest indicator that this person will not pass the course.

In the end, it turns out that the female gender is associated with passing the course while the male gender with the opposite. Yet, it seems that the male students' result variance tends to be greater than that of women, by possibly having numerically more students that are considered excellent, while at the same time also having numerically more students that do not seem to engage enough with the course.

Something else that could be considered interesting is that, among all the rules resulting in a student failing to pass the course, neither the factor of them being of female gender appears, nor any errors. While for the female gender it can be explained by the above findings, the situation for the errors means that the students are generally capable of managing to handle them at least by the time of the exams.

3.7 Clustering Results

Clustering can be used to discover the groups of observations formed in the examined data, as well as information like their number, density and certain other characteristics. Different algorithms will be able to determine different clusters or "communities" based on their attributes. In this section, the findings from the application of different clustering methods on the work's data are shown.

3.7.1 Clustering Experiment Details

This subsection provides details on the clustering methods as well as the metrics that are being used. Beginning with the first subject, the algorithms used in order to realize this clustering venue are the following three.

- **Agglomerative Clustering**: it is a "bottom-up" hierarchical clustering algorithm that begins with each observation being its own cluster and continues by linking together the clusters considered as the closest, until it reaches a point where only one cluster remains. Different metrics exist that can be used to achieve this linking step, like the minimum, maximum and average linkages. The act of merging each cluster can be represented in a visual mean through a tree (Web: 2.3. Clustering — scikit-learn 1.3.0 documentation (04/06/2023)).
- **Spectral Clustering**: this method requires the creation of a "Laplacian Matrix" for the graph to be clustered, which allows the graph's elements to be mapped to a lower-dimensional representation, so that the clustering can be performed there. In more details, as a prerequisite, a matrix representation of the dataset has to be made, through the Laplacian Matrix. Then, the eignenvalues and eigenvectors of

said matrix are computed, each datapoint is mapped to its corresponding lower dimentional representation through an eigenvector or a subset of them and these representations are sorted. The clusters are identified either by splitting these lower dimensional representations or by using on them more conventional clustering methods (Zafarani, R., Abbasi, M. A., & Liu, H. (2014) pp.188-192; Von Luxburg, U. (2007)).

- **Mean Shift Clustering**: it is an algorithm based on the idea of iteratively moving and renewing possible centroids, which function as the mean-points of their "neighbourhood", a term that refers to the space defined by a radius surrounding a centroid, towards the area that, in each case, is considered to have the highest density of observations. This change is being dictated by a "mean-shift vector", aiming to indicate the direction where the density increase is the highest, while the family of algorithms this "shifting" towards denser spaces belongs to is called "hill-climbing". Convergence is achieved once all the centroids cannot be moved anywhere else, as the highest density regions have already been integrated. In case two centroids with their neighbourhoods overlap, the one that has integrated the least amount of observations is rejected (Web: 2.3. Clustering — scikit-learn 1.3.0 documentation (04/06/2023); Web: scikit-learn/sklearn/cluster/_mean_shift.py at main – GitHub (04/06/2023)).

Regarding the metrics used for evaluating the clustering algorithms' results, most of them are based on the idea of evaluating the clusters without making use of the ground truth of the data's results. More detailed descriptions follow.

- **Silhouette Score**: it is the mean of silhouette coefficients from the entirety of the data points and it indicates how well defined the clusters are. These coefficients indicate how well an observation fits in the cluster it already belongs to versus how well it would fit in another adjacent cluster. Each observation's silhouette coefficient can receive a value in the space between -1 and 1. -1 means that the observation would fit significantly better in a different cluster while 1 suggests that it is well extremely well suited to its own cluster. The metric does not require knowledge of the "ground truth" of the data, of how it is supposed to be clustered by default (Web: 2.3. Clustering — scikit-learn 1.3.0 documentation (04/06/2023); Web: sklearn.metrics.silhouette_score (04/06/2023)).

- **Calinski-Harabasz Index**: it measures the "goodness" of the clustering results through a ratio alluding to the intra-cluster and inter-cluster variances. The values it can take are exclusively positive numbers and 0, although unlike the Silhouette Score, there is no upper limit, which translates to higher values signifying more successful clustering. Indeed, better results are indicated by higher inter-cluster divergence and lower intra-cluster divergence. It can also be found under the name "Variance Ratio Criterion" and it does not require knowledge of how the data is supposed to be clustered (Web: 2.3. Clustering — scikit-learn 1.3.0 documentation (04/06/2023); Wang, X., & Xu, Y. (2019, July); Gustriansyah, R., Suhandi, N., & Antony, F. (2020)).
- **Davies-Bouldin Index**: it is a measure of "goodness" of clustering results, based on a ratio of inter-cluster and intra-cluster distances. The results of the clustering procedure are considered better when there is lower inter-cluster distance and higher intra-cluster distance, meaning that the observations belonging in the same cluster are more "similar" between them, while those belonging in different clusters are more "different" between them. Unlike in the above two metrics, where higher values are better, here values approaching its lowest limit, which is 0, are preferable. Knowledge of the "ground truth" of the data's clustering is not required (Gustriansyah, R., Suhandi, N., & Antony, F. (2020); Web: 2.3. Clustering — scikit-learn 1.3.0 documentation (04/06/2023); Web: sklearn.metrics.davies_bouldin_score (04/06/2023)).
- **Purity**: this metric assumes that the label characterizing most observations in a cluster could also characterize the entire cluster. It is defined as the ratio of the summation of the observations having the "majority label" in each cluster to the summation of the total number of observations in all the clusters (Wu, Di & Zhang, Mengtian & Shen, Chao et al. (2020); Zafarani, R., Abbasi, M. A., & Liu, H. (2014) p.207; Manning, C. D. (2009) pp.393-394).

From the above metrics, the only one requiring some knowledge of the "ground truth" of the clustered data is purity. In this case, purity is examined for the students' gender, year and course pass (C. Pass) status.

The clustering experiments are done in two ways. In both cases, a subset of data features was chosen for the clustering to be applied, although the set of features chosen is not the same between these two times. In the first case, the features chosen are standardized and undergo through principal component analysis before the clustering algorithms are applied, while in the second case, the features are only standardized.

3.7.2 Clustering after applying PCA

In this part the results of the application of clustering methods on the study's data are shown. An important notice is that, the clustering methods were not directly applied on the dataset in its original form. Firstly, from all the available data columns or features, since many of them presented heavily overlapping information, only a subset of them were used, which would not cause VIF values over 5, preventing multicollinearity from being a major issue (Daoud, J. I. (2017, December); Kalnins, A. (2018)). The features chosen are "first_enrollment", "enrollment_semester", "error_count", "has_cheated", "submitted_assignment_count", "final_grade". The table with the VIF values of each feature is shown below:

Table 3-6: The Variance Inflation Factor of the Clustering Features

feature	VIF
first_enrollment	2.087287
enrollment_semester	1.210745
error_count	1.194092
has_cheated	3.957519
submitted_assignment_count	4.447209
final_grade	4.238331

This subset is then standardized and afterwards Principal Component Analysis is applied, from which the first two principal components produced are chosen. The variance explained is 52.8% from the 1st one and 22.6% from the 2nd, meaning they can account for a little more than 75% of the chosen data's total variance. The method of PCA application after standardization was used in order to make the data friendlier to the clustering procedure.

The ground truth of how the data is clustered is shown below:

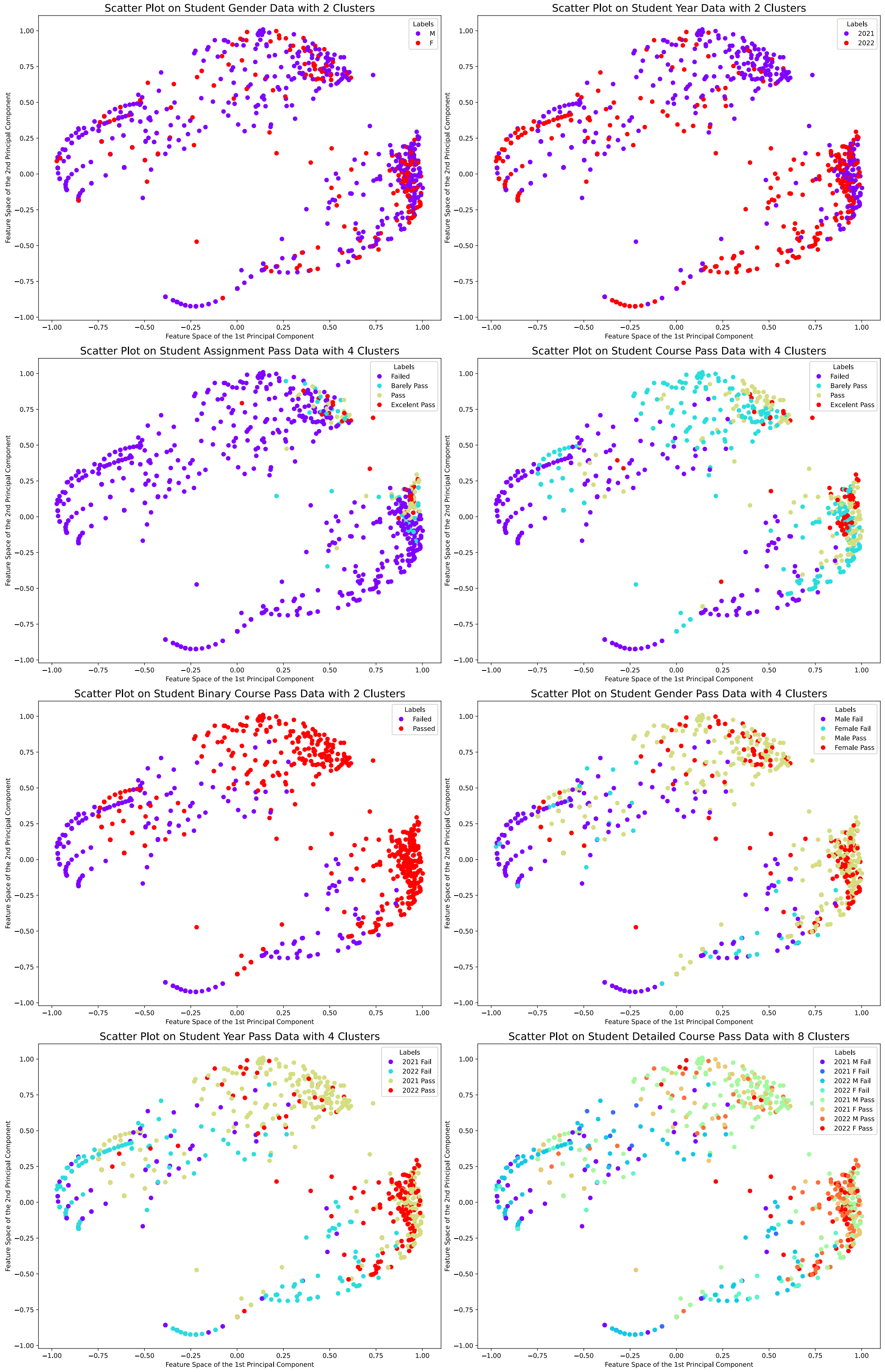


Figure 3-7: The Ground-Truth Clusters of the Data

Beginning with the Agglomerative Hierarchical clustering method, since the number of clusters that are supposed to be made is defined by the user, the numbers between 2 and 8 are tested. This range is chosen as the data comes from 2 years (2021 and 2022), is for students considered to belong in 2 genders (male and female) and can result in 2 outcomes (being the person passing the course or not), making the highest number of possible combinations 2^3=8. In the following table, the metrics for each of the mentioned cluster numbers are shown.

Table 3-7: Hierarchical Clustering Metrics

Clusters Metrics	2	3	4	5	6	7	8
Silhouette Score	0.505	0.588	0.664	0.659	0.577	0.557	0.533
Calinski-Harabaz	1460.086	2091.508	3601.752	3936.497	4290.55	4374.435	4434.682
Davies-Bouldin	0.854	0.647	0.476	0.49	0.633	0.565	0.676
Gender Purity	0.742	0.742	0.742	0.742	0.742	0.742	0.742
Year Purity	0.628	0.628	0.628	0.628	0.677	0.677	0.677
C. Pass Purity	0.832	0.832	0.832	0.85	0.873	0.873	0.873

According to the Silhouette and Davies-Bouldin indexes, the best number of clusters is 4, while according to Calinski-Harabaz index, it is 8. Since, though, the other two indexes do not have nearly as good values as Calinski-Harabaz for 8 clusters (the silhouette scores for 8 and 4 clusters respectively are 0.533 and 0.664, the corresponding Davies-Bouldin indices are 0.676 and 0.476, while the corresponding Calinski-Harabaz indexes are 4434.682 and 3601.752), it can probably be assumed that 4 manages the overall better results. But in order to give slightly more evidence supporting this assumption, the silhouette analysis plot for both cluster numbers is shown below. First is the plot for 4 clusters:

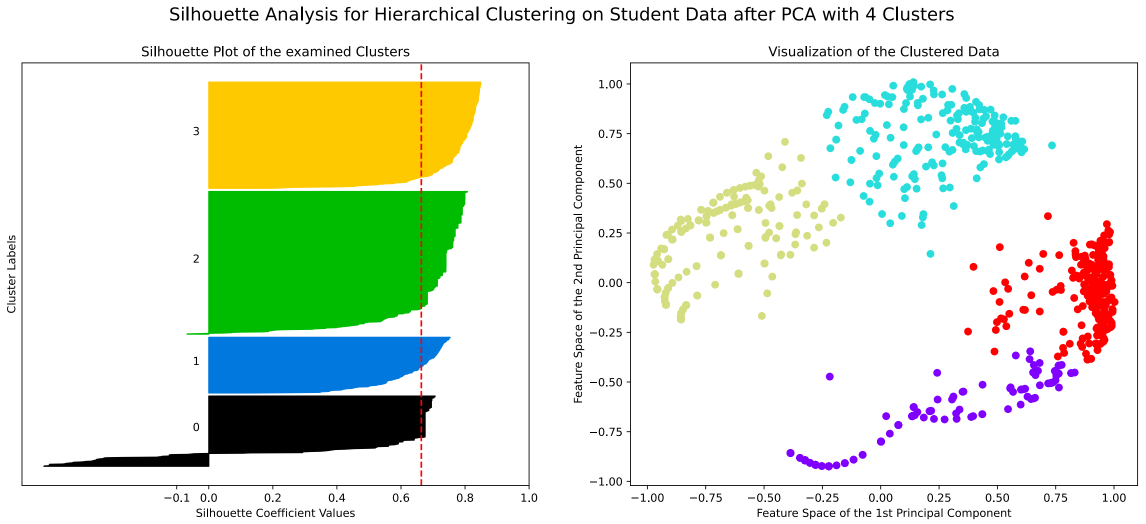


Figure 3-8: Silhouette Plot for Hierarchical Clustering with 4 Clusters

And below is the corresponding plot for the 8 clusters.

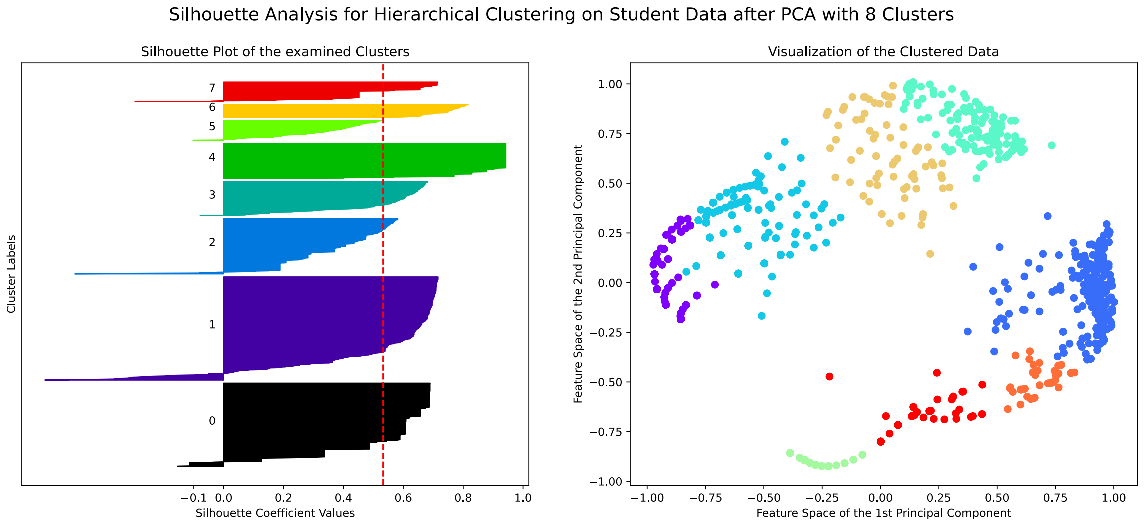


Figure 3-9: Silhouette Plot for Hierarchical Clustering with 8 Clusters

While the Silhouette plots do not indicate completely uniform clusters from terms of size, to the point where two clusters in both are visibly larger than the rest, since the plot with the 4 clusters shows higher uniformity among the 2 smaller clusters in comparison to the 6 smaller ones of the plot with the 8 clusters, the assumption that the 4 clusters are a better choice probably feels more plausible. As another interesting point, among the different cluster numbers, the Gender Purity metric is the exact same (at 0.742), meaning that all the different clustering attempts yield the same results regarding the per gender purity of the produced clusters. The other two purity metrics are not nearly as stationary, as the Year Purity remains at 0.628 from 2 to 5 clusters and then augments, reaching 0.677 for cluster numbers between 6 and 8, while the Course Pass purity starts from

0.832 with a cluster number up to 4, becomes 0.85 for 5 clusters and then remains at 0.873 for 6 up to 8 clusters.

The next method examined is Spectral clustering. Again, as the number of clusters that will be made is manually defined, the numbers tested are the range between 2 and 8 for the same reasons as above. The metrics for each cluster number are shown in the following table.

Table 3-8: Spectral Clustering Metrics

Clusters Metrics	2	3	4	5	6	7	8
Silhouette Score	0.547	0.612	0.695	0.537	0.544	0.524	0.52
Calinski-Harabaz	1866.141	2274.164	4554.084	1662.538	3545.597	3444.713	2739.474
Davies-Bouldin	0.735	0.515	0.431	0.991	0.648	0.715	0.719
Gender Purity	0.742	0.742	0.742	0.742	0.742	0.742	0.742
Year Purity	0.587	0.599	0.608	0.627	0.663	0.649	0.667
C. Pass Purity	0.811	0.835	0.843	0.839	0.853	0.853	0.866

According to all the metrics of clustering evaluation that are not using any ground truth information, the best result is achieved with 4 clusters. The silhouette plot for this number of clusters is the following.

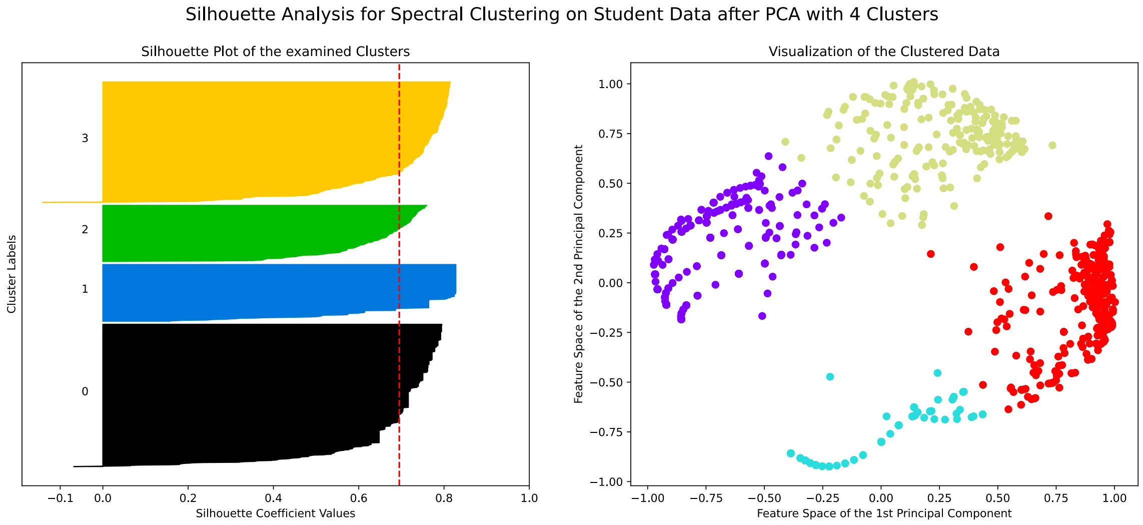


Figure 3-10: Silhouette Plot for Spectral Clustering

The Silhouette Score and Calinski-Harabaz Index are the highest among all the clusters, while Davies-Bouldin Index is the lowest. When the Spectral Clustering results are compared to the Hierarchical results, the Silhouette Score is higher (0.695 > 0.664), the Calinski-Harabaz Index is also higher (4554.084 > 3601.752), to the point where it is higher than even the value of the 8 clusters made with Hierarchical Clustering (4554.084 > 4434.682), while the Davies-Bouldin Index is lower (0.431 < 0.476). When compared to the hierarchical clustering results, while the best cluster number remains the same (4), the differences with the other cluster numbers' results are even more apparent. Apart from the purities, all the metrics are improved over the same result of Agglomerative Clustering. The Purity per Gender metric specifically happens to be identical not only between the different cluster numbers with this method, but also with all the hierarchical ones. The Year Purity is characterized by a gradual rise the more the cluster number augments, a pattern that is only slightly violated with the 7 clusters as there, the metric value is 0.649 which is visibly lower than 0.663 for 6 clusters but with 8 clusters, its value is now higher than both the 6 and 7 clusters (0.667 > 0.663 > 0.649). Compared to Hierarchical Clustering, the Spectral Clustering's minimum Year Purity is lower (0.587 < 0.628), the maximum is slightly lower (0.667 < 0.677), while for the 4 clusters, it is also comparatively lower (0.608 < 0.628). In all cases, Spectral's Year Purity is lower than Hierarchical's. The Course Pass Purity also has shifts between peaks and valleys, as the transition to 5 from 4 clusters causes a slight drop in the metric's value (0.839 < 0.843), while the transition to 7 clusters from 6, indicates a "plateau" seeing that the metric remains the same (at 0.853). Its value for 4 clusters (0.843) is neither the highest among the chosen clusters which is 0.866 for 8 clusters, neither its lowest, being 0.811 for 2 clusters. Compared to the same metric of Hierarchical Clustering, Spectral slightly surpasses it only in two cases, slightly for 3 clusters (0.835 > 0.832) and more visibly for 4 clusters (0.843 > 0.832), which also happens to be the number of optimal clusters for both methods.

The final method examined is the Mean-Shift clustering. Here, by default, the number of clusters is automatically defined, meaning that the metrics concern only one specific cluster number, so the comparisons will only be made with the above cluster numbers.

Table 3-9: Mean Shift Clustering Metrics

Clusters Metrics	4
Silhouette Score	0.695
Calinski-Harabaz	4568.141
Davies-Bouldin	0.434
Gender Purity	0.742

Year Purity	0.608
C. Pass Purity	0.841

Firstly, in this method, 4 clusters are also formed, which means that the optimal cluster number is the same according to all the methods used. The corresponding silhouette plot is shown below.

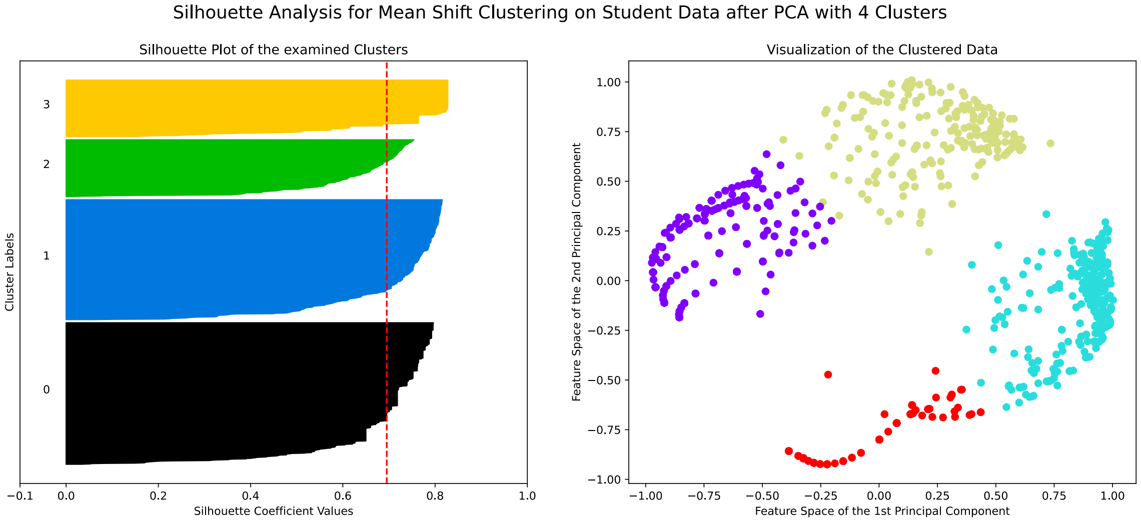


Figure 3-11: Silhouette Plot for Mean Shift Clustering

This method's results are somewhat interesting, as the metrics have different positions among the clustering techniques used. The metric of Purity per gender is identical to all the other cases (and fixed to 0.742). The metric of Purity per year is the same as Spectral Clustering for 4 clusters, at 0.608, and slightly lower that Hierarchical Clustering for the same cluster number (0.628). The Course Pass Purity is 0.841 which, when compared to the same cluster number of the other techniques, is marginally lower than Spectral Clustering's 0.843, but slightly higher than Hierarchical Clustering's 0.832. Regarding the metrics that do not require previous knowledge of how the clusters should be, its Silhouette Score is higher than that of Hierarchical and the same as that of Spectral Clustering methods (so 0.695 which is indeed higher than 0.664), its Calinski-Harabaz Index is the highest among all the methods' metric values with all the tested cluster numbers, thus also higher than the corresponding indices from both the Spectral and the Hierarchical methods (4568.141 > 4554.084 > 3601.752) but its Davies-Bouldin Index is between those of the other clustering methods, being lower than Hierarchical (0.434 < 0.476) and slightly higher than Spectral Clustering (0.434 > 0.431).

Although it cannot be definitively said which method is preferable for producing the clusters, with all the non-“Ground Truth” indices excluding the Calinski-Harabaz for Hierarchical Clustering, and according to all the 3 methods, the best number of clusters seems to be 4.

3.7.3 Clustering without using PCA

In this part the results of an alternative attempt for the application of clustering methods on the study’s data are shown. Like the previous subsection, the clustering methods were not directly applied on the dataset in its original form. Firstly, from all the available data columns or features, since many of them presented heavily overlapping information, only a subset of them was used, which is characterized by VIF values less than 7, so that multicollinearity will not be a very severe issue (Daoud, J. I. (2017, December); Kalnins, A. (2018)). The features chosen are "error_count", "final_grade", “gender”, "first_enrollment", "enrollment_semester", "has_cheated", "hw1_grade",…, “hw10_grade”, making the information more detailed than the previous attempt. The table with the VIF values of each feature is shown below:

Table 3-10: The Variance Inflation Factor of the 2nd attempt’s Clustering Features

feature	VIF
error_count	1.189537
final_grade	4.717944
gender	1.412999
first_enrollment	2.196400
enrollment_semester	1.278370
has_cheated	3.523940
hw1_grade	3.714151
hw2_grade	6.907497
hw3_grade	5.477734
hw4_grade	6.273076
hw5_grade	4.745142
hw6_grade	5.936767
hw7_grade	6.384713
hw8_grade	6.110087
hw9_grade	4.606865

hw10_grade	3.093035

This subset undergoes standardization in order to make the data friendlier to the clustering procedure, but principal component analysis is not applied in this case.

The ground truth of how the data is clustered is shown below:

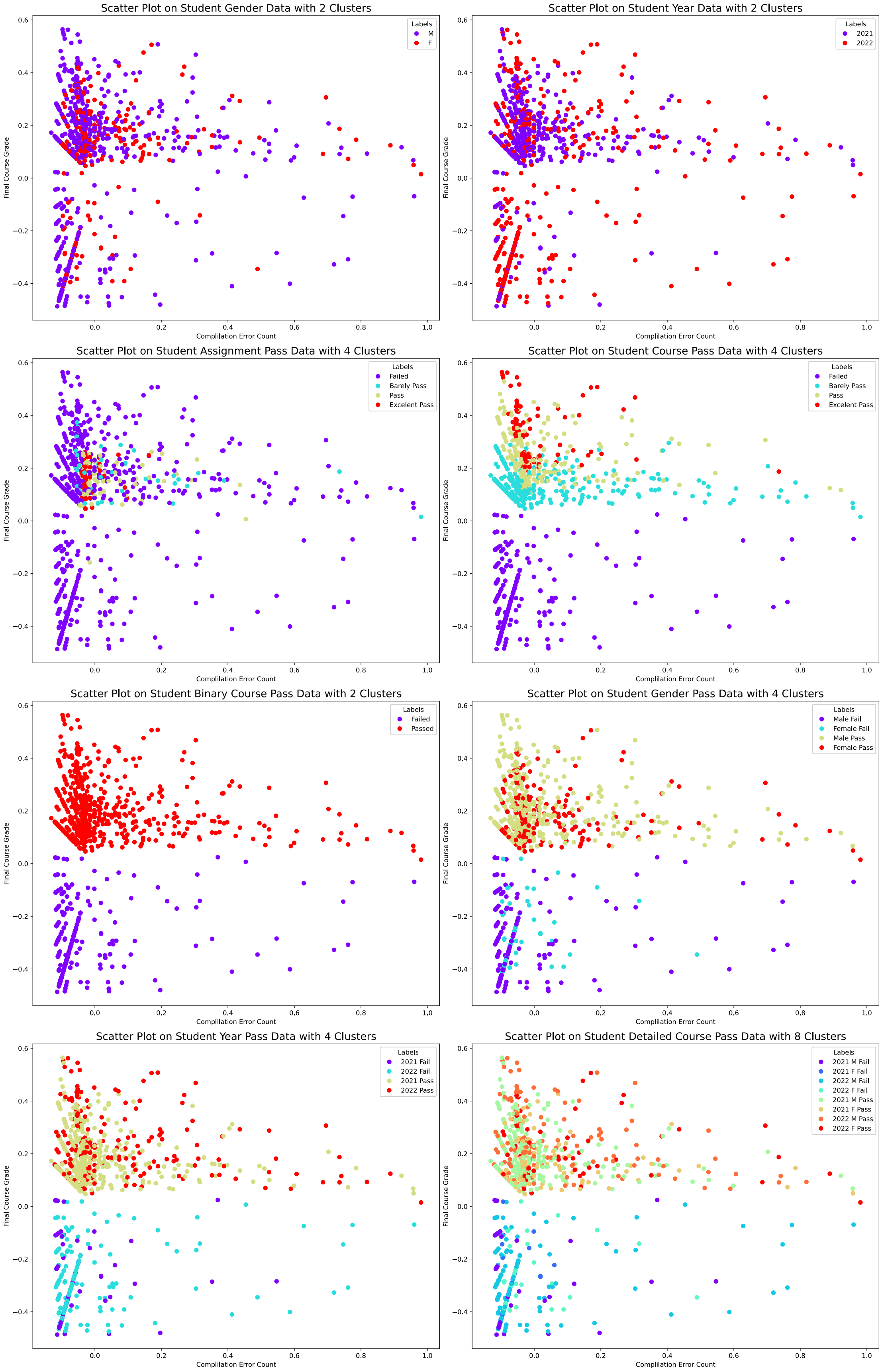


Figure 3-12: The Ground-Truth Clusters of the 2nd attempt's Data

Beginning with the Agglomerative Hierarchical clustering method, since the number of clusters that are supposed to be made is defined by the user, the numbers between 2 and 8 are tested. The reasoning is the same as with the previous clustering attempts, regarding the situations where the number of clusters to be created is set manually. This means that, as the data comes from 2 years (2021 and 2022), is for students considered to belong in 2 genders (male and female) and can result in 2 outcomes (being the person passing the course or not), the number of clusters tested ranges from 2^1=2 to 2^3=8. In the following table, the metrics for each of the mentioned cluster numbers are shown.

Table 3-11: Hierarchical Clustering Metrics with the non-PCA Data

Clusters Metrics	2	3	4	5	6	7	8
Silhouette Score	0.347	0.272	0.267	0.305	0.321	0.312	0.324
Calinski-Harabaz	699.535	528.552	466.832	439.129	435.042	410.49	387.232
Davies-Bouldin	1.281	1.575	1.549	1.537	1.418	1.339	1.342
Gender Purity	0.742	0.742	0.742	0.862	0.862	0.862	0.862
Year Purity	0.636	0.636	0.636	0.636	0.636	0.642	0.644
C. Pass Purity	0.763	0.763	0.763	0.763	0.783	0.783	0.783

According all 3 metrics that do not demand ground-truth knowledge, which are the Silhouette Coefficient Average, Calinski-Harabaz and Davies-Bouldin indexes, the best number of clusters is 2. This is because the Silhouette Coefficient Average and Calinski-Harabaz Index have their highest values for 2 clusters among the examined numbers, while the Davies-Bouldin Index has its lowest value for 2 clusters. The silhouette analysis plot for these 2 clusters is shown below.

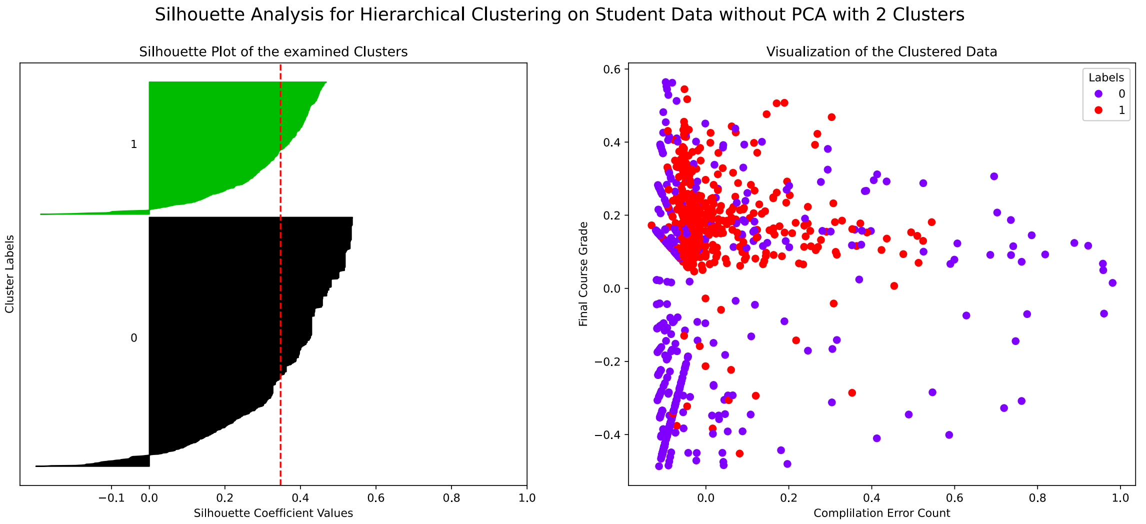


Figure 3-13: Silhouette Plot for Hierarchical Clustering with the non-PCA Data

Regarding the metrics that require clustering "Ground-truth" knowledge, the 3 Purity metrics do not indicate the same cluster number as the other 3 metrics. For 2 clusters, the Gender Purity is 0.742, the Year Purity is 0.636, while the Course Pass (C. Pass) Purity is 0.763. The exact same values are also seen with 3 and 4 clusters. The highest value for the Gender Purity is 0.862, which is achieved for cluster numbers between 5 and 8, the highest Year Purity is 0.644 for 8 clusters, while the highest Course Pass Purity is 0.783 and it is achieved for 6, 7 and 8 clusters. From the Figure 3-7 with the different real clusters the data is divided in, it can be seen that the clusters created bear resemblance to both the gender and binary course pass clusters, as indicated by the corresponding Purity indexes. It almost seems that the created clusters are a step between the course-pass and the gender clusters.

The next method examined is Spectral clustering. Again, as the number of clusters that will be made is manually defined, the numbers tested are the range between 2 and 8 for the same reasons as above. The metrics for each cluster number are shown in the following table.

Table 3-12: Spectral Clustering Metrics with the non-PCA Data

Clusters Metrics	2	3	4	5	6	7	8
Silhouette Score	0.362	0.29	0.313	0.316	0.327	0.314	0.329
Calinski-Harabaz	762.763	499.615	535.453	473.409	458.552	432.537	398.896
Davies-Bouldin	1.231	1.484	1.393	1.438	1.396	1.328	1.273
Gender Purity	0.742	0.742	0.742	0.849	0.893	0.878	0.878
Year Purity	0.616	0.589	0.624	0.624	0.616	0.621	0.665

C. Pass Purity	0.779	0.805	0.788	0.76	0.794	0.797	0.855

According to all the metrics of clustering evaluation that are not using any ground truth information, the best result is achieved with the 2 clusters, like in Hierarchical Clustering. The silhouette plot for 2 clusters with this technique is the following.

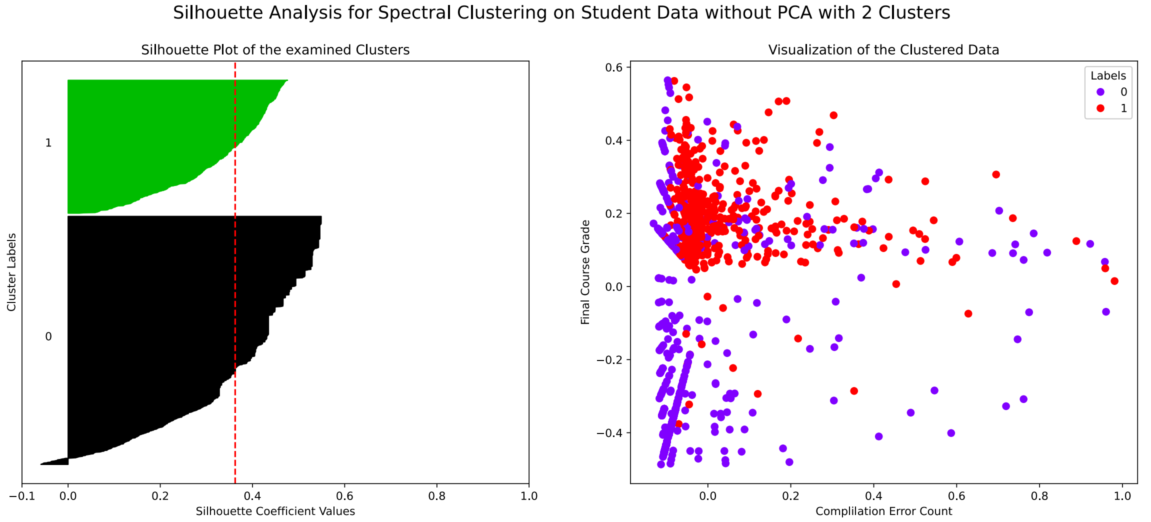


Figure 3-14: Silhouette Plot for Spectral Clustering with the non-PCA Data

The Silhouette Score and Calinski-Harabaz Index are the highest among all the clusters, while Davies-Bouldin Index is the lowest. The three Purity metrics are 0.742 for gender, 0.616 for year and 0.779 for course pass, from which none are the highest among the different tested cluster numbers. The best Gender Purity is achieved for 6 clusters and is 0.893, the best Year Purity is achieved for 8 clusters with 0.665, while the best Course Pass Purity is also achieved for 8 clusters and is 0.855. When compared to the hierarchical clustering results, while the best cluster number remains the same (being 2), differences exist. From this comparison of Spectral Clustering results to the Hierarchical results, the Silhouette Score is higher (0.362 > 0.347), the Calinski-Harabaz Index is also higher (762.763 > 699.535), while the Davies-Bouldin Index is lower (1.231 < 1.281). The Course Pass Purity is slightly higher too for the Spectral method compared to Hierarchical (0.779 > 0.763). Apart from Gender Purity, which is the same in both cases (0.742) and Year Purity which is slightly worse for the Spectral method compared to Hierarchical (0.616 < 0.636), the rest of the metrics are improved over the same result of Agglomerative Clustering. When the spectral clusters are compared to those of the different "ground-truth" clusters, they again resemble both the binary course pass clusters and the gender clusters, as indicated by the Purity indexes. Unlike with the agglomerative clusters, which seem to be in-between these two ground-truths, here the results seem closer to the course pass clustering more than the gender, despite the influence of both being noticeable.

The final method examined is the Mean-Shift clustering. Here, by default, the number of clusters is automatically defined, meaning that the metrics concern only one specific cluster number, so the comparisons with the results above will only be made with said cluster number.

Table 3-13: Mean Shift Clustering Metrics with the non-PCA Data

Clusters Metrics	2
Silhouette Score	0.363
Calinski-Harabaz	779.315
Davies-Bouldin	1.248
Gender Purity	0.742
Year Purity	0.621
C. Pass Purity	0.803

Firstly, in this method, 2 clusters are also formed, which means that the optimal cluster number is the same according to all the methods used. The corresponding silhouette plot is shown below.

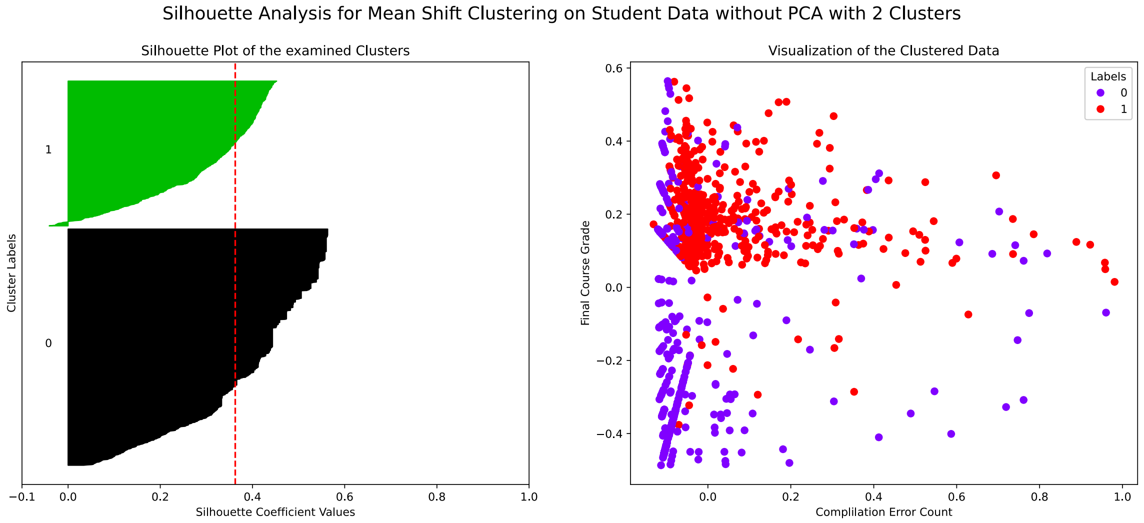


Figure 3-15: Silhouette Plot for Mean Shift Clustering with the non-PCA Data

The method’s Silhouette Score is visibly higher than that of Hierarchical and marginally higher than that of Spectral Clustering methods (so 0.363 which is indeed higher than

0.347 and 0.362 respectively). Its Calinski-Harabaz Index is higher than the corresponding indices from both the Spectral and the Hierarchical methods (779.315 > 762.763 > 699.535) but its Davies-Bouldin Index is between those of the other clustering methods, being lower than Hierarchical's (1.248 < 1.281) but higher than Spectral's (1.248 > 1.231). Then, the metric of Purity per gender is identical to the other cases for 2 clusters (and fixed to 0.742). The Mean Shift's Year Purity is the higher than Spectral's (0.621 > 0.616 respectively), but lower than Hierarchical's (0.621 < 0.636), while the Course Pass Purity is higher than both the Spectral and Hierarchical Methods (0.803 > 0.779 > 0.763). If the clusters of the mean-shift method are to be compared to those of the different "ground-truth" clusters, they also resemble both the binary course pass clusters and the gender clusters, as shown by the Purity metric values, just as the agglomerative and spectral clusters. The mean-shift clusters though might even be a little closer to the course-pass clusters when compared to the other clustering techniques used, which in turn makes the overall results closer to those of spectral clusters than the agglomerative ones, just with the influence of course pass ground truth over the gender being a little stronger.

Although it cannot be definitively said which method is preferable for producing the clusters, the best number of clusters seems to be 2. Spectral and Mean-Shift clustering produce very similar results as they both have very similar Silhouette Scores, though with Mean-Shift being marginally superior, the former has a lower (and better) Davies-Bouldin index while the latter has higher Calinski-Harabaz and Course Pass Purity values. So, in the end, it could be argued that the Mean-Shift method produces the best clusters among the different ones used.

3.7.4 Comparison of the results with and without PCA

The clustering approaches shown in the previous 3.7 subsections have their own strengths and weaknesses, which is the reason both are shown. At a very high level, it could be argued that the clusters produced with the method involving PCA manage generally higher metric values, while those produced without the involvement of PCA feel much more explainable or interpretable, given the known information for the data.

In more detail, on the one hand, regarding the metrics involved, especially when comparing the 3 metrics that do not require ground truth knowledge, it can be seen that their worst values with the first attempt are higher, usually substantially so, than the best values of the second attempt. The worst Silhouette Score in the first attempt is found for 2 clusters with the Hierarchical method and is 0.505, while the best corresponding metric value in the second attempt is achieved by the 2 clusters produced with the Mean-Shift method and is 0.363, which is visibly lower than 0.505. The worst Calinski-Harabaz Index in the first attempt is found again for 2 clusters with the Hierarchical method and is 1460.086, while the best corresponding metric value in the second attempt is also achieved by the 2 clusters produced with the Mean-Shift method and is 779.315, a value that is a bit more than half of the 1460.086. Then, the worst Davies-Bouldin Index in the first attempt is found for 5 clusters with the Spectral method and is 0.991, while the best corresponding metric value in the second attempt is also achieved by Spectral method when 2 clusters are made and is 1.231, which is much higher than 0.991, while for this metric, lower values are considered better.

On the other hand, the preferred clusters produced by the first attempt do not seem to correspond particularly well to any of the examined ground truth clusters for that attempt, something that happens across all three algorithms applied, while the preferred clusters with the second attempt indicate results that resemble both the Gender and even more the Course Pass ground truth clusters. The Agglomerative results indicate clusters that look similar to both of these previously known true clusters, the Spectral Results tend to look even more like the Course Pass clusters compared to the Gender ones, while the Mean-Shift clusters are seemingly the closest to the Course Pass ones and the least close to the Gender ones from all the “best” clusters produced in this second attempt.

3.8 Machine Learning Predictions

As demonstrated in the Related Work section of the study, having at least a general idea of students’ performance before the exams is very desirable for a department’s faculty. One of the applicable practices is training machine learning models capable of predicting the students’ performance in the examined course or courses before the exams, in order to intervene appropriately. In many cases, the models created manage extremely accurate results, although this can entail requiring data that could come from a variety of different sources, be difficult or impractical to collect or to organize into a single dataset.

In this part, an attempt is made to try and predict the student’s results based on only two features gathered from data available in the assignment assessment files: the compilation errors per student and the submitted assignments’ average grade. This is done at first in the end of the semester, just before the exams, and then for the data produced after each weekly assignment’s assessment is completed. In a little more detail, the variable that is attempted to be predicted is of binary nature where the value of 1 indicates that the examined student passes the course successfully whereas a value of 0 indicates the opposite. These features are chosen as they provide information about the students exclusively through their assignment assessment data, as well as because their Variance Inflation Factor, when examined alongside each student’s final grade, is always lower than 3 (or more precisely less than 2). The VIF of these three characteristics is shown below:

Table 3-14: The Variance Inflation Factor of the Features for Classification

feature	VIF
error_count	1.095056
average_assignment_grade	1.923464
final_grade	1.998182

In order to do so, four types of machine learning models are created belonging in the categories of logistic regression, decision tree, support vector machine and neural network. In all cases, the variables used for making predictions are only the student’s number of compilation errors found in their submitted assignments using the 46 most common categories, in combination with their average grade from the assignments they submitted by considering that each year’s assignments are just 10, not 12 or 11 as it was the case in 2021 and 2022 respectively. It should be noted that later, when the maximum assignment number (up to 10) differs depending on how many assignments up to that point have been submitted, this assignment number concerning data up to that point is used for finding the average. Something else to note before starting with the model creation is that, while all the analysis performed so far was with the data from all the students enrolled in the course during the two years of 2021 and 2022, at least a part of them constitute a category that can be called as “completely inactive students”. Students that can be considered as belonging into this category, given the data used in the study, do not make any errors in their assignments, simply because they do not submit any assignments, while they also lack a proper final course grade as they did not come to the course’s final exams. So, from the 1323 students used in the rest of the study, by filtering those who have 0 compilation errors, 0 submitted assignments and have managed a 0 as their final course grade, 940 individuals remained. While these “completely inactive” students reflect something that can happen in at least some cases in the Greek tertiary education, their existence in the training/validation and/or testing data subsets could tamper with the real predictive behavior of the created classifiers, even though that behavior would have been more representative of this case.

The creation and evaluation of the models was done with the Stratified K-fold Cross Validation process (with K = 10). More precisely, from the data of both the available years, being 2021 and 2022, two thirds of it was used for the K-fold training and validation process, while the other one third was excluded and used solely after the creation of each fold’s models, in order to test them on unknown data. As a means to achieve this dichotomy while preserving the percentages of each years’ students in the two new data subsets in levels broadly similar to those of the original data, a Stratified 3-fold split was performed on the data, where the target class in both folds is the person’s year (2021 or 2022). So from this attempt, two subsets are created, the first having 626 students and the second the remaining 314. As a means to determine in the most accurate manner the year a student took the course in, two steps were made. Firstly, a student’s data was considered to belong in 2022 if their id appeared in the supplementary data file of 2022, otherwise they were considered to belong in 2021. Then, depending on if a student’s assignment grade average from all the 10 weeks is higher in 2021 than in 2022, their year is automatically set to be 2021, otherwise it is left to what it already was from the previous step. The term “Stratified” refers to that, in each of the folds created, the representation of both target classes (passes the course, fails the course) is of essentially the same percentage as in the data the folds were created from. The undersampling method NearMiss and more precisely, from its Sci-kit Learn implementation, its 3rd version with 4 neighbours is also used exclusively on the training data subset as a method to equalize the number of observations in both target classes in the folds, seeing that the two classes would have been rather unbalanced otherwise, due to the removal of said completely inactive students.

The next part describes the metrics used for the study. Before going to explain them, it should be noted that the following Acronyms will be used:

- **True Positive** or **TP**: a positive element that is correctly identified as positive.

- **False Positive** or **FP**: a negative element incorrectly considered to be positive, indicates a Type 1 error.
- **False Negative** or **FN**: a positive element incorrectly considered to be negative, indicates a Type 2 error.
- **True Negative** or **TN**: a negative element that is correctly identified as negative.

After having seen the acronyms used and their meaning, the metrics used for evaluating the classification models are the following:

- **Accuracy**: the ratio of the correct predictions made to the total number of observations to predict:

$$\frac{TP + TN}{TP + FN + FP + TN}$$

- **Precision**: the ratio of the number of correctly predicted observations to that of all the observations that were considered to belong in the examined group:

$$\frac{TP}{TP + FP}$$

- **Recall**: the ratio of the number of correctly predicted observations to that of all the observations actually belonging in the examined group:

$$\frac{TP}{TP + FN}$$

- **F1 Metric**: a “harmonic mean” between precision and recall, the fracture multiplied by 2 that has the numerator precision * recall and the denumerator precision + recall. The possible values it can have are in the range between and including 0 and 1 (Web: sklearn.metrics.f1_score — scikit-learn 1.3.0 documentation (24/06/2023)):

$$2 * \frac{Precision * Recall}{Precison + Recall}$$

- **Area Under the Curve** or **A***: it measures how good is a model’s discriminatory capability, which is its ability to output correct predictions. Visually, this can be represented as the part of a plot that is situated under the “Receiver operating characteristics” curve.
- **Cohen’s Kappa**: the ratio with the numerator of the achieved agreement minus the expected agreement and denumerator the 1 minus the expected agreement. As the expected agreement is the agreement by chance, this quantity expresses how

much better than chance is a model’s inter-rater or, more suitably for this use-case, inter-classifier agreement:

$$\frac{Agreement - Expected\ Agreement}{1 - Expected\ Agreement}$$

Following the metrics are some additional details about the models created for this section’s purposes:

- The Logistic Regression model uses the default Sklearn setting for its tolerance level, being 10^{-4}, but uses the “balanced” class weight so as to treat both possible classes (of passing or failing the course) as of equal importance, its maximum iteration number has been raised to 1 million in order to avoid cases where convergence is not reached in the permitted number of iterations, while the number of CPU processing cores it can use is not limited to only 1, as default, but is defined to automatically be the same as the number of cores available in the CPU it is being run, thus becoming -1 (Web: sklearn.linear_model.LogisticRegression - scikit-learn (23/05/2023)).
- The Decision Tree uses the “gini index” instead of the default “entropy” criterion to calculate the information gain, uses the “balanced” class weight and is limited to a maximum depth of 6, in order to gain better performance and avoid overfitting. It also defines the random state to be 1, so as to allow the reproducibility of the results.
- The type of Support Vector Machine used is the Linear Support Vector Classifier or LinearSVC with a maximum iteration number of 1 million in order to avoid cases where convergence is not achieved with the predetermined number of iterations, the parameter telling to solve the dual version of the problem is set to be false, as the number of records is larger than the number of features, not similar or smaller which would make solving the dual problem preferable (Web: sklearn.svm.LinearSVC — scikit-learn 1.3.0 documentation (12/08/2023)), while the tolerance is made to be 10^{-4}.
- Regarding the Neural Network, two types are made. The first one is a custom made Sequential model, using commands from Tensorflow, while the other one is the Multi-Layer Perceptron classifier from Sci-kit Learn.
  - The Sequential Neural Network’s structure is of 4 dense layers. The 1st is the input layer and is comprised of 640 neurons, the 2nd and 3rd are

hidden layers and are both comprised of 1280 neurons, while the 4th is the network's output layer. All the layers, apart from the output, are using the "normal" distribution for weight initialization, the l2 method of weight normalization with a value of 0.001, the "zeros" as the bias initializer of choice and the simple ReLU activation method. The network's output layer is only of one sigmoid neuron, and the loss function is BinaryCrossentropy, as the predictions are binary in nature. The used optimizer is Stochastic Gradient Descent with a learning rate of $16*10^{-2}$ and a Nesterov-type momentum of $36*10^{-2}$. The mini-batch size used is 24. By default, the network is trained for 320 epochs. From the data designated exclusively for the training procedure, so from 9 of the 10 folds each time, its 1/10th is used for a first and smaller validation step different from the recorded one which uses the data of the excluded i-th fold each of the 10 times it happens. The validation loss of this first smaller validation step is used so that that if the loss function does not improve after 6 epochs in a row, the training is stopped prematurely (Web: tf.keras.callbacks.EarlyStopping | TensorFlow v2.13.0 (10/08/2023)).

- The Multi-Layer Perceptron has its hidden layer size to be 32, the solver of choice to be "adam", a random state of 1 in order to allow the reproducibility of the results, as well as a maximum iteration number of 30 thousand.

3.8.1 Results in the semester's end

In the following part, the results of the models with the data from all the available assignments are used to predict whether the students will pass the course or not.

Table 3-15: The 10-fold validation result averages and standard deviations

Model Metric	Logistic Regression	Decision Tree	Support Vector Machine	Sequential Neural Network	Multi-Layer Perceptron
Accuracy	0.689 (+/- 0.053)	0.701 (+/- 0.046)	0.674 (+/- 0.047)	0.671 (+/- 0.039)	0.684 (+/- 0.043)
Precision	0.925 (+/- 0.059)	0.869 (+/- 0.053)	0.948 (+/- 0.052)	0.928 (+/- 0.047)	0.868 (+/- 0.049)

Recall	0.638 (+/- 0.047)	0.712 (+/- 0.061)	0.599 (+/- 0.046)	0.61 (+/- 0.042)	0.684 (+/- 0.046)
F1 Metric	0.754 (+/- 0.043)	0.78 (+/- 0.037)	0.733 (+/- 0.042)	0.735 (+/- 0.035)	0.764 (+/- 0.033)
A* - AUC	0.739 (+/- 0.073)	0.691 (+/- 0.067)	0.749 (+/- 0.061)	0.732 (+/- 0.054)	0.683 (+/- 0.067)
Cohen’s Kappa	0.365 (+/- 0.111)	0.322 (+/- 0.106)	0.365 (+/- 0.089)	0.346 (+/- 0.078)	0.3 (+/- 0.105)

The above table shows the averages and standard deviations in the form of “metric value (+/- standard deviation)” from the results of the validation process from the 10-fold training and validation used for creating the machine learning models. By looking at the metrics, it can be seen that all of the models manage only a very middling performance level. Higher values are generally desirable for all the metrics and, apart from Precision and possibly F1, none can be considered particularly good. According to Accuracy, Recall and F1, the Decision Tree model will have the best predictive performance, while according to Precision, A* and Kappa, the Support Vector Machine should have the best predictive performance. While, on their own, these metrics are unremarkable, the problem comes from the fact that, while the 10-fold average Kappa values are always at least at the level of 0.3, which is the usual minimum for the models to be considered as decent enough (Asif, R., Merceron, A., Ali, S. A., et al. (2017)), when taken into account the worst case scenario for the metric, all five models manage Kappa values that can be significantly under 0.3. Support Vector machine, which alongside the Logistic Regression manage the best average Kappa of 0.365, slips to 0.365-0.089=0.276, the Sequential Neural Network from 0.346 becomes 0.346-0.078=0.268, the mentioned Logistic Regression goes from 0.365 to 0.365-0.111=0.254, taking a value that is smaller than Neural Network’s, the Decision Tree falls from 0.322 to 0.322-0.106=0.216, while the Multi-Layer Perceptron falls from 0.3 to 0.3-0.105=0.195 a value smaller than even 0.2. Given the above and especially the worst-case scenarios with the Kappa values, the predictions will need to be made with caution, although the less than stellar metrics could be explained by the fact that the training data went through an undersampling procedure, which further diminished the available records down to the number of the minority class from the two.

Table 3-16: The 10-fold testing result averages and standard deviation

Model / Metric	Logistic Regression	Decision Tree	Support Vector Machine	Sequential Neural Network	Multi-Layer Perceptron
Accuracy	0.735 (+/- 0.004)	0.756 (+/- 0.011)	0.719 (+/- 0.009)	0.726 (+/- 0.018)	0.766 (+/- 0.007)
Precision	0.948 (+/- 0.004)	0.921 (+/- 0.011)	0.97 (+/- 0.005)	0.961 (+/- 0.01)	0.927 (+/- 0.008)
Recall	0.713 (+/- 0.005)	0.765 (+/- 0.02)	0.675 (+/- 0.009)	0.692 (+/- 0.03)	0.773 (+/- 0.012)
F1 Metric	0.814 (+/- 0.003)	0.836 (+/- 0.009)	0.796 (+/- 0.007)	0.804 (+/- 0.017)	0.843 (+/- 0.006)
A* - AUC	0.772 (+/- 0.006)	0.74 (+/- 0.018)	0.792 (+/- 0.01)	0.784 (+/- 0.009)	0.754 (+/- 0.014)
Cohen's Kappa	0.385 (+/- 0.008)	0.375 (+/- 0.021)	0.388 (+/- 0.015)	0.388 (+/- 0.017)	0.398 (+/- 0.017)

The above table shows the averages and standard deviations from testing results of the models created with the 10-fold cross validation technique. By looking at the metrics it can be seen that, differences exist with the validation results as, according to all the metrics, the models' performance has been improved, reaching levels that could be considered as somewhat decent. According to Accuracy, Recall, F1 and Kappa, which are the two thirds of the used metrics, the best predictions will be done by the Multi-Layer Perceptron model, while according to the other third, that consists of Precision and A*, the best predictions will be done by the Support Vector Machine. The interesting part comes from the fact that, according to most metrics including Kappa, Multi-Layer Perceptron is the model with the best predictive capabilities, which is the opposite of what happened in the 10-fold validation process, where the same model was considered the least preferable according to those Kappa values. While Multi-Layer Perceptron and Support Vector Machine are the best models according to the metrics, both of them also have some weaknesses that should be considered. Support Vector Machine has the lowest average values for Accuracy and Recall, which are 0.719 and 0.675 respectively, while the Multi-Layer Perceptron has the second lowest Precision and A* values of 0.927 and 0.754. The only model that has worse values in these metrics is the Decision Tree with a Precision of 0.921 and an A* of 0.74. In general, it seems that different metrics indicate different models as the most suitable for Predictions. For example, by taking the 10-fold averages for Accuracy, Recall and F1, the models from best to worst seem to be MLP>DT>LR>NN>SVM, although, in the worst case scenario, SVM would marginally surpass the NN. This happens as the worst Accuracy values for these two models are 0.719-0.009=0.71 and 0.726-0.18=0.708, the respective Recall values become 0.675-0.009=0.666 and 0.692-0.03=0.662, while the respective F1 values become 0.796-0.007=0.789 and 0.804-0.017=0.787. Then, regarding the Precision and A* metrics, the preferred models become SVM>NN>LR>MLP>DT. According to the average Kappa values, the models' descending order becomes MLP>SVM=NN>LR>DT, while according to the worst case Kappa values, they become MLP>LR>SVM>NN>DT, as although the NN has the same 10-fold average Kappa value with the SVM of 0.388, it also has slightly higher variance at 0.017 and 0.015 respectively, while the LR model which manages a value of 0.385 which is slightly lower than the 0.388 of the other two models, it also has half of the variance, being at just 0.008 instead of 0.015 or 0.017. When examining the worst cases for the Cohen's Kappa metric, it can be seen that, across all the models in the order of MLP, LR, SVM, NN and DT, their values become 0.398-0.017=0.381, 0.385-0.008=0.377, 0.388-0.015=0.373, 0.388-0.017=0.371 and 0.375-0.021=0.354, so 0.381, 0.377, 0.373, 0.371 and 0.354 respectively, all of which are above 0.35 thus also higher than 0.3, which means that all the models are decent enough, based on the remarks about the Kappa value from (Asif, R., Merceron, A., Ali, S. A., et al. (2017)). Still, the dominance of the Multi-Layer Perceptron model in the majority of the metrics, as well as the fact that its Precision is still high, being over 0.92 and its A* being on average over 0.75 still does render it a well-considered choice, especially if the aim is to manage the best possible accuracy, so the highest number of correct predictions are made compared to the total ones, or highest recall, which entails finding students who actually need extra attention. As another model that could be recommended in a general way, that would be the Logistic Regression model, as although its recall is rather low, being at 0.713 at average or 0.713-0.005=0.708 in the worst case, its values across all the other metrics are still good enough compared to the other models and especially its A* value is somewhat high, which means that the model's discriminatory capability for the two classes (of passing or failing the course) is quite decent.

Seeing that, unlike the K-fold cross validation results, the predictions at the end of the semester are at a somewhat satisfactory level as well as that, in order to intervene appropriately, it would be useful to determine how early it is possible to decently predict whether a student will pass or fail the course, in the next subsection, the demonstrated models will be tested with the data gathered from all the weeks up to a certain point.

3.8.2 Results with the weekly assignment assessment data up to the i-th week

In this part, the results of the 5 model's predictions of the student's two possible course pass states, using only the data up to a certain assignment are shown. As the models were tested after the 10-fold cross validation process, the 10-fold averages and the corresponding standard deviations were noted, similarly to how it was done in the above training/validation and testing processes.

The first model tested is the Logistic Regression.

Table 3-17: Logistic Regression's Predictions with Weekly Data up to the i-th Week

Week Metric	Logistic Regression Predictions with Assignment Assessment Data up to i-th Week - Weekly Assignment									
	1	2	3	4	5	6	7	8	9	10
Accuracy	0.638 (+/- 0.014)	0.673 (+/- 0.004)	0.728 (+/- 0.003)	0.714 (+/- 0.004)	0.738 (+/- 0.004)	0.735 (+/- 0.003)	0.734 (+/- 0.004)	0.736 (+/- 0.003)	0.733 (+/- 0.003)	0.735 (+/- 0.004)
Precision	0.925 (+/- 0.003)	0.944 (+/- 0.002)	0.939 (+/- 0.006)	0.97 (+/- 0.004)	0.951 (+/- 0.003)	0.947 (+/- 0.002)	0.94 (+/- 0.004)	0.947 (+/- 0.007)	0.946 (+/- 0.003)	0.948 (+/- 0.004)
Recall	0.585 (+/- 0.02)	0.646 (+/- 0.005)	0.691 (+/- 0.002)	0.671 (+/- 0.004)	0.714 (+/- 0.007)	0.713 (+/- 0.005)	0.718 (+/- 0.007)	0.715 (+/- 0.006)	0.712 (+/- 0.005)	0.713 (+/- 0.005)
F1 Metric	0.716 (+/- 0.015)	0.767 (+/- 0.003)	0.796 (+/- 0.002)	0.793 (+/- 0.003)	0.816 (+/- 0.004)	0.814 (+/- 0.003)	0.814 (+/- 0.004)	0.815 (+/- 0.003)	0.813 (+/- 0.003)	0.814 (+/- 0.003)
A* - AUC	0.707 (+/- 0.007)	0.728 (+/- 0.003)	0.771 (+/- 0.007)	0.788 (+/- 0.006)	0.778 (+/- 0.004)	0.771 (+/- 0.003)	0.76 (+/- 0.005)	0.77 (+/- 0.011)	0.768 (+/- 0.004)	0.772 (+/- 0.006)
Cohen's Kappa	0.28 (+/- 0.015)	0.279 (+/- 0.005)	0.414 (+/- 0.009)	0.375 (+/- 0.008)	0.394 (+/- 0.006)	0.384 (+/- 0.003)	0.374 (+/- 0.006)	0.384 (+/- 0.011)	0.38 (+/- 0.005)	0.385 (+/- 0.008)

The general performance pattern seems to be that of peaks and valleys and these are not formed in the same weeks across all the metrics. First of all, based on the metrics' values and especially on Kappa, the model's predictions are not accurate enough with the data up to the 1st and 2nd weeks, something that, if repeated in at least most of the other models, might indicate that, up to the 2nd week, it would be too early to make predictions about the students' course pass status. The Kappa values from the 3rd week up to the 10th are always above 0.3 even in the worst case, which means that it can be used to make predictions with the data up to any week from the 3rd until the semester's end. Regarding the weeks the best metric values were achieved, the best Kappa is for the

3rd week with a value of 0.414, the best Accuracy is found in the 5th week with 0.738, the best Precision is from the 4th with 0.97, the best Recall is from the 7th with 0.718, the best A* metric is from the 4th week with 0.788 and the best F1 is for the 5th with 0.816, although its worst value of 0.816-0.004=0.812 ties with the 8th week’s 0.815-0.003=0.812 which are on their own only marginally higher than the 0.814 of the weeks 6, 7, 10 and the 9th week’s 0.813. Since some of the metrics (Accuracy and F1) have their best values in the 5th week, this week could be a good choice for making the early predictions.

The second tested model is the Decision Tree.

Table 3-18: Decision Tree's Predictions with Weekly Data up to the i-th Week

Week	Decision Tree Predictions with Assignment Assessment Data up to i-th Week - Weekly Assignment									
Metric	1	2	3	4	5	6	7	8	9	10
Accuracy	0.647 (+/- 0.022)	0.696 (+/- 0.01)	0.667 (+/- 0.095)	0.697 (+/- 0.011)	0.723 (+/- 0.015)	0.74 (+/- 0.028)	0.757 (+/- 0.011)	0.743 (+/- 0.025)	0.746 (+/- 0.024)	0.756 (+/- 0.011)
Precision	0.892 (+/- 0.007)	0.918 (+/- 0.003)	0.9 (+/- 0.018)	0.923 (+/- 0.01)	0.914 (+/- 0.016)	0.92 (+/- 0.02)	0.915 (+/- 0.012)	0.919 (+/- 0.012)	0.918 (+/- 0.007)	0.921 (+/- 0.011)
Recall	0.624 (+/- 0.035)	0.697 (+/- 0.014)	0.636 (+/- 0.129)	0.687 (+/- 0.016)	0.728 (+/- 0.017)	0.745 (+/- 0.033)	0.773 (+/- 0.011)	0.75 (+/- 0.037)	0.754 (+/- 0.033)	0.765 (+/- 0.02)
F1 Metric	0.734 (+/- 0.023)	0.792 (+/- 0.009)	0.739 (+/- 0.101)	0.787 (+/- 0.01)	0.81 (+/- 0.011)	0.823 (+/- 0.022)	0.838 (+/- 0.008)	0.825 (+/- 0.021)	0.828 (+/- 0.02)	0.836 (+/- 0.009)
A* - AUC	0.676 (+/- 0.011)	0.693 (+/- 0.006)	0.704 (+/- 0.057)	0.714 (+/- 0.016)	0.715 (+/- 0.029)	0.732 (+/- 0.04)	0.731 (+/- 0.023)	0.731 (+/- 0.02)	0.731 (+/- 0.016)	0.74 (+/- 0.018)
Cohen’s Kappa	0.254 (+/- 0.022)	0.261 (+/- 0.01)	0.314 (+/- 0.11)	0.296 (+/- 0.02)	0.321 (+/- 0.038)	0.352 (+/- 0.056)	0.368 (+/- 0.032)	0.355 (+/- 0.033)	0.358 (+/- 0.033)	0.375 (+/- 0.021)

When examining the Kappa values of the Decision Tree, it can be seen that, like the Logistic Regression model, the data up to the 1st and 2nd weeks is characterized by Kappa values lower than 0.3, which supports that in those cases, it might be too early to make predictions. Unlike that model, though, the Decision Tree has a Kappa value less than 0.3 in the 4th week and when the worst case is taken into account, Kappa values larger than 0.3 are seen only from the 7th week and onwards. This indicates that the model might not be the most suitable for making early predictions. Regarding the weeks the best metric values were achieved, the best Kappa is for the 10th week with a value of 0.375, the best Accuracy is from the 7th week with 0.757, the best overall Precision is found in the 4th week with 0.923, while the best value in the weeks after the 6th is found in the 10th week with 0.921, the best Recall is from the 7th with 0.773, the best F1 metric is from the 7th week with 0.838 and the best A* metric is from the 10th week with 0.74.

The third model tested is the Support Vector Machine.

Table 3-19: Support Vector Machines's Predictions with Weekly Data up to the i-th Week

Week	Support Vector Machine Predictions with Assignment Assessment Data up to i-th Week - Weekly Assignment									
Metric	1	2	3	4	5	6	7	8	9	10
Accuracy	0.603 (+/- 0.002)	0.658 (+/- 0.004)	0.723 (+/- 0.004)	0.698 (+/- 0.005)	0.731 (+/- 0.004)	0.728 (+/- 0.005)	0.723 (+/- 0.006)	0.723 (+/- 0.007)	0.72 (+/- 0.006)	0.719 (+/- 0.009)
Precision	0.934 (+/- 0.003)	0.958 (+/- 0.0)	0.941 (+/- 0.007)	0.982 (+/- 0.002)	0.962 (+/- 0.002)	0.964 (+/- 0.006)	0.972 (+/- 0.004)	0.973 (+/- 0.003)	0.968 (+/- 0.004)	0.97 (+/- 0.005)
Recall	0.53 (+/- 0.004)	0.617 (+/- 0.005)	0.683 (+/- 0.003)	0.643 (+/- 0.006)	0.696 (+/- 0.005)	0.691 (+/- 0.008)	0.679 (+/- 0.009)	0.678 (+/- 0.009)	0.678 (+/- 0.008)	0.675 (+/- 0.009)
F1 Metric	0.676 (+/- 0.003)	0.75 (+/- 0.004)	0.791 (+/- 0.003)	0.777 (+/- 0.005)	0.808 (+/- 0.004)	0.805 (+/- 0.005)	0.799 (+/- 0.006)	0.799 (+/- 0.006)	0.798 (+/- 0.005)	0.796 (+/- 0.007)
A* - AUC	0.698 (+/- 0.002)	0.741 (+/- 0.003)	0.77 (+/- 0.009)	0.794 (+/- 0.004)	0.789 (+/- 0.004)	0.79 (+/- 0.008)	0.797 (+/- 0.006)	0.798 (+/- 0.005)	0.79 (+/- 0.005)	0.792 (+/- 0.01)
Cohen's Kappa	0.253 (+/- 0.003)	0.282 (+/- 0.005)	0.409 (+/- 0.012)	0.367 (+/- 0.007)	0.396 (+/- 0.007)	0.394 (+/- 0.009)	0.396 (+/- 0.008)	0.397 (+/- 0.009)	0.388 (+/- 0.008)	0.388 (+/- 0.015)

Like the previous two models, the Support Vector Machine also starts having more acceptable predictions from the 3rd week and onwards, as that is the point its Kappa values become larger than 0.3. Also, in a manner more similar to the Logistic Regression model compared to the Decision Tree, with the data up to the 3rd week and up to the 10th, the Kappa values never drop to something less than 0.3 neither in their averages, nor in the worst cases, thus based solely on that, the model can be used to make predictions early enough in the semester. The best metric values are again achieved in different weeks. The best Kappa is 0.409 from the 3rd week, the best Accuracy is 0.731 from the 5th week, the best Precision is 0.982 in the 4th week, the best Recall is 0.696 from the 5th week, the best F1 metric is 0.808 in the 5th week and the best A* metric is 0.798 in the 8th week. Like in the Logistic Regression model, the 3rd and 5th weeks have some of the best metric values for the model, which could entail that these weeks could be suitable for making early predictions.

The fourth model tested is the Sequential Neural Network.

Table 3-20: Neural Network's Predictions with Weekly Data up to the i-th Week

Week	Neural Network Predictions with Assignment Assessment Data up to i-th Week - Weekly Assignment									
Metric	1	2	3	4	5	6	7	8	9	10
Accuracy	0.592 (+/- 0.081)	0.658 (+/- 0.013)	0.696 (+/- 0.022)	0.641 (+/- 0.044)	0.725 (+/- 0.016)	0.678 (+/- 0.068)	0.699 (+/- 0.043)	0.696 (+/- 0.066)	0.727 (+/- 0.013)	0.726 (+/- 0.018)
Precision	0.919 (+/- 0.035)	0.952 (+/- 0.008)	0.956 (+/- 0.016)	0.99 (+/- 0.004)	0.969 (+/- 0.009)	0.97 (+/- 0.009)	0.962 (+/- 0.012)	0.959 (+/- 0.01)	0.949 (+/- 0.007)	0.961 (+/- 0.01)
Recall	0.522 (+/- 0.103)	0.621 (+/- 0.02)	0.635 (+/- 0.039)	0.567 (+/- 0.055)	0.683 (+/- 0.025)	0.623 (+/- 0.088)	0.656 (+/- 0.061)	0.654 (+/- 0.086)	0.702 (+/- 0.018)	0.692 (+/- 0.03)
F1 Metric	0.661 (+/- 0.104)	0.751 (+/- 0.013)	0.762 (+/- 0.025)	0.719 (+/- 0.046)	0.801 (+/- 0.015)	0.755 (+/- 0.07)	0.778 (+/- 0.041)	0.775 (+/- 0.068)	0.807 (+/- 0.011)	0.804 (+/- 0.017)
A* - AUC	0.683 (+/-	0.733 (+/-	0.768 (+/-	0.77 (+/-	0.794 (+/-	0.769 (+/-	0.771 (+/-	0.767 (+/-	0.769 (+/-	0.784 (+/-

	0.055)	0.01)	0.011)	0.025)	0.009)	0.038)	0.019)	0.035)	0.011)	0.009)
Cohen's Kappa	0.24 (+/- 0.075)	0.275 (+/- 0.012)	0.386 (+/- 0.021)	0.312 (+/- 0.044)	0.395 (+/- 0.016)	0.344 (+/- 0.071)	0.357 (+/- 0.044)	0.355 (+/- 0.066)	0.376 (+/- 0.018)	0.388 (+/- 0.017)

Like with the previous models, the Kappa values from the data up to the 1st and 2nd weeks indicate that these are indeed too early for sufficient predictions to be made. But while the Neural Network manages average Kappa values over 0.3 for the data up to the 3rd week and afterwards, the worst case Kappa values in the weeks up to 4, 6 and 8 are 0.312-0.044=0.268, 0.344-0.071=0.273 and 0.355-0.066=0.289 respectively, which are all lower than 0.3. This indicates either that the model's early predictions should be made in general with caution or that they should be only made for the weeks 3, 5, 7, 9 or 10, where the Kappa values including the worst case are always over 0.3. Then, determining which weeks manage the best metric values, it can be seen that the best Kappa is 0.395 and is achieved in the 5th week, the best Accuracy is 0.727 in the 9th week, the best overall Precision is 0.990 in the 4th week or among the weeks with Kappa exclusively over 0.3, it is 0.969 in the 5th, the best Recall is 0.702 from the 9th week, the best F1 metric is 0.807 in the 9th week and the best A* metric is 0.794 in the 5th week. According to the above, the choice of making the early predictions in the 5th and possibly the 3rd weeks still seems decent, even though half of this model's metrics (Accuracy, Recall and F1) have their highest values in the 9th week.

The fifth model tested is the Multi-Layer Perceptron.

Table 3-21: Multi-Layer Perceptron's Predictions with Weekly Data up to the i-th Week

Week	Multi-Layer Perceptron Predictions with Assignment Assessment Data up to i-th Week - Weekly Assignment									
Metric	1	2	3	4	5	6	7	8	9	10
Accuracy	0.621 (+/- 0.02)	0.69 (+/- 0.01)	0.739 (+/- 0.005)	0.733 (+/- 0.008)	0.764 (+/- 0.005)	0.766 (+/- 0.007)	0.757 (+/- 0.012)	0.764 (+/- 0.011)	0.767 (+/- 0.009)	0.766 (+/- 0.007)
Precision	0.934 (+/- 0.005)	0.938 (+/- 0.003)	0.937 (+/- 0.01)	0.961 (+/- 0.009)	0.944 (+/- 0.007)	0.943 (+/- 0.01)	0.934 (+/- 0.009)	0.927 (+/- 0.006)	0.93 (+/- 0.01)	0.927 (+/- 0.008)
Recall	0.555 (+/- 0.029)	0.673 (+/- 0.015)	0.708 (+/- 0.008)	0.703 (+/- 0.016)	0.755 (+/- 0.007)	0.758 (+/- 0.013)	0.755 (+/- 0.019)	0.769 (+/- 0.014)	0.771 (+/- 0.014)	0.773 (+/- 0.012)
F1 Metric	0.696 (+/- 0.023)	0.783 (+/- 0.009)	0.807 (+/- 0.004)	0.812 (+/- 0.008)	0.839 (+/- 0.004)	0.84 (+/- 0.006)	0.835 (+/- 0.01)	0.841 (+/- 0.008)	0.843 (+/- 0.007)	0.843 (+/- 0.006)
A* - AUC	0.707 (+/- 0.009)	0.726 (+/- 0.002)	0.775 (+/- 0.011)	0.786 (+/- 0.01)	0.78 (+/- 0.012)	0.78 (+/- 0.017)	0.761 (+/- 0.013)	0.754 (+/- 0.012)	0.759 (+/- 0.018)	0.754 (+/- 0.014)
Cohen's Kappa	0.272 (+/- 0.018)	0.289 (+/- 0.007)	0.427 (+/- 0.014)	0.39 (+/- 0.007)	0.42 (+/- 0.014)	0.421 (+/- 0.019)	0.397 (+/- 0.018)	0.396 (+/- 0.02)	0.404 (+/- 0.022)	0.398 (+/- 0.017)

This model's Kappa value patterns are more similar to Logistic Regression and Support Vector Machine, rather than the Decision Tree or the Neural Network. The only cases where the Kappa values are less than 0.3 are with the data up to the 1st and 2nd weekly assignments, meaning that more accurate predictions can only be made after having available the data from the 3rd weekly assignment, as all the Kappa values from this point and afterwards are above 0.3. Indeed, as all the models have shown, at least with the steps taken to train them, making predictions with the data up to the 1st or 2nd assignment seems very early. Regarding which weeks have the best metrics, the best Kappa is 0.427 and is achieved in the 3rd week, the Precision is 0.961 in the 4th week, the best Recall is 0.773 from the 10th week, the best F1 metric is 0.843 in the 10th week, the best A* metric is 0.786 in the 4th week and the best Accuracy is 0.767 from the 9th week in the average case or 0.759 in the worst case which is achieved either by 0.766-0.007=0.759 in the weeks 6 and 10, or from 0.764-0.005=0.759 from the 5th week.

Now that the model's predictive capabilities have been established, it would be useful to compare the performances they achieve with the data up to specific weeks. As 10 weeks are probably an excessive amount, the comparisons will be made for data up to the weeks 3 and 5, as mentioned above. As the Decision Tree classifier manages Kappa values lower than 0.3 in the worst case for the data up to these weeks, it will not be included among the different classifiers of the comparison.

3.8.3 Predictions with Weekly Assignment Data up to the 3rd week

Since most of the models begin to exhibit Kappa value results over 30% with the usage of the data from the point of the 3rd weekly assignment and since this point in time is close to the beginning of the semester, it is the chosen point for making very early predictions.

Table 3-22: Model Predictions with Data up to the 3rd Weekly Assignment

Model Metric	Logistic Regression	Support Vector Machine	Sequential Neural Network	Multi-Layer Perceptron
Accuracy	0.728 (+/- 0.003)	0.723 (+/- 0.004)	0.696 (+/- 0.022)	0.739 (+/- 0.005)
Precision	0.939 (+/- 0.006)	0.941 (+/- 0.007)	0.956 (+/- 0.016)	0.937 (+/- 0.01)
Recall	0.691 (+/- 0.002)	0.683 (+/- 0.003)	0.635 (+/- 0.039)	0.708 (+/- 0.008)
F1 Metric	0.796 (+/- 0.002)	0.791 (+/- 0.003)	0.762 (+/- 0.025)	0.807 (+/- 0.004)
A* - AUC	0.771 (+/- 0.007)	0.77 (+/- 0.009)	0.768 (+/- 0.011)	0.775 (+/- 0.011)
Cohen's Kappa	0.414 (+/- 0.009)	0.409 (+/- 0.012)	0.386 (+/- 0.021)	0.427 (+/- 0.014)

From the comparison of the different metric values with the data up to the 3rd week, it can be seen that in almost all of them, the Multi-Layer Perceptron classifier achieves the best results. Apart from Precision, where the Multi-Layer Perceptron has the worst value among all the models, as well as the worst-case for A* where the 0.775-0.011=0.764 ties with Logistic Regression's 0.771-0.007=0.764 its performance visibly surpasses all the models. Multi-Layer Perceptron's Precision on its own, though, is still fairly high and not far from the other models, which might be the trade-off for managing the best Recall among all the models, which also leads to a visibly higher F1 metric value, with it being the only model with a value higher than 0.8 including even the worst case (0.807-0.004=0.803). Regarding the descending order the models are sorted according to the different metrics, it is observed that, according to all metrics excluding Precision, the order becomes MLP>LR>SVM>NN or in the worst case for the A* metric MLP=LR>SVM>NN, while according to Precision the order is reversed, being NN>SVM>LR>MLP. As the case for Precision's metrics has already been explained, it could be argued that the order the models are sorted in according to the majority of the chosen metrics reflects their predictive capabilities with the data up to the 3rd weekly assignment. The overall metric values achieved at this rather early point in the semester are not generally very low, but Accuracy and especially Recall could be improved.

If the best metric values of the predictions made with data up to the 3rd week, are compared to those up to the 10th, in most cases the results are slightly worse. Regarding Accuracy, the best value from the data up to the 3rd week is managed by the Multi-Layer Perceptron and it is 0.739, while up to the 10th it is managed by the same model and it is 0.766, so as 0.739 < 0.766, the value has seen an improvement with the addition of more data. The best Precision in week 3 is managed by the Neural Network with 0.956, while in week 10 by the Support Vector Machine, which is 0.97, higher than the 0.956 of before. Also, in week 10, the Neural Network achieves Accuracy 0.961 which is also higher than the same model's 0.956 in week 3. The best Recall in week 3 is managed by the Multi-Layer Perceptron with 0.708, while in week 10 also by the same model, with the higher value of 0.773. The best F1 value in week 3 is managed by the Multi-Layer Perceptron with 0.807, while in week 10 also by the same model, with the higher value of 0.843. The best A* of week 3 is achieved by the Multi-Layer Perceptron with 0.775, while in week 10 by the Support Vector Machine, with 0.792. Moreover, in week 10, the Multi-Layer Perceptron achieves an A* of 0.754 which is much lower than the same model’s 0.775 in week 3. Lastly, the best Kappa metric value from week 3 is Multi-Layer Perceptron’s 0.427, while in week 10 it is the 0.398 from the same model, which is a visibly lower value than that of the 10th week, unlike in the previous examined cases.

3.8.4 Predictions with Weekly Assignment Data up to the 5th week

The 5th week represents more or less the middle of the semester, which is the reason it is chosen as the point where the timely predictions are made.

Table 3-23: Model Predictions with Data up to the 5th Weekly Assignment

Model Metric	Logistic Regression	Support Vector Machine	Sequential Neural Network	Multi-Layer Perceptron
Accuracy	0.738 (+/- 0.004)	0.731 (+/- 0.004)	0.725 (+/- 0.016)	0.764 (+/- 0.005)
Precision	0.951 (+/- 0.003)	0.962 (+/- 0.002)	0.969 (+/- 0.009)	0.944 (+/- 0.007)
Recall	0.714 (+/- 0.007)	0.696 (+/- 0.005)	0.683 (+/- 0.025)	0.755 (+/- 0.007)
F1 Metric	0.816 (+/- 0.004)	0.808 (+/- 0.004)	0.801 (+/- 0.015)	0.839 (+/- 0.004)
A* - AUC	0.778 (+/- 0.004)	0.789 (+/- 0.004)	0.794 (+/- 0.009)	0.78 (+/- 0.012)
Cohen’s Kappa	0.394 (+/- 0.006)	0.396 (+/- 0.007)	0.395 (+/- 0.016)	0.42 (+/- 0.014)

Like with the case of the predictions up to the 3rd week, most of the metrics of those up to the 5th also indicate the Multi-Layer Perceptron as the model with the best predictive performance. Said metrics are Accuracy, Recall, F1 and Kappa. The Multi-Layer Perceptron’s Precision is the worst among all the models tested, although the value is still fairly high, while its A* score is either the second worst in the average case, as 0.78 > 0.778 of the Logistic Regression, or the worst in the worst case as 0.78-0.012=0.768 < 0.778-0.004=0.774 from the Logistic Regression. The descending order the models are sorted according to Accuracy, Recall and F1 is MLP>LR>SVM>NN, according to Precision and the worst case for A* it becomes reversed, being NN>SVM>LR>MLP, according to the average case for A* it the last two models swap places, becoming NN>SVM>MLP>LR, while according to the average Kappa it is MLP>SVM>NN>LR and according to the worst case Kappa the last two also swap places, being MLP>SVM>LR>NN. When compared to the results of the 3rd week, the metrics have generally been improved, apart from Kappa which has seen a not very large but visible drop. The Accuracy and especially the Recall metric values still continue to not be very high, but focusing on those of the Multi-Layer Perceptron, they are acceptable enough.

When the results of the predictions with data up to the 5th week are compared to those of the 3rd and 10th weeks, they tend to be between the other two, though not always. The best Accuracy in week 5 is achieved by the Multi-Layer Perceptron with 0.764, in week 3 by the same model and it is 0.739, while in week 10 also by the same model and it is 0.766. The 5th week’s 0.764 is ever so slightly lower than the 10th's 0.766 but higher than the 3rd’s 0.739. Interestingly, in Accuracy’s worst case, the values for both of the weeks 5 and 10 become the same, being 0.764-0.005=0.759 and 0.766-0.007=0.759 respectively. The best Precision metric value from the 5th week is 0.969 from the Neural Network, in the 3rd week it is 0.956 from the same model, which is a lower value, while from the 10th week it is 0.97 from the Support Vector Machine, which is a marginally higher value than the 5th week’s 0.969. So, the best Precision metric values from the 3rd, 5th and 10th weeks are 0.956<0.969<0.97 respectively. Seeing that the Neural Network’s Precision from the 10th week is 0.961, comparing the values of only this model, the weeks in descending order become 5>10>3 because 0.969>0.961>0.956. Regarding Recall, in all weeks the best values come from the Mutli-Layer Perceptron model and since the values in ascending order are 0.708<0.755<0.773, the weeks corresponding to these values are 3<5<10. The best F1 metric values from all weeks also come from the Mutli-Layer Perceptron model and as the values in ascending order are 0.807<0.839<0.843, the weeks corresponding to these values are 3<5<10. Then, the highest A* values are actually managed by different models, as in the 3rd week it is 0.775 from the Multi-Layer Perceptron, in the 5th it is 0.794 from the Neural Network, while in the 10th, it is 0.792 from the Support Vector Machine, thus the ascending order of the weeks becomes 3>10>5, as 0.775>0.792>0.794. Remaining only at the Neural Network’s values among the different weeks, in the 5th it is 0.794 as already mentioned, in the 3rd it is 0.768 and in 10th it is 0.784, thus specifically for this model, which manages the best results in the examined 5th week, the ascending week order also is 3>10>5, as 0.768>0.784>0.794. Finally, the best Kappa values are all achieved by the Multi-Layer Perceptron model across all these more thoroughly examined weeks, as in week 5 it manages a value of 0.427, in week 3 it manages 0.42 which is lower than the 5th’s, while in week 10 it manages 0.398, which is the lowest among the examined weeks, thus their order becomes 5>3>10, as 0.427>0.42>0.398.

4 Conclusion

This study analyzed the data from the student's weekly coding assignments from an introductory programming course in a per gender manner. It firstly introduced the readers to the topics this dissertation studies and afterwards it reviewed other studies belonging in its broader field. Then, it explains the data sources used as well as the different types of analysis the data was subject to. Following that, the results of the data analysis procedure were shown in a detailed manner. Finally, some concluding remarks are made, including the discussion on the results, the limitations faced, in addition to what could be done in possible future additions.

4.1 Summary and conclusions

This study analyzed the data from the students' weekly assignments from an introductory programming course in a per gender manner, utilized said data for clustering and association rule creation in order to uncover hidden information for the course as well as its students and also used it to make predictions about whether a student will be able to pass the course. In the following paragraphs, the findings of the study are briefly summarized according to the Research Question they answer to, while they are also compared to the results from other broadly related works.

RQ1: *Do students' errors and academic performance in programming differ based on gender and if so, how?*

In this work, the results showed that women may possibly tend to make a few more compilation errors than their male counterparts, but they may also possess the advantage in terms of academic achievement. The statistical tests shown indicate that differences rarely ever exist in either case, but also that women had the advantage on those specific situations, whenever they occurred. The results of Pillay, N. and Jugoo, V. R. (2005) on a "Procedural Programming" course indicate that the two genders have no statistically significant differences in terms of academic performance, which is also the finding of this study, but there, male students were the ones who seemed to achieve higher performance, unlike in this case.

RQ2: *Can the association rules and clustering aid in better understanding the relationships formed by attributes within the course?*

Starting with the clustering findings, the two procedures followed aimed to determine how the students can be clustered and if the results seem close to any of the "ground truth" clusters, created when examining the data per year, per gender or for whether the students passed or not. In the first attempt, all 3 algorithms used indicated 4 clusters as the optimal number, though they do not seem to correspond particularly well to any of the examined ground truths. In the second attempt only 2 are created, again from all the algorithms, which mainly seem to align with the course pass results though with some influence from the gender ground-truth.

Regarding the association rule mining, the more detailed goal is to find which are the factors that contribute to the students' passing the course, as well as whether the student's gender does seem to predetermine the result in any way. The completion of certain assignments indeed aids the students in passing the course, which can be considered as a comparable finding to (Höök, L. J., & Eckerdal, A. (2015, April)), since it entails actively engaging with the act of programming. Then, continuing with these aiding factors, either the students being enrolled in the course for their first time or attending it at the semester the department's undergraduate studies syllabus dictates also has a certain role. About the students' gender, it seems that women are slightly "predetermined" to pass the course, while men to not manage this, although among the course's more active participants, gender does not seem to be nearly as important.

Now, seeing how the association rules' results compare with the existing studies' ones, beginning with those of Damaševičius R. (2010), while the course studied is not the exact same, being "Object Oriented Programming" and not "Data Structures" like this study, there it is found that the contents of the instruction and lab sessions regarding the subjects of "Linear Dynamic Lists" are indicated to cause high student failure by many of the used metrics. Thought, it should be noted that, in said study, emphasis is given in finding the causes that result in the students not being able to pass the course. In this study, a seemingly similar subject regarding "Lists - Dynamic Stack - Queue" is among the assignment subjects aiding the students in passing the course successfully. Then, while the data of Matetic, M., Bakaric, M. B. and Sisovic, S. (2015, June) is not quite the exact same as of this study, the general finding of that study is similar to this one. In that study, which uses more data regarding access to different course materials instead of being based mainly on assignments and more general student statistics like this one, the authors' findings are quite comparable to those of this study. More precisely, they found that engagement with course's video lectures is positively associated with passing the course, while not enough engagement with self-assessments is associated with failing it. In this study, even though material access times are not included, similar ideas can be inferred by the fact that engagement with the course's assignments is indeed highly associated with passing the course, while the lack of the two enrollment factors explained previously, with the idea of failing it. While the connection here it much more general, and points to the rather obvious conclusion that engagement with a course should lead to much higher chance of passing it, the fact that this is confirmed does show that the respective studies' courses do not suffer from factors that can lead to truly excessive student failure. Very similar general remarks can also be made when the results of this study are compared with those of (Ayub, M., Toba, H., Yong, S., et al. (2017)) where it was found that higher LMS activity could translate in the student at least passing the course, possibly even excellently. They do point out, though, the case of having excellent grade with low LMS activity, something that can happen if, for example, the student downloaded the instruction material.

RQ3: *Can a student's final grade category be predicted before the exams or, preferably even earlier in the semester by using only the number of their compilation errors and their average assignment grade?*

On the subject of the work's third Research Question, it turned out that this prediction is indeed possible to be realized somewhat decently by almost the first third of the semester and even more so by the middle of the semester and at its end. So, it can be done at least in the weeks 3, 5 and 10 from a total of 10. But since the models and their results have been already written in more detail, it would be more useful to compare them to the results of other similar studies.

In the study of Koprinska I., Stretton J. and Yacef, K. (2015), a Decision Tree was trained with student data about their assessment and assignment tasks, as well as their online community activity in Piazza. Their goal is the prediction of whether the student will just pass the course's final exams, pass the final exams with a high grade or fail it entirely, which they achieved with an accuracy of 66.52% in the middle of the semester or 72.69% in its end. In this study, the aim is only the prediction of whether the student will pass the course or fail, not also whether they will pass with a high grade. The best Accuracy from the middle of the semester was achieved by the Multi-layer Perceptron model, with a percentage of 76.4%, which is better than both the corresponding study's middle of the semester result (66.52%) and the results of the end of the semester (72.69%), while this study's end of the semester result from the same model is 76.6% which is also higher than the target study's values. Then again, here only 2 classes are possible instead of 3, so this could explain the difference.

Another study with the aim of predicting student performance is that of Mueen, A., Zafar, B. and Manzoor, U. (2016), where the data used is significantly more varied, containing demographic elements and more specific learning capabilities for the students, information about online discussion activity and grades both from the course as well as their Grade Point Average. Three classifier models from WEKA were trained, being Naïve Bayes, Multi-layer Perceptron and Decision Tree, once with all the features and once with the 7 considered the best for prediction. In both feature choice cases, Naïve Bayes managed the best Accuracy, Precision and Specificity metrics, while the Multi-layer Perceptron managed the best Recall. In this study, Specificity is not used. When the rest of the results are compared, they are substantially lower than the ones of said Weka-based study. With data gathered by the semester's end, this study's model with the best Accuracy is Multi-layer Perceptron, which manages 76.6%, which is far lower than the Weka study's 86% with Naïve Bayes, a Precision of 97% with the Support Vector Machine classifier which is much higher than the target study's 89.3% with the Naïve Bayes classifier, while the best Recall is achieved by the Multi-layer Perceptron model with a value of 77.3% which is substantially lower that the target study's Multi-layer Perceptron with 86.5%. In general, the target study's models do manage much better results, while they also manage a comparatively smaller divide between Precision and Recall, but this study's models use significantly less data and can actually work much earlier in the semester, during its middle or slightly before its first third, instead of the target study's end of it.

Prediction of students' performance was also done in (Kostopoulos, G., Lipitakis, A. D., Kotsiantis, S., et al. (2017)), the features used for making predictions include students' more demographic-based data, attendance at the instruction sessions and grades in "written assignments". The models used involve Decision Trees (J48 and JRip), Naïve Bayes, Logistic Regression, Support Vector Machine through Sequential Minimal Optimization and Multi-layer Perceptron. Since they provide predictions for different points in the duration of the course, including the end of the semester and its middle, the comparisons are going to be made firstly with the results of the middle of the semester and then with its end. In the middle of the semester, the target study’s best Accuracy value is achieved with Sequential Minimal Optimization and is 75.54%, which is slightly lower than the best of this study’s 76.4%, achieved by the Multi-layer Perceptron, although lower than the corresponding value of this study’s Support Vector Machine model, since 73.1% < 75.54%. In contrast, the target study’s Multi-layer Perceptron achieved an Accuracy level of 72.97% which is lower than this study’s Multi-Layer Perceptron value of 76.4%. The target study’s Logistic Regression achieved an Accuracy level of 72.09% which is lower than this study’s 73.8% with the same model. Then, Sequential Minimal Optimization’s Area Under the Curve or A* value in the same timeframe is 71.758%, which is lower than this study’s Support Vector Machine model, being at 78.9%. Now, focusing on the end of the semester, the target study’s best Accuracy value, is achieved with Sequential Minimal Optimization which it is 80.82%, a value higher than this study’s 76.6%, achieved by the Multi-layer Perceptron, thus also higher than the corresponding value of this study’s Support Vector Machine model of 71.9%. The target study’s Multi-layer Perceptron achieved an Accuracy level of 81.09% which is also visibly higher than this study’s Multi-Layer Perceptron value of 76.4%. The target study’s Logistic Regression achieved an Accuracy level of 79.34% which is much higher than this study’s 73.5% with the same model. Then, Sequential Minimal Optimization’s Area Under the Curve or A* value in the end of the semester is 78.425%, which is a little lower than this study’s Support Vector Machine being at 79%. In general, the models from the target study continue to improve with the passing of the weeks, while the models of this study seem to plateau once they reach a point towards or slightly after the middle of the semester.

The last study from the “Literature Review” Section the results can be compared to this one’s quite directly is that of Bucos, M. and Drăgulescu, B. (2018), which uses data from instruction environments with physical presence and, after feature selection, contains information like students’ presences in the instruction sessions, assignment and activity grade averages, and also some additional elements like the students’ “number of credit points from the previous year”. Said study uses Python for the creation of its models, which are the Logistic Regression, Decision Tree and “Extra Tree” Classifiers, Random Forest and Support Vector Classifier, while from the metrics used, being Recall, Area Under the Curve or A*, F1 and Specificity, the first three are also used in this study. The data is available for the Weeks 6, 8 and 12 and, as the 6th week is more or less the middle of the semester and the 12th is its end, this is where the comparisons with the current study will be made. Also, since the 2nd result table in the target study uses a specific year for the result prediction, the 1st result table will be used for the comparisons, as they are more in-line with this study's combination of the data from both years in training/validation and testing alike. In the middle of the semester, the target study's best Recall was achieved by the Random Forest algorithm, with 85%, while this study's best Recall is achieved by the Multi-Layer Perceptron and it is 75.5%, which is a far lower value. Next, the target study's best Accuracy was achieved by the Random Forest algorithm, with 79%, while this study's best Accuracy is achieved by the Multi-layer Perceptron and it is 76.4%, which is much closer compared to the Recall values and it actually surpasses the Accuracy values of the other models from the target study (which are 76% or 74%). Then, the target study's best Area Under the Curve metric was achieved by both the Logistic Regression and Support Vector Classifier algorithms, with 85%, while this study's best Area Under the Curve score is also comparatively lower and it is achieved by the Sequential Neural Network classifier and it is 79.4%. This study's Support Vector Machine classifier, on average, achieves the slightly lower AUC value of 78.9%. Lastly, the target study's best F1 score was achieved by the Random Forest algorithm, with 85%, while this study's best corresponding score is achieved by the Multi-layer Perceptron and it is 83.9%. Now, in the end of the semester, the target study's best Recall was achieved by the Decision Tree, Extra Trees and Random Forest classifiers, with 87%, while this study's best Recall is achieved by the Multi-Layer Perceptron and it is 77.3%, which is a significantly lower value. Next, the target study's best Accuracy was achieved by the Logistic Regression model, with 86%, while this study's best Accuracy is achieved by the Multi-layer Perceptron and it is 76.6%, which is even further compared to the Recall values. This study's Logistic Regression model's Accuracy is the even lower 73.5%. Then, the target study's best Area Under the Curve metric was achieved by the Logistic Regression algorithm with 94%, while this study's best Area Under the Curve score is also comparatively lower, it is achieved by the Support Vector Machine classifier and it is 79.2%. The Logistic Regression model of this study managed only a lower A* value of 77.2% Lastly, the target study's best F1 score was achieved by the Random Forest and Logistic Regression classifiers with 89%, while this study's best corresponding score is achieved by the Multi-layer Perceptron and it is 84.3%. This study's Logistic Regression managed a value of 81.4%. All the above indicate that the study of (2018) manages better results than the current one, significantly so towards the end of the semester, though the fact that the data is more detailed than that of the current study's probably does aid the results, when compared to the current one.

In the end, regarding the predictive capabilities of this study's models, their overall performance level ranges from being generally a bit better to quite worse when compared to similar studies, but unlike them, the data used comes from only one source and only two of its features are used. Even with this, since the predictive capabilities of the models do not always improve much from around the middle point of the total of 10 weeks, or might even become slightly worse, it is very probable that the compilation errors determined in students' assignment code are not the absolute best measure of predicting their performance, at least regarding the chosen "Data Structures" course. Something similar can probably also be inferred from the absence of any types of compilation errors from the association rules that lead to a student not passing the course.

Lastly, as some additional last remarks for the study's 1st Research Question, if one were to summarize the results of the per gender analysis, it seems that, based on the course's data, the two gender's performance is broadly similar. Even though it is not possible to definitively accept or reject the notion of an actual performance advantage by any gender in this case, if it does exist, it would almost certainly belong to the female students. Having this as a basis, if any actions were to be taken to reduce the gender gap in the field in general or in the studied department in particular, emphasis should be given in the findings regarding the performance of the two genders and how they compare with what is expected by the usual stereotypes. They provide empirical evidence in the direction that, the views regarding preferences for specific jobs by specific genders ought to be reevaluated.

4.2 Research limits and limitations

The first limitation of this research is that only one course is being studied and only from one department. Ideally, in order to gain more representative results, both the number of introductory programming courses and the number of the universities or other educational institutions the target courses are being taught at should be higher.

Then, regarding the data that was available for the course, it could be argued that another limitation is the amount of data available for the studied students. Although one of the goals was the simplicity of the data collection procedure, so that it would be limited to what can be extracted by the students' assignment code after its submission, it is true that in other studies more data is available concerning the students' demographic characteristics, their interaction with the educative materials, their presence in the instruction sessions and their coding characteristics. It could also be argued that access to the original student's code files instead of just their assessment results could be helpful, but it could prove more concerning from a privacy and personal data angle.

4.3 Future extensions

Regarding the additions that could be made to the study or what could be changed if it were to be redone with a slightly expanded scope, there are definitely some ideas that could be realized. Firstly, it would be to also study the data from the course that could be considered its logical prerequisite, the "Procedural Programming" of the program's first semester. Since it turned out that the students in this part were generally capable of managing their errors, at least based on their absence from the rules regarding the student not passing the course, maybe this could indicate which errors are causing the highest difficulties for the students. Also, in that course, it would be interesting to test if the feature of the students' compilation errors, warnings and notes can work well enough as a factor for predicting their academic performance.

Then, if courses about the learning of programming languages other than C were to be studied alongside these from the point of view of the comparison per gender, the introductory courses for Java/C++, web development or android could also be included. That could shed additional light on the point of the two gender's error making and their programming academic performance.

About the production of association rules, a second attempt on a higher end computing machine could be preferable, as it would allow for the production of a higher number of rules with lower thresholds. Alternatively, if the Association Rules were to be created by a completely different environment outside of Python, an algorithm finding the Least Association Rules could be attempted.

Moreover, if the environment chosen to conduct the experiments was different, clustering with Expectation Maximization could have been attempted.

Lastly, if more prediction methods were to be applied within a different experiment environment, it would be interesting to determine if algorithms in the category of Semi-Supervised Learning could produce better results.

References

A.1 Bibliography

Russel, J. S., Norvig P. & Ρεφανίδης Ι. (2005). *Τεχνητή Νοημοσύνη Μία σύγχρονη προσέγγιση Δεύτερη Αμερικάνικη Έκδοση*. Αθήνα: Εκδόσεις “Κλειδάριθμος” p.32 «Εισαγωγή»

Haykin, S. & Ε. Γκαγκάτσιου (2010). *Νευρωνικά Δίκτυα και Μηχανική Μάθηση Τρίτη Έκδοση*. Αθήνα: Εκδόσεις Παπασωτηρίου p.2 «Εισαγωγή»

Suthaharan, S., & Suthaharan, S. (2016). Support vector machine. Machine learning models and algorithms for big data classification: thinking with examples for effective learning, 207-235. pp.39-40, 42

Zafarani, R., Abbasi, M. A., & Liu, H. (2014). Social media mining: an introduction. Cambridge University Press. Available at: http://socialmediamining.info/ (May 25th 2023) pp.188-192 & p.207

A.2 Journal Articles

de Carvalho, C. V., Cerar, Š., Rugelj, J., Tsalapatas, H., & Heidmann, O. (2020). Addressing the gender gap in computer programming through the design and development of serious games. IEEE Revista Iberoamericana de Tecnologias del Aprendizaje, 15(3), 242-251.

Forrester, C., Schwikert, S., Foster, J., & Corwin, L. (2022). Undergraduate R programming anxiety in ecology: Persistent gender gaps and coping strategies. CBE—Life Sciences Education, 21(2), ar29.

Werth, L. H. (1986). Predicting student performance in a beginning computer science class. ACM SIGCSE Bulletin, 18(1), 138-143.

Pillay, N., & Jugoo, V. R. (2005). An investigation into student characteristics affecting novice programming performance. ACM Sigcse Bulletin, 37(4), 107-110.

Islam, N., Shafi Sheikh, G., Fatima, R., & Alvi, F. (2019). A study of difficulties of students in learning programming. Journal of Education & Social Sciences, 7(2), 38-46.

ElGamal, A. F. (2013). An educational data mining model for predicting student performance in programming course. International journal of computer applications, 70(17), 22-28.

Mueen, A., Zafar, B., & Manzoor, U. (2016). Modeling and predicting students' academic performance using data mining techniques. International Journal of Modern Education and Computer Science, 8(11), 36.

Bucos, M., & Drăgulescu, B. (2018). Predicting student success using data generated in traditional educational environments. TEM Journal, 7(3), 617.

Damaševičius, R. (2010). Analysis of academic results for informatics course improvement using association rule mining. Information Systems Development: Towards a Service Provision Society, 357-363.

Ayub, M., Toba, H., Yong, S., & Wijanto, M. C. (2017). Modelling students' activities in programming subjects through educational data mining. Global Journal of Engineering Education, 19(3), 249-255.

Bruha, I., & Famili, A. (2000). Postprocessing in machine learning and data mining. ACM SIGKDD Explorations Newsletter, 2(2), 110-114.

Şahİn, M., & Yurdugül, H. (2020). Educational data mining and learning analytics: past, present and future. Bartın University Journal of Faculty of Education, 9(1), 121-131.

Liñán, L. C., & Pérez, Á. A. J. (2015). Educational Data Mining and Learning Analytics: differences, similarities, and time evolution. RUSC. Universities and Knowledge Society Journal, 12(3), 98-112.

Romero, C., & Ventura, S. (2020). Educational data mining and learning analytics: An updated survey. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 10(3), e1355.

Popenici, S. A., & Kerr, S. (2017). Exploring the impact of artificial intelligence on teaching and learning in higher education. Research and Practice in Technology Enhanced Learning, 12(1), 1-13.

Alzubi, J., Nayyar, A., & Kumar, A. (2018, November). Machine learning from theory to algorithms: an overview. In Journal of physics: conference series (Vol. 1142, p. 012012). IOP Publishing.

Teng, X., & Gong, Y. (2018, July). Research on application of machine learning in data mining. In IOP conference series: materials science and engineering (Vol. 392, No. 6, p. 062202). IOP Publishing.

Guenther, N., & Schonlau, M. (2016). Support vector machines. The Stata Journal, 16(4), 917-937.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. Neural computation, 9(8), 1735-1780.

Patrício, M., Ferreira, F., Oliveiros, B., & Caramelo, F. (2017). Comparing the performance of normality tests with ROC analysis and confidence intervals. Communications in Statistics-Simulation and Computation, 46(10), 7535-7551.

Razali, N. M., & Wah, Y. B. (2011). Power comparisons of shapiro-wilk, kolmogorov-smirnov, lilliefors and anderson-darling tests. Journal of statistical modeling and analytics, 2(1), 21-33.

Oppong, F. B., & Agbedra, S. Y. (2016). Assessing univariate and multivariate normality. a guide for non-statisticians. Mathematical theory and modeling, 6(2), 26-33.

McGee, M. (2018). Case for omitting tied observations in the two-sample t-test and the Wilcoxon-Mann-Whitney Test. PloS one, 13(7), e0200837.

Milenović, Ž. (2011). Application of Mann-Whitney U test in research of professional training of primary school teachers. Metodički obzori: časopis za odgojno-obrazovnu teoriju i praksu, 6(11), 73-79.

Abdullah, Z., Herawan, T., Ahmad, N., & Deris, M. M. (2011). Mining significant association rules from educational data using critical relative support approach. Procedia-social and behavioral sciences, 28, 97-101.

Lang, Q., Zhang, C., Qi, H., Du, Y., Zhu, X., Zhang, C., & Li, M. (2023). Mining and utilizing knowledge correlation and learners' similarity can greatly improve learning efficiency and effect: A case study on Chinese writing stroke correction. Sustainability, 15(3), 2393.

Von Luxburg, U. (2007). A tutorial on spectral clustering. Statistics and computing, 17, 395-416.

Gustriansyah, R., Suhandi, N., & Antony, F. (2020). Clustering optimization in RFM analysis based on k-means. Indonesian Journal of Electrical Engineering and Computer Science, 18(1), 470-477.

Wu, Di & Zhang, Mengtian & Shen, Chao & Huang, Zhuyun & Gu, Mingxing. (2020). BTM and GloVe Similarity Linear Fusion-Based Short Text Clustering Algorithm for Microblog Hot Topic Discovery. IEEE Access. PP. 1-1. 10.1109/ACCESS.2020.2973430.

Daoud, J. I. (2017, December). Multicollinearity and regression analysis. In Journal of Physics: Conference Series (Vol. 949, No. 1, p. 012009). IOP Publishing.

Kalnins, A. (2018). Multicollinearity: How common factors cause Type 1 errors in multivariate regression. Strategic Management Journal, 39(8), 2362-2385.

Asif, R., Merceron, A., Ali, S. A., & Haider, N. G. (2017). Analyzing undergraduate students' performance using educational data mining. Computers & education, 113, 177-194.

A.3 Webpages

gender gap | European Institute for Gender Equality - europa.eu. Available at: https://eige.europa.eu/thesaurus/terms/1178 (February 4 2023).

Data Analysis vs. Data Analytics: Definition and Types – Indeed, March 12 2023. Available at: https://ca.indeed.com/career-advice/career-development/data-analysis-vs-data-analytics (June 4 2023)

What is Data Analytics? - Definition from WhatIs.com. Available at: https://www.techtarget.com/searchdatamanagement/definition/data-analytics (June 3 2023)

Analysis vs. Analytics: How Are They Different?, May 12 2023. Available at: https://365datascience.com/trending/analysis-vs-analytics/ (June 4 2023)

Data Analysis vs. Data Analytics: 5 Key Differences – Upwork, January 10 2023. Available at: https://www.upwork.com/resources/data-analysis-vs-data-analytics (June 4 2023)

What is Data Analytics? – AWS. Available at: https://aws.amazon.com/what-is/data-analytics/ (June 28 2023)

What Is Semi-Supervised Learning - Machine Learning Mastery, April 09 2021. Available at: https://machinelearningmastery.com/what-is-semi-supervised-learning/ (July 24 2023)

What is Logistic regression? – IBM. Available at: https://www.ibm.com/topics/logistic-regression (June 27 2023)

Logistic Regression: Understanding odds and log-odds. Available at: https://medium.com/wicds/logistic-regression-understanding-odds-and-log-odds-61aecdc88846 (June 27 2023)

scipy.stats.kstest — SciPy v1.11.2 Manual. Available at: https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.kstest.html (August 3 2023)

scipy.stats.shapiro — SciPy v1.11.2 Manual. Available at: https://docs.scipy.org/doc/scipy-1.11.2/reference/generated/scipy.stats.shapiro.html (August 3 2023)

scipy.stats.ttest_ind — SciPy v1.11.2 Manual. Available at: https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.ttest_ind.html (August 7 2023)

scipy.stats.mannwhitneyu — SciPy v1.11.2 Manual. Available at: https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.mannwhitneyu.html (August 8 2023)

2.3. Clustering — scikit-learn 1.3.0 documentation. Available at: https://scikit-learn.org/stable/modules/clustering.html (June 4 2023)

scikit-learn/sklearn/cluster/_mean_shift.py at main – GitHub. Availabe at: https://github.com/scikit-learn/scikit-learn/blob/364c77e04/sklearn/cluster/_mean_shift.py (June 4 2023) sklearn.metrics.silhouette_score. Available at: https://scikit-learn.org/stable/modules/generated/sklearn.metrics.silhouette_score.html (June 4 2023)

sklearn.metrics.davies_bouldin_score. Available at: https://scikit-learn.org/stable/modules/generated/sklearn.metrics.davies_bouldin_score.html (June 4 2023)

sklearn.metrics.f1_score — scikit-learn 1.3.0 documentation. Available at: https://scikit-learn.org/stable/modules/generated/sklearn.metrics.f1_score.html (June 24 2023)

sklearn.linear_model.LogisticRegression - scikit-learn. Available at: https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html (May 23 2023)

sklearn.svm.LinearSVC — scikit-learn 1.3.0 documentation. Available at: https://scikit-learn.org/stable/modules/generated/sklearn.svm.LinearSVC.html (August 12 2023)

tf.keras.callbacks.EarlyStopping | TensorFlow v2.13.0. Available at: https://www.tensorflow.org/api_docs/python/tf/keras/callbacks/EarlyStopping (August 10 2023)

A.4 Conference/Symposiums/Workshops

Watson, C., & Li, F. W. (2014, June). Failure rates in introductory programming revisited. In Proceedings of the 2014 conference on Innovation & technology in computer science education (pp. 39-44).

Höök, L. J., & Eckerdal, A. (2015, April). On the bimodality in an introductory programming course: An analysis of student performance factors. In 2015 International conference on learning and teaching in computing and engineering (pp. 79-86). IEEE.

Koprinska, I., Stretton, J., & Yacef, K. (2015). Predicting student performance from multiple data sources. In Artificial Intelligence in Education: 17th International Conference, AIED 2015, Madrid, Spain, June 22-26, 2015. Proceedings 17 (pp. 678-681). Springer International Publishing.

Kostopoulos, G., Lipitakis, A. D., Kotsiantis, S., & Gravvanis, G. (2017). Predicting student performance in distance higher education using active learning. In Engineering Applications of Neural Networks: 18th International Conference, EANN 2017, Athens, Greece, August 25–27, 2017, Proceedings (pp. 75-86). Springer International Publishing.

Okubo, F., Yamashita, T., Shimada, A., & Ogata, H. (2017, March). A neural network approach for students' performance prediction. In Proceedings of the seventh international learning analytics & knowledge conference (pp. 598-599).

Figueiredo, J., Lopes, N., & García-Peñalvo, F. J. (2019, October). Predicting student failure in an introductory programming course with multiple back-propagation. In Proceedings of the Seventh International Conference on Technological Ecosystems for Enhancing Multiculturality (pp. 44-49).

Alzahrani, N., Vahid, F., Edgcomb, A. D., Lysecky, R., & Lysecky, S. (2018, June). An analysis of common errors leading to excessive student struggle on homework problems in an introductory programming course. In 2018 ASEE Annual Conference & Exposition.

Cobo, G., García-Solórzano, D., Santamaría, E., Morán, J. A., Melenchón, J., & Monzo, C. (2010, June). Modeling students' activity in online discussion forums: a strategy based on time series and agglomerative hierarchical clustering. In Educational data mining 2011.

Bian, H. (2011, April). A Preliminary Study on Clustering Student Learning Data. In MAICS (pp. 128-132).

Lopez, M. I., Luna, J. M., Romero, C., & Ventura, S. (2012). Classification via clustering for predicting final marks based on student participation in forums. International Educational Data Mining Society.

Matetic, M., Bakaric, M. B., & Sisovic, S. (2015, June). Association rule mining and visualization of introductory programming course activities. In Proceedings of the 16th International Conference on Computer Systems and Technologies (pp. 374-381).

Caton, S., Russell, S., & Becker, B. A. (2022, February). What fails once, fails again: Common repeated errors in introductory programming automated assessments. In Proceedings of the 53rd ACM Technical Symposium on Computer Science Education-Volume 1 (pp. 955-961).

Merceron, A., & Yacef, K. (2008, June). Interestingness measures for association rules in educational data. In: Educational Data Mining 2008, The 1st International Conference on Educational Data Mining, pp. 57–66. Montreal, Québec, Canada.

Karakasidis, A. (2023). Diorthotis: A Parallel Batch Evaluator for Programming Assignments. In: 29th International European Conference on Parallel and Distributed Computing Euro-Par 2023: Parallel Processing Workshops.

Flowers, J. C. (2019, March). Strong and Weak AI: Deweyan Considerations. In AAAI spring symposium: Towards conscious AI systems (Vol. 2287, No. 7).

Pedro, M. O., Baker, R., Bowers, A., & Heffernan, N. (2013, July). Predicting college enrollment from student interaction with an intelligent tutoring system in middle school. In Educational Data Mining 2013.

Wang, X., & Xu, Y. (2019, July). An improved index for clustering validation based on Silhouette index and Calinski-Harabasz index. In IOP Conference Series: Materials Science and Engineering (Vol. 569, No. 5, p. 052024). IOP Publishing.

Appendix A – Dissertation Literature Review Tables

The following tables summarize the information contained in the dissertation's Literature Review section.

The first one contains the abridged version of the section's information.

Related Work Summary Table	**Data Information**					**Methods**			**Aim**	
Research Name	**Contains Gender**	**Department Statistics**	**Course Data**	**Assignment Data**	**Student/Final Grades**	**Association Rules**	**Clustering**	**ML models**	**Course Analysis**	**Predictions**
Pillay, N., & Jugoo, V. R. - An investigation into student characteristics affecting novice programming performance - (2005)	V	V	V	V	V				V	
Damaševičius, R. - Analysis of academic results for informatics course improvement using association rule mining - (2010)			V	V (lab data)	V	V			V	
Cobo, G., García-Solórzano, D., Santamaría, E., Morán, J. A., Melenchón, J., & Monzo, C. - Modeling students' activity in online discussion forums: a strategy based on time series and agglomerative hierarchical clustering - (2010)			V				V		V	
Bian, H. - A Preliminary Study on Clustering Student Learning Data - (2011)				V	V		V		V	V
Lopez, M. I., Luna, J. M., Romero, C., & Ventura, S. - Classification via clustering for predicting final marks based on student participation in forums - (2012)			V		V		V			V
ElGamal, A. F. - An educational data mining model for predicting student performance in programming course - (2013)	V	V	V		V			V	V	V
Höök, L. J., & Eckerdal, A. - On the bimodality in an introductory programming course: An analysis of student performance factors - (2015, April)			V (questionnaire)		V				V	
Koprinska, I., Stretton, J., & Yacef, K. - Predicting student performance from multiple data sources - (2015)			V	V	V			V	V	V
Matetic, M., Bakaric, M. B., & Sisovic, S. - Association rule mining and visualization of introductory programming course activities - (2015, June)			V	V	V	V			V	
Mueen, A., Zafar, B., & Manzoor, U. - Modeling and Predicting Students' Academic Performance Using Data Mining Techniques - (2016)		V	V	V	V			V		V
Kostopoulos, G., Lipitakis, A. D., Kotsiantis, S., & Gravvanis, G. - Predicting student performance in distance higher education using active learning -	V	V	V	V	V			V		V

(2017)										
Okubo, F., Yamashita, T., Shimada, A., & Ogata, H. - A neural network approach for students' performance prediction - (2017, March)			V	V	V			V		V
Ayub, M., Toba, H., Yong, S., & Wijanto, M. C. - Modelling students' activities in programming subjects through educational data mining - (2017)		V	V	V (quizzes)	V	V			V	
Alzahrani, N., Vahid, F., Edgcomb, A. D., Lysecky, R., & Lysecky, S. - An analysis of common errors leading to excessive student struggle on homework problems in an introductory programming course - (2018, June)				V					V	
Bucos, M., & Drăgulescu, B. - Predicting student success using data generated in traditional educational environments - (2018)	V (not in final data)	V	V	V (activities)	V			V		V
Islam, N., Shafi Sheikh, G., Fatima, R., & Alvi, F. - A Study of Difficulties of Students in Learning Programming - (2019)			V			V	V		V	
Figueiredo, J., Lopes, N., & García-Peñalvo, F. J. - Predicting student failure in an introductory programming course with multiple back-propagation - (2019, October)			V (field)	V	V			V		V
Caton, S., Russell, S., & Becker, B. A. - What Fails Once, Fails Again: Common Repeated Errors in Introductory Programming Automated Assessments - (2022, February)		V (student emails)	V	V		V			V	

The second table contains more detailed information, like in the written section.

Research Name	**Data Info**	**Programming Language & Subject**	**Methods**	**Aim**
Pillay, N., & Jugoo, V. R. - An investigation into student characteristics affecting novice programming performance - (2005)	Gender contained, demographic characteristics, answers from questionnaire, from field observations (learning style), from assignments, final grades	Java, procedural programming	Questionnaire analysis, statistics tests	Determine the factors shaping students' programming performance – a student's problem solving capacity and their maternal language were found to be the most influential.

Damaševičius, R. - Analysis of academic results for informatics course improvement using association rule mining - (2010)	Class/instruction-related, from online e-learning activity, assignment grades, final grades	Object oriented programming	Association Rule Mining	Define the relationship between lab test and exam grades with course failure – the “Linear Dynamic List” lab was found to cause significant student failure, according to many of the chosen metrics, prompting the professors to reevaluate the corresponding lecture’s instruction material.
Cobo, G., García-Solórzano, D., Santamaría, E., Morán, J. A., Melenchón, J., & Monzo, C. - Modeling students' activity in online discussion forums: a strategy based on time series and agglomerative hierarchical clustering - (2010)	From online community activity (forums)	Electronic Circuits Theory	Agglomerative Hierarchical Clustering	Introduce a method to analyze student activity in online forums through student clustering in different groups according to their forum activity. Each student’s forum reading or posting patterns are represented as time series, while the hierarchical clustering method was chosen as the amount of clusters is not known in advance. The resulting dendrograms from the clustering are vertically divided, not horizontally.
Bian, H. - A Preliminary Study on Clustering Student Learning Data - (2011)	From assignments, final grades	Computer Science Service Course	(Simple)KMeans Clustering, Subspace Clustering	The application and comparison of Subspace Clustering and other clustering methods, most notably KMeans, in student activity grade data to predict which students are more likely to fail as early as possible, along with the reasons behind this failure. Specific subsets of activities were found to be good indicators of whether the students would fail.

Lopez, M. I., Luna, J. M., Romero, C., & Ventura, S. - Classification via clustering for predicting final marks based on student participation in forums - (2012)	From online community activity, final grades	First year computer course in computer engineering	Clustering with Weka’s: “EM, FarthestFirst, HierarchicalClusterer, sIB, SimpleKMeans, and XMeans”, Classification with Weka’s: DTNB, JRip, NNge, Ridor, ADTree, J48, LADTree, RandomForest, Logistic, MultilayerPerceptron, RBFNetwork, SMO, BayesNet and NaiveBayesSimple	The aim is double: determine if the students’ final course grades can be predicted based solely on the course’s online discussion data as well as if classification by clustering can be used instead of the native classification algorithms. Tests were conducted both with all the available forum indicators and with only their best subset. It turned out it is possible to use only forum-based data to predict the students’ final course grade as well as that, the EM clustering algorithm can achieve similar levels of accuracy as the more usual classification algorithms, though the rest of the tested clustering algorithms cannot.
ElGamal, A. F. - An educational data mining model for predicting student performance in programming course - (2013)	Gender contained, from department’s student statistics, from online e-learning activity, final grade	An e-learning course, probably programming, from a Computer Science department	Decision Tree	Determine elements and factors that would could represent students’ programming course performance and demonstrate the Decision Tree’s usefulness for this – factors like “High School Mathematics grade” and “Programming Aptitude” were deemed to be useful.
Höök, L. J., & Eckerdal, A. - On the bimodality in an introductory programming course: An analysis of student	Answers from questionnaire, final grades	Introductory programming course	Questionnaire Analysis, visualization	Explain the relationship between different factors characterizing students and their course engagement with passing an introductory programming course – many of the students who considered the subject challenging did not engage with the course actively as much as the higher performers, solo work might be

performance factors - (2015, April)				preferable to teamwork.
Koprinska, I., Stretton, J., & Yacef, K. - Predicting student performance from multiple data sources - (2015)	From assignments, from online e-learning activity, from online community activity, final grades	First year programming course	Decision Tree	Determine profiles of low, average and high performing students and predict the exam grade early enough for intervention.
Matetic, M., Bakaric, M. B., & Sisovic, S. - Association rule mining and visualization of introductory programming course activities - (2015, June)	From online e-learning activity, from assignments, final grades	Second Introductory Programming Course (named “Programming 2”)	Association Rules, visualization, PCA	Find the factors responsible for the student academic results and improve certain course aspects – not participating in self-assessments or the forum entails course failure, while watching the video lectures is associated with success, thus among the decided measures, it is the coverage of more units with video lectures.
Mueen, A., Zafar, B., & Manzoor, U. - Modeling and Predicting Students' Academic Performance Using Data Mining	Demographic characteristics, from assignments, from online e-learning activity, from online community activity, final grades	“Programming Fundamental” and “Advanced Operating System”	Naïve Bayes, Multilayer Perceptron, decision tree	Prediction of the students’ passing or failing the course, Naïve Bayes managed the highest accuracy: 86% with all the characteristics or 85.7% with only the chosen 6.

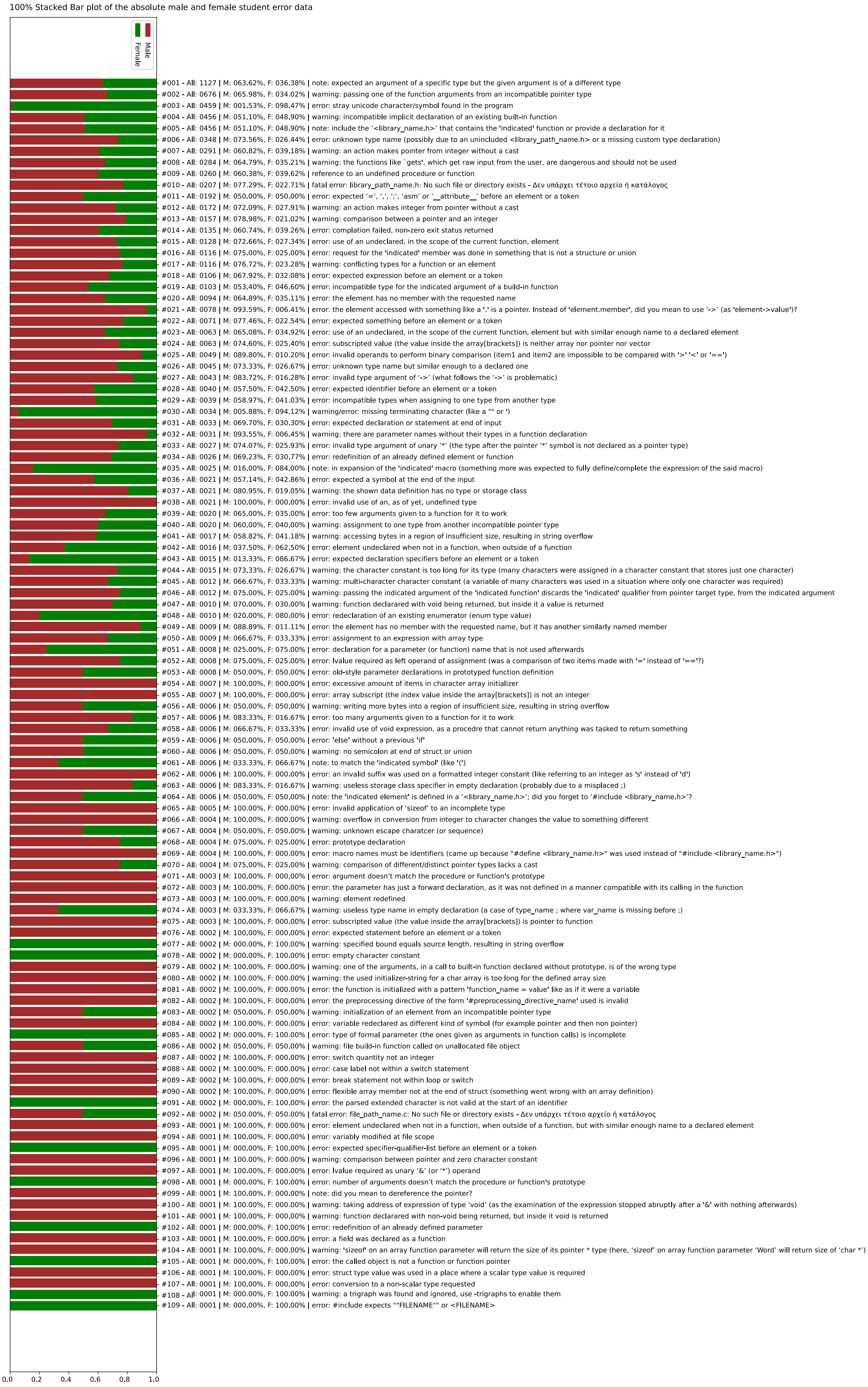
Techniques - (2016)				
Kostopoulos, G., Lipitakis, A. D., Kotsiantis, S., & Gravvanis, G. - Predicting student performance in distance higher education using active learning - (2017)	Gender contained, demographic characteristics, class/instruction related, assignment grades, final grades	Undergraduate distance learning module “Introduction to Informatics”	J48 Decision Tree and JRip, Logistic Regression, Multilayer Perceptrons, Naïve Bayes and Sequential Minimal Optimization, Semi-Supervised Learning	Predict whether the students would pass or fail the course, SMO was the best performer with accuracy >75% and was followed by NB and MLP.
Okubo, F., Yamashita, T., Shimada, A., & Ogata, H. - A neural network approach for students' performance prediction - (2017, March)	From assignments, from online e-learning activity, from online community activity, final grades	“Information Science” course	RNN (LSTM) and multiple regression analysis	Prediction of students’ final course grade – accuracy over 90% from the LSTM RNN by week 6 out of 15.
Ayub, M., Toba, H., Yong, S., & Wijanto, M. C. - Modelling students’ activities in programming subjects through educational data mining - (2017)	From online e-learning activity, from department’s student statistics, from assignments (quizzes) final grades	“Introductory Programming” course and “Algorithm and Data Structures” course	Association Rules, Visualization	Produce association rules to determine the characteristics students’ engagement with the courses and improve the LMS used – correlation was found between students’ educative material access time activities in the LMS with their final grade – gamification is proposed to make LMS utilization more alluring.

Alzahrani, N., Vahid, F., Edgcomb, A. D., Lysecky, R., & Lysecky, S. - An analysis of common errors leading to excessive student struggle on homework problems in an introductory programming course - (2018, June)	From assignments	Computer Science 1 introductory lesson instructed with C++	Error Analysis, visualizations	Propose a struggle metric and determine the activities causing the most struggle as well as the most common compilation errors, so that instructors would act accordingly to alleviate the situation.
Bucos, M., & Drăgulescu, B. - Predicting student success using data generated in traditional educational environments - (2018)	Gender contained, field observations, Class/Instruction-related, from assignments (activities), final grades	Object Oriented Programming	Python's Decision Tree CART, Extra Trees Classifier, Random Forest Classifier, Logistic Regression Classifier and C-Support Vector Classification	Prediction of whether students will pass the course or not – the accuracy achieved is around 86% by week 8's data, from the weeks 6, 8 and 12, by the classifiers of Random Forest, Logistic Regression and Support Vector Machine.
Islam, N., Shafi Sheikh, G., Fatima, R., & Alvi, F. - A Study of Difficulties of Students in Learning	Answers from questionnaire	First programming course	Questionnaire Analysis, Visualization, Association Rules, Clustering	Determine students' feelings and potential difficulties with their 1st programming course – specific parts were considered difficult by the students, while the mediums the students learn from with greater ease were located.

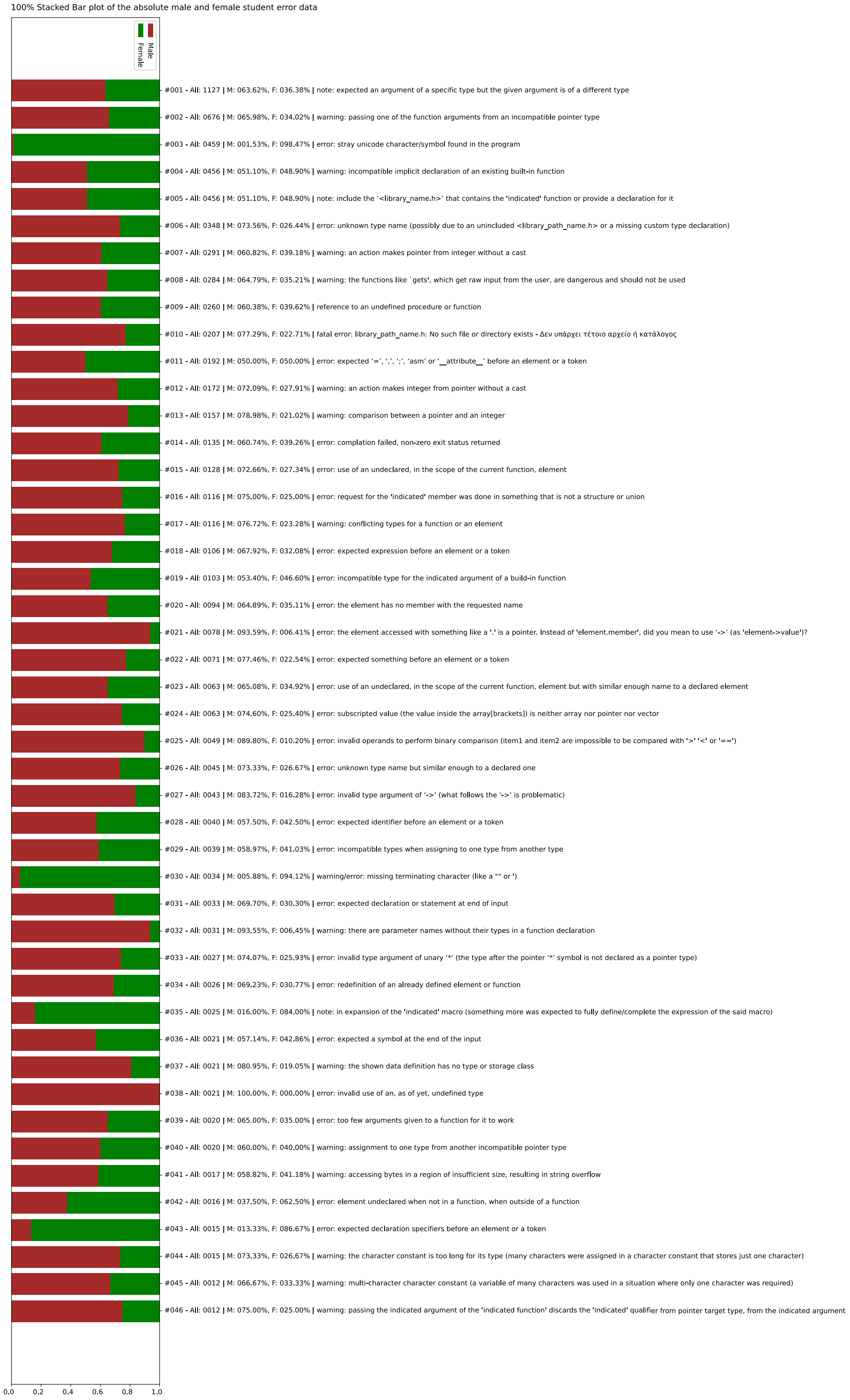
Programming - (2019)				
Figueiredo, J., Lopes, N., & García-Peñalvo, F. J. - Predicting student failure in an introductory programming course with multiple back-propagation - (2019, October)	Field observations, from assignments, final grades	Introduction to Programming	Multiple back-propagation neural network	Predict whether students will fail the course – achieved accuracy over 94%, only 3 students were misclassified and just 1 of them was wrongfully considered successful.
Caton, S., Russell, S., & Becker, B. A. - What Fails Once, Fails Again: Common Repeated Errors in Introductory Programming Automated Assessments - (2022, February)	From assignments, from department’s student statistics (student email)	Java, introductory programming course	Marcov chain analysis, association rules	Find the common but hard to solve compilation errors students could do.

Appendix B – Errors and their Frequencies

Plot of all the 109 errors, their frequencies and their gender percentages.



Plot of the 46 most usual errors, their frequencies and their gender percentages.



Appendix C – Conference Contribution Paper

Part of the findings of this dissertation and more precisely most of the Visualizations, Statistical Test results and created Association Rules are used as the basis for the Contribution Paper, accepted in the “26th International Conference on Interactive Collaborative Learning & 52nd IGIP International Conference on Engineering Pedagogy (ICL2023)” which will be realized in Madrid, Spain, from the 26th to the 29th of September of 2023. The contribution’s title is “Programming Errors and Academic Performance in an Introductory Data Structures Course: a Per Gender Analysis” from the authors Evangelos Dagklis, Maya Satratzemi, Georgia Koloniari and Alexandros Karakasidis and it will be published as a part of the ICL2023 Proceedings with Springer in the series Lecture Notes in Networks and Systems. As noted in the paper itself, it is a result of research conducted within the “MSc in Artificial Intelligence and Data Analytics” of the Department of Applied Informatics of University of Macedonia and its presentation is funded by the University of Macedonia Research Committee.