Universidad Politécnica
de Madrid

Escuela Técnica Superior de Ingenieros Informáticos

European Master in Software Engineering

Master Thesis

Studying digital literacy training performance with Machine Learning by clustering, prediction and data analytics

Author: Wenqian Gao

Madrid, June 2025

This Master Thesis has been deposited in ETSI Informáticos de la Universidad Politécnica de Madrid.

Master Thesis

European Master in Software Engineering

Title: Studying digital literacy training performance with Machine Learning by clustering, prediction and data analytics

June 2025

Author:	Wenqian Gao
Supevisor:	Susana Munoz Hernandez Departamento de Lenguajes y Sistemas Informáticos e Ingeniería de Software Escuela Técnica Superior de Ingenieros Informáticos Universidad Politécnica de Madrid
Co-supervisor:	Angel Herranz Nieva Departamento de Lenguajes y Sistemas Informáticos e Ingeniería de Software Escuela Técnica Superior de Ingenieros Informáticos Universidad Politécnica de Madrid

Abstract

Online learning platforms have gained popularity in recent years and have evolved rapidly since the COVID-19 pandemic. Throughout this thesis, we are focusing on a specific online course, called RETOMadrID, which provided our students with essential computer skills from scratch. The end goal is to reduce the accessibility gap between people and technology, ensuring that no adults are left behind within this ever-evolving digital world.

This thesis presents a Learning Analytics (LA) study conducted on RETOMadrID. The goal is to improve the platform by understanding students' behavior using modern machine learning (ML) and data analysis techniques. Understanding students' behavior helps us to keep them engaged and possibly identify at-risk students. To do this, we highlighted two research questions that will be focused on: 1) Will the students answer the questions in the questionnaire provided? 2) Will the students be able to complete the course? The main tool that we chose to use is the Elixir Livebook. During the implementation process, both supervised and unsupervised learning strategies were applied, including k-mode algorithm, decision tree, and logistic regression. The performance of the model was evaluated using different corresponding metrics such as accuracy, recall, silhouette score, SHAP values, and more.

Unsupervised learning via k-mode clustering successfully identified 3 distinct student profiles within our student population. Supervised learning through decision tree and random forest models achieved an accuracy of 99% in predicting whether the student will answer the question. With the help of the feature importance scale, we identified key factors contributing to a high response rate, such as student engagement. Another supervised learning method, called logistic regression, was used to predict the likelihood of course completion, achieving 74% accuracy. There are several long-term predictors affecting the outcome, including the total time spent on the course, the number of triggered activities, and the enrollment duration.

The study identified that inactive students and unfamiliarity with Likert scale questions were major barriers to interaction. Additionally, while time-based features were strong predictors of course completion, they are not suitable for early risk detection, highlighting a key area for future research.

Despite challenges such as limited data size and the need to refine clustering algorithm stability, this work demonstrated how ML can support personalized

educational interventions. Furthermore, it showcased the effective integration of Python and Elixir within Livebook, contributing a practical, reproducible workflow for future data analysis in online education environments.

Contents

Chapter 1

Introduction

As technology advances and the widespread impact of the COVID-19 pandemic, remote work and online learning have gained popularity. In particular, online education has evolved in recent years, providing a more accessible and effortless experience for everyone across the world. According to the World Economic Forum, the online learning platform Coursera recorded 20 million new student registrations in 2021, which is equivalent to total growth in the three years pre-pandemic [1]. This trend highlights how individuals are evolving to cope with the demands of today's world of work.

The online course we are focusing on in this thesis is targeted toward those people who may lack the knowledge to use modern technology effectively. It is derived by RETOMadrID [2]. Most of the students here are over 60 years old, a group that often faces unique challenges when using digital tools. By offering this course, we aim to help these learners with the skills they need to navigate technology more independently and effortlessly. Furthermore, we are trying to reduce the accessibility gap between people and technology, ensuring that no adults are left behind within this ever-evolving digital world. This initiative promotes digital inclusion and supports more equitable access to information and services for all.

1.1 RETOMadrID

The project 'Reequilibrio Territorial en Madrid con Inclusión Digital', also known as RETOMadrID, is conducted in collaboration with the local Government of the Region of Madrid and the Technical University of Madrid (Universidad Politécnica de Madrid). The initiative addresses the growing need for digital skills to carry out many essential tasks in our daily life, such as booking medical appointments, staying in touch with family, or simply accessing any online services. The goal is to ensure people situated in rural areas are not left behind in the current digital era, or anyone within the target audience, by providing them with essential computer skills from scratch. The target audience includes adults over 18 living in rural areas within the region of Madrid, with a particular focus on women in vulnerable situations, the unemployed, elderly people,

cultural minorities, and anyone lacking computer skills.

1.2 Elixir Livebook

The RETOMadrID platform is built using Elixir [3], a functional programming language known for its scalability and robustness in concurrent systems. Elixir has gained popularity in building scalable and fault-tolerant applications. In recent years, it has also been increasingly adopted for data analysis tasks, thanks to a growing ecosystem of supportive libraries. Libraries such as Explorer and Vega-Lite have been used for data manipulation and visualization, while libraries such as Scholar facilitate the creation of machine learning models.

To facilitate the use of the Elixir programming language in the field of data analysis, Livebook is designed and used broadly. Livebook is a web application for writing interactive Elixir code notebooks [4]. Furthermore, it supports inline documentation, rich data visualization features that significantly enhance the data analysis process. To build on the fact that the application is implemented in Elixir, Livebook was chosen as the development environment due to its strong alignment with Elixir and its well-defined features that facilitate the data analysis procedure. For clarity of the code sessions, explanatory markdown is also added throughout the Livebook session, providing a comprehensive walkthrough of the entire implementation.

1.3 Research questions

Throughout this thesis, we will make use of machine learning techniques and try to answer the following questions:

- **Will the students answer the questions in the questionnaire?** - In some cases, unanswered questions may reflect challenges in comprehension, either due to cognitive abilities or difficulties in understanding the questions. Another reason may be due to the fact that they simply don't want to answer, or even due to personal reasons. Therefore, it is interesting to discover and analyze which questions are unlikely to be answered by the student and which factors are affecting it.
- **Will the students be able to complete the course?** - For this research question, we are concentrating on the student performance. This focus is critical because early identification of students at risk of not finishing allows us to adapt the course strategy to better support their learning. By understanding the factors that contribute to course completion, we can implement targeted support mechanisms, such as personalized feedback, additional resources, to improve overall student outcomes and reduce the number of at-risk students.

By answering the above-mentioned questions, we can extend our understanding of student behavior and performance patterns. Furthermore, it will help inform the design of the project toward more personalized online education strategies,

specifically tailored to the needs and challenges of our target audience: Students with difficulties in engaging with the digital learning platform.

1.4 General Structure

To facilitate a clearer understanding of the entire documentation, we divide the content according to the following structure:

- **State of the art:** Description of current approaches in the field.
- **Methodology:** Description of the machine learning approaches used in this thesis. Including the machine learning algorithms applied.
- **Analysis and results:** The description of the machine learning results and the corresponding findings.
- **Limitations and future work:** The problem faced throughout this research, and from that, we derive our future work that could be focused.
- **Conclusion:** Summary of the entire thesis.

Chapter 2

State of the art

Data analysis has been widely known for its ability to discover useful information, draw conclusions, and support decision-making. Learning Analysis (LA), also known as Educational Data Mining (EDM) [5], the ability to find patterns in learners' data for decision-making purposes, has attracted many researchers in recent years. This type of analysis allows institutions to learn about student behavior and status, allowing each institution to tailor their materials for individual students based on observed outcomes.

In the context of online learning, due to the absence of face-to-face interaction in web-based systems, it is difficult to measure student engagement and performance, such as attendance and interaction of the students in the courses. As a result, LA mainly relies on data, such as student profile, their online behavior, and surveys via questionnaires, which are mainly used as the main resources. For prediction and analysis purposes of students' behavior, researchers have investigated several machine learning models and algorithms, including clustering, linear regression, decision trees, and deep learning models such as convolutional neural networks.

Throughout this session, we will dive into some of the existing research on clustering and various supervised learning methods. We will also highlight how the existing studies have informed the specific methodological choices used in our study.

2.1 Clustering in student performance

Clustering is an unsupervised machine learning algorithm used to group similar objects when there is no prior knowledge about the structure or categories within the dataset [6]. Its goal is to reveal the underlying classes present within the data. Clustering has been applied in several applications, such as medical science, education, wholesales, etc [7].

In Learning Analytics (LA), clustering techniques have been applied to student login data to gain insight into the level of achievement in online courses. More specifically, metrics such as the average number of logins per assignment, the

average number of attempts per assignment, and the average score per student are used to classify students into two groups: high achievers and low achievers. There are a variety of algorithms that you can use to perform clustering, one of which K-means algorithm is widely used. For instance, data from the Black-Board Vista platform was collected and analyzed using the K-means clustering algorithm. Which results in identifying 3 different learners' styles [8]. In addition to K-means, other algorithms such as CLARA and PAM were used. These have also been employed and evaluated on the online learning platform dataset [9].

While K-means and other above-mentioned algorithms are effective for numerical data, they are less suitable for categorical datasets. In the case of categorical data, the K-modes algorithm would be the popular choice. The K-mode algorithm is specifically designed by replacing the Euclidean distance metric with a simpler matching dissimilarity measure. In recent studies, such an algorithm has been used to classify student stress levels, using categorical data such as responses to questions about family issues or social relationships. [10].

2.2 Supervised learning

Clustering techniques are typically used in scenarios where there are no target labels, while supervised learning relies on a known target label to guide the model during its training process. In the field of LA, supervised learning has been widely introduced to predict student performance [11] and to identify at-risk students [12]. Traditional machine learning algorithms such as regression analysis, decision trees, neural networks, and other methods have shown strong predictive capabilities in this context. For example, decision tree models have been successfully used to explore the association of students' demographic characteristics and their performance, revealing the fact that region, neighborhood poverty level, and prior education, respectively, were strong predictors of overall learning outcomes [11].

However, these old-fashioned machine learning algorithms are still available; their algorithms are based on a single prediction model, making their performance highly dependent on the characteristics of the training dataset [12]. To overcome this drawback, ensemble learning has been introduced, which makes use of multiple models. It introduces a stronger emphasis on multiple weak classifiers into a strong classifier, improving the overall performance of predictions. One prominent example is the Random Forest model, which constructs multiple decision trees, performs training in each tree on a different random subset of the training data, and aggregates their outputs to enhance prediction accuracy [12].

Logistic regression is another widely used model. Logistic regression is derived from linear regression models, but it is specifically designed to predict categorical values. Current studies have shown success in predicting student performance using data related to student behaviors, such as course click rates [13] and the rate at which the student stays online [14]. These are examples of strong

indicators for judging active learning and withdrawal behaviors.

The above-mentioned machine learning methods work well with moderately sized datasets. Some popular online platforms, such as MOOCs, introduce a new challenge due to their huge data volume. In one of the recent records, MOOCs have reached over 220 million learners and 19 thousand courses [15]. In this scenario, it is necessary to analyze such a huge amount of data using a more sophisticated architecture, called a Neural Network (NN). Within the NN domain, a Convolutional Neural Network (CNN), a special type of NN, is broadly used. CNN is also well-suited when handling complex data like images or facial expressions. For instance, an analysis of student engagement level has been studied using three different CNNs (All-CNN, NiN-CNN, and VD-CNN) through visual imagery analysis in the online environment [16]. Moreover, Spiking Neural Networks (SNNs) offer a brain-inspired approach to online learning that can adapt to changing environments without the need for retraining, unlike traditional models that require retraining [17]. This characteristic allows SNNs to be particularly suitable for real-time scenarios [17].

2.3 Our Focus

Now that we have explored all the current literature and methodologies in the field, we move our attention to the approach adopted in this study. Given that our data is mostly categorical (See Chapter 3.1), K-modes has been selected as the primary algorithm for the clustering process in this study. This clustering technique will later be used to classify and describe the participants' group effectively. For the supervised learning component, we will experiment with most of the methods mentioned, including decision tree, random forest, and logistic regression. Ensuring that we gain an answer to our research question highlighted in chapter 1.3. Furthermore, we will not explore Neural Networks due to the limited size of our dataset.

Chapter 3

Methodology

In this chapter, we will outline the structured workflow applied in this study for data analysis on student performance. The methodology we select follows a systematic approach to ensure the effectiveness of the analytical process. It includes 5 key stages: data Understanding, data preprocessing, feature selection, model development, and model evaluation (See figure 3.1).

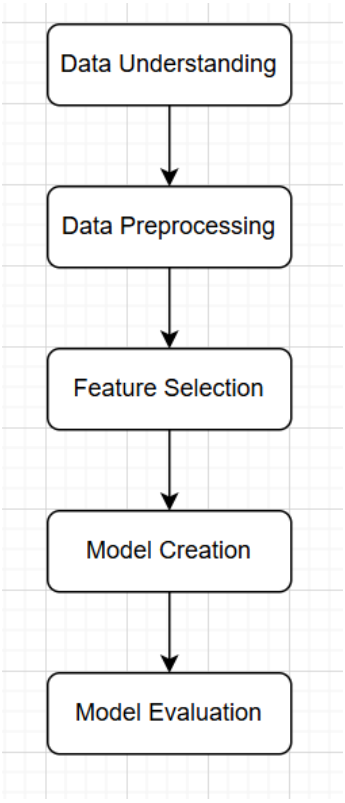


Figure 3.1: Methodology workflow

Keep in mind that throughout the study, Livebook is the tool that we are exploring and focusing. Hence, the entire methodology workflow is carried out in

Livebook.

3.1 Data understanding

The platform hosts a list of courses, each consisting of multiple topics. The dataset that we are using in this study consists of 3 different CSV files, all of which can be downloaded from the user interface (see Figure 3.2):

- **answer.csv**: A CSV file containing the answers submitted by the students in response to a collection of questions provided to them.
- **participation.csv**: A CSV file containing a summary of each student's participation in the course. For example, it includes an indicator of whether the student has completed 75 percent of the course.
- **events.csv**: A CSV file containing different events that the students have performed throughout the course.

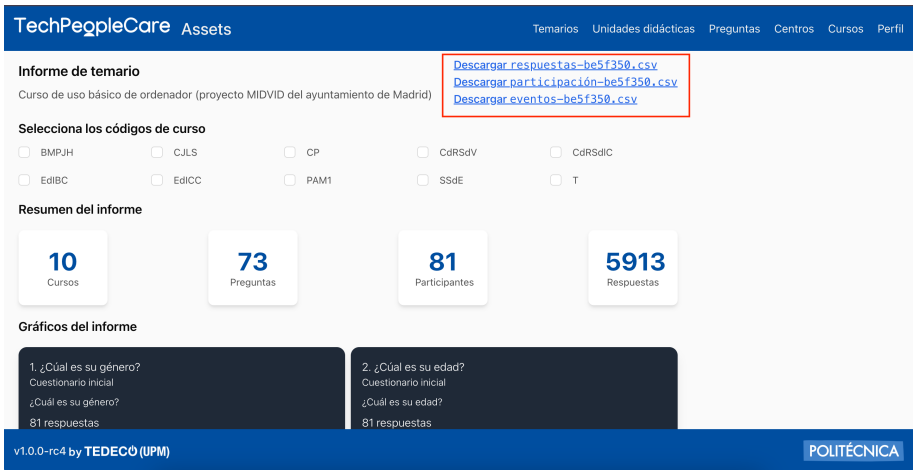


Figure 3.2: A screenshot indicating where to find the CSVs used in this study

3.1.1 Answers.csv

The data in this CSV file contains a collection of answers provided by the students. It is important to note that not all questions were answered by the students. These unanswered questions are not included in this CSV file.

The overall structure, in other words, the schema of the data is listed in table 3.1. The table also includes a description of each column to clarify further the type of information represented.

Table 3.1: An overview of "Answers.csv" schema

Begin of Table		
Column Name	**Type**	**Description**
ID de temario / Course ID	Type	A unique identifier for each course program (contents).
Título de temario / Course title	String	Title of the course.
Código de curso / Course code	String	A unique code representing a specific training course (the training of contents in a host institution during the time between a beginning and an ending date).
Código de participants / Participant's code	String	A unique code representing a specific participant in a specific training course.
ID Actividad / Activity ID	String	A unique identifier for each activity performed by the student. These activities include answering questions, entering the course, and more.
ID de pregunta / Question ID	String	An unique identifier for each question.
Categoría de la pregunta / Question category	String	Categories of the questions. It consists of 7 categories: • *Profile* = Student's Background information. • *Access* = Level of access to technology. • *selfpre* = Self perception of knowledge level related to the technology at the beginning of the course. • *feedback* = Feedback related to a topic of the contents. • *Selfpost* = Self perception of the knowledge level after completing the course. • *Satisfaction* = Satisfaction level after the completion of the course. • *Usability* = Perceived usability of the contents of the course.

Continuation of Table 3.1

Column Name	**Type**	**Description**
Tipo de pregunta / Question type	String	There are two types of questions: Multiple Choice Questions (MCQ) or Likert scale questions.
Texto de la pregunta / Question text	String	The complete wording of the question displayed to students during the survey or activity.
Selección de la respuesta / Answer selection	Integer	The answer chosen by the student.
Texto de la respuesta / Answer text	String	Full text of the answer option.
ID de participante / Participant ID	String	A unique ID for each of the students in the course

End of Table

Among all the columns, the "question text" column and the "answer text" columns are particularly useful in gaining meaningful insights. Together with the "question category" column, they allow us to collect general information such as: the overall participants' age range and the overall students' satisfaction.

Age distribution

Figure 3.3 shows the age distribution of the students enrolled in the course. We can see that the data are strongly skewed toward older adults, reflecting the fact that the course is particularly popular among those looking to improve digital literacy later in life. The largest group of participants has an age between 60 and 69 years old. Followed by a group of student between the ages of 70 and 79. We can also see that from the age of 80, the number of students decreases significantly. And those under 17 years of age comprise only a tiny fraction of the participants. Younger adults, such as those aged 18 to 39 and 40 to 49, are present but represent a smaller portion of the overall population.

In general, the age distribution suggests that the course is designed to attract older adults and is likely to be specifically marketed to this demographic.

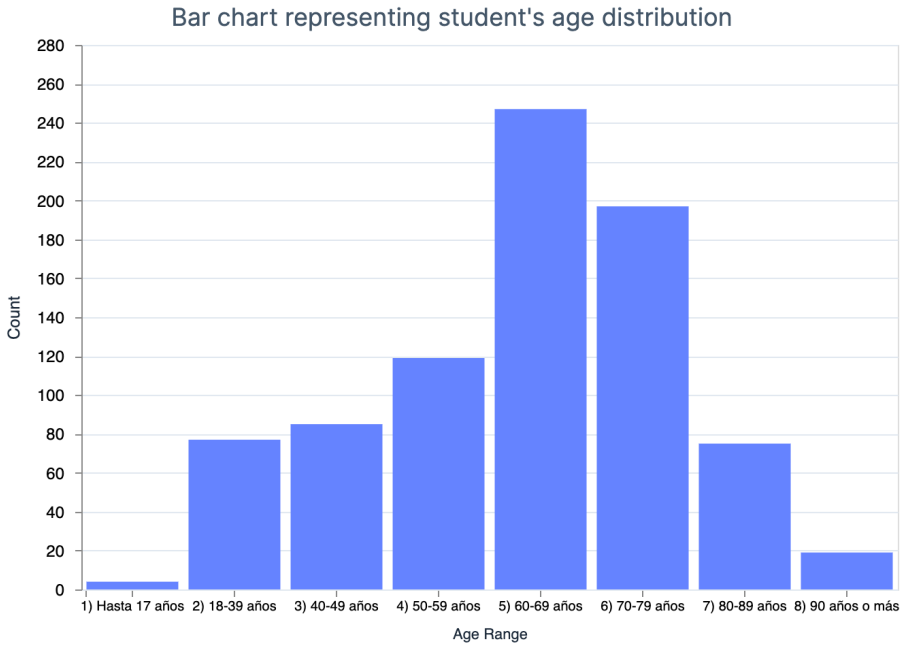


Figure 3.3: Age distribution of the students

Student satisfaction

Figure 3.4 shows the distribution on the level of satisfaction levels based on the level of likes provided by the student. From the figure, we can see that our students expressed a strong positive sentiment toward the course. The majority of students selected "A lot" (210 responses) and "Quite a lot" (115 responses). On the other hand, only 3 students selected "A bit", and another 3 chose "Nothing", demonstrating a very low satisfaction among a minority of the overall group of students.

Overall, the data suggests that the course is liked by a majority of students. More specifically, over 90% of participants reported that they liked the course while selecting either "Quite a lot" or "A lot", providing the evidence that the course is effectively meeting their expectations.

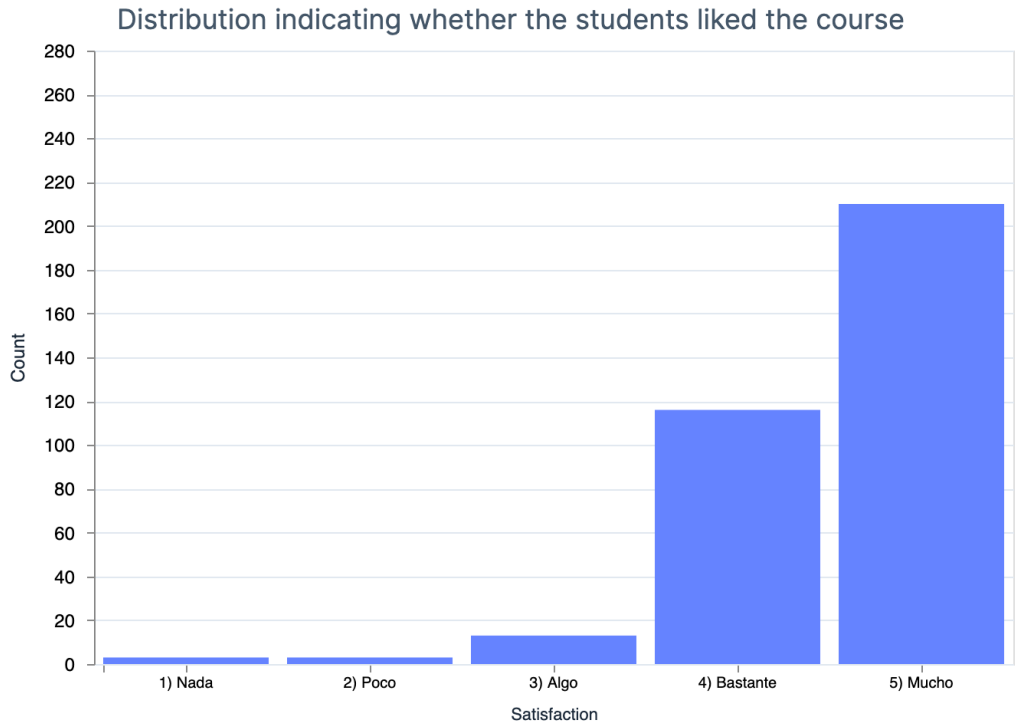


Figure 3.4: Level of satisfaction distribution of the students

3.1.2 Participation.csv

The data in this CSV file contain a summary of the student performance. It includes various metrics that reflect their level of participation throughout the course.

The schema of the data is listed in Table 3.2. The table also includes a description of each column to further clarify the type of information represented. The columns may include data such as the number of events triggered by the student, the number of answers provided by the student, etc.

Table 3.2: An overview of "Participation.csv" schema

Begin of Table		
Column Name	**Type**	**Description**
Participante / Participant	String	A unique code representing a specific participant in a specific course.
Centro / Center	String	Name of the organization where the online course is carried out.
Curso / Course	String	An unique code representing a particular course.
Temario / Course	String	Title of the course

Continuation of Table 3.2

Column Name	Type	Description
Al menos 7h30 / At least 7h30	Boolean	This is a boolean indicator, indicating whether the student has watched the course video for at least 7 hours and 30 minutes.
Último tema terminado / Finished the last topic	Boolean	This is a boolean indicator, indicating whether the student has finished the last topic.
Al menos ha visto un 75% / At least 75% has been watched	Boolean	This is a boolean indicator, indicating whether the student has seen at least 75% of the course.
#eventos / Number of events	Integer	The number of events triggered by the student.
Número de videos reproducidos / Number of videos reproduced	Integer	The number of videos reproduced by each student.
Número de respuestas / Number of answers	Integer	The total number of answers provided by each of the students.
Tiempo mirando videos / Time watching video	String	The total recorded time of watching all the videos of the contents
Tiempo siguiendo curso / Time following the course	String	The total recorded time of following the course, watching videos plus practicing.
Hora comienzo curso / Course start timestamp	String	The time when the student started the course.
Hora ultima actividad / Last activity timestamp	String	The time when the student performed their last activity.
ID de participante / Participant ID	String	A unique ID for each student in a course.

End of Table

Considering the column descriptions, it is clear that this dataset plays a crucial role in identifying student engagement levels and evaluating overall student performance and course completion. By deriving specific metrics, we can gather information on both the engagement and the process of the students.

Time Spent on Videos

Time spent on video is an interesting metric to focus on, as it reveals insight into the active level of students by capturing how deeply they interact with course content. In an online course environment, passive participation is often measured by the amount of time a student spends viewing instructional videos. If

we combine these data with the indicator of whether a student has completed 75% of the course (Figure 3.5), we can explore how students below and just above that 75% threshold differ in their viewing patterns, identify potential outliers, and understand the spread of the data.

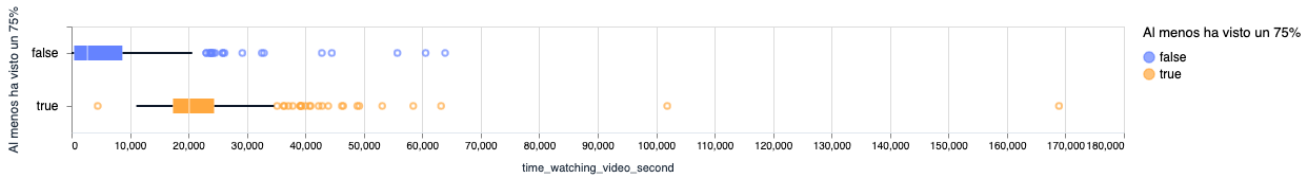


Figure 3.5: Box plot of video watching time for students who watched at least 75% of the course vs. those who didn't.

From Figure 3.5, we can see that people who completed the course by at least 75% have a longer median time to watch the video. This reflects the fact that people who managed to complete $\geq$75% of the course tend to spend more time watching videos. However, we can also notice that this group of students also includes outliers which the person managed to complete at least 75% of the video but only spent a few minutes following videos. This explains the situation where students skipped the course content because of a low concentration level on the course content, or simply due to the fact that the student finds it too easy.

Building on this, let's now focus on the total video watching time with course completion status (Figure 3.6). Since we don't have a column that describes whether or not a student finished the course, we will make use "At least 7h30" column. This column acts as an indicator when the student has spent at least 7 hours and 30 minutes watching course videos, serving as a practical threshold to approximate course completion. This boxplot illustrates differences in distribution and the presence of outliers between completers and non-completers.

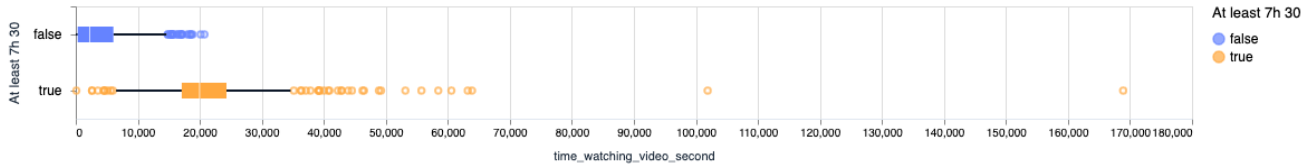


Figure 3.6: Box plot of video watching time for students who finished the course vs. those who haven't.

From Figure 3.6, we can observe a clear distinction in the distribution of video watching time between students who did manage to complete the course and those who did not. Students who met the 7 hours and 30 minutes threshold (our proxy of course completion) generally have a higher median time spent watching videos, compared to non-completers. Furthermore, we can find outliers in the group of students who did not meet the 7 hours and 30 minutes

threshold (our proxy of course incompletion), although they have also spent a significant amount of time following the course. This could indicate that high video engagement alone does not guarantee course completion. These outliers might reflect those who find it hard to follow the course. Their presence highlights the complexity of learner behavior and suggests that time spent is an important factor, but other factors might be involved in the success in an online course environment.

3.1.3 Events.csv

The data in this CSV file contains a collection of events triggered by students, each accompanied by its respective timestamp.

The overall structure, in other words, the schema of the data, is listed in table 3.3. The table also includes a description of each column to clarify further the type of information represented.

Table 3.3: An overview of "Events.csv" schema

Begin of Table		
Column Name	**Type**	**Description**
Participante / Participant	String	A unique code representing a participant in a course
Curso / Course	String	An unique code representing a specific student in the course.
ID Actividad / Activity ID	String	A unique identifier for each activity performed by the student. These activities include answering questions, entering the course, and more.
Tipo de actividad / Activity type	String	The type of activity performed by the student: • *video* = Video type activity. • *mcq* = Multiple choice question. • *likert* = likert scale question.

Continuation of Table 3.3

Column Name	**Type**	**Description**
Tipo de evento / Event type	String	The type of event triggered by the student, each activity is completed by the students by triggering different events: • *entered* = Enter event to the specific activity. • *moved_backward* = Move backward event when watching video. • *paused* = Pause event when watching video. • *resumed* = Resumed event when watching video. • *exited* = Exited event from the activity. • *finished* = Finished event of the activity. • *replayed* = Replayed even when watching video. • *android_on_start* = Event to represent the start course event on an Android device. • *android_on_stop* = Event to represent the stop course event on an Android device.
Ocurrío en / Ocurred in	String	The timestamp when the event is triggered.
ID de participante / Participant ID	String	A unique ID for each of the students in the course.

End of Table

Based on the column descriptions, it is clear that this dataset highly reflects student engagement levels throughout the course. Generally, the more events a student triggers, the higher their level of engagement. Furthermore, if there is a high number of "move backward" events on video, it may indicate that the content is difficult for the student to understand, prompting repeated attempts to re-watch it.

Student engagement level

In order to assess student engagement levels, one direct metric is to count the number of events performed by students. A higher number of events typically indicates higher engagement, as it reflects more interaction with the online course. These events may include entering and exiting the questionnaire, exiting and entering the course video, resuming the video, and moving backward within course videos, among others. In this analysis, we focus on the distribution of the total number of events triggered by each student. By seeing this, it may help us summarize the overall spread tendency of student engagement based on their collective video interactions (see Figure 3.7).

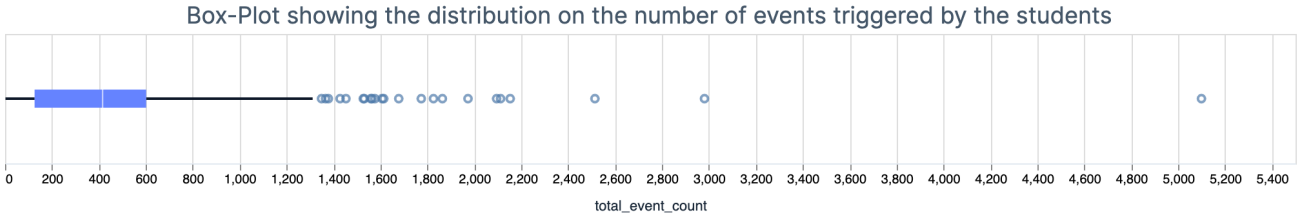


Figure 3.7: Box plot of the total number of events triggered by the students.

From Figure 3.7, we can see that the number of events is concentrated between 180 to 600. There are quite a lot of outliers at the same time. These outliers trigger a very high number of events, showing that they have an extreme level of engagement. This suggests that while most students interact with the course content within the range, a subset of students are highly active, possibly reviewing material multiple times.

"move backward" events on video

Viewing the pattern of total events triggered by the students only provides us with a general idea of the student engagement. However, if we want to focus on the student performance, more specifically, the difficulty our students may face with the course material, another metric should be considered. That is, the number of "move backward" events in videos. Moving backward on the course video tends to be a stronger signal that the student is struggling to understand a specific part, as it shows active effort to revisit a particular moment.

From Figure 3.8, we can see that the number of moves backward on video events tends to fall between 10 to 30. However, we can also observe that there exist outliers, where there is a high count of such events. These outliers likely represent students who are encountering difficulties in comprehending the course content. Notably, the presence of a considerable number of these outliers highlights that a high portion of students may be struggling with certain parts of the content.

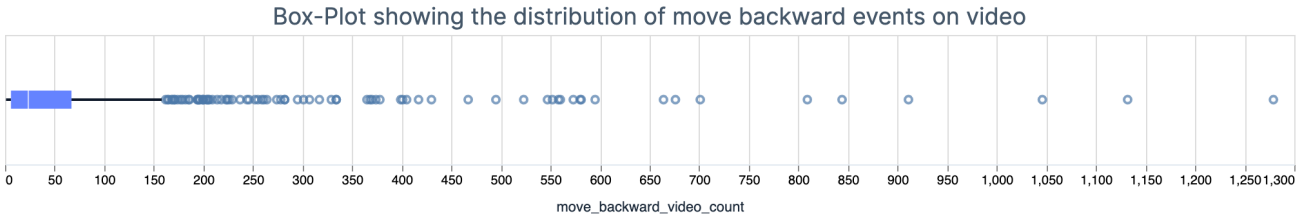


Figure 3.8: Box plot of the number of moves backward on video events triggered by the students.

3.2 Data Pre-processing

We have now gained an understanding of the data and how the different columns in our CSV file can reflect on the level of engagement, difficulty levels, and student satisfaction. With this foundation, we now proceed to the data pre-processing step, where we will prepare the data for applying machine learning techniques to address our research questions highlighted in Section 1.3.

3.2.1 Filter columns

In Table 3.1, there are two relevant columns: "Answer Selection" and "Answer text". The first one records the index of the selected option, and the latter one records the text representation of the selected option. Intuitively, they both convey the same piece of information, but one in numerical and one in string format. From a machine learning perspective, the machine learning algorithms operate on numerical values. Since the "Answer Selection" column already encodes the responses numerically, the "Answer Text" column becomes unnecessary. Therefore, it is removed as an initial step in the pre-processing steps.

Participation.csv is used mainly to answer our second research question related to student performance. In Table 3.2, we can see there are some direct columns that we exclude directly: Course, Course code, and Center. The information displayed in these columns does not directly affect how a student engages with the course content. Hence, these can also be removed.

3.2.2 Convert to numerical presentation

To determine whether a question is likely to be answered by a student, it is important to know the type of question, whether it's a multiple-choice question or a Likert question. Furthermore, the question category is also important (e.g. profile, usability, etc). These attributes can influence response behavior. For example, Likert-scale questions may feel more subjective for participants to answer compared to multiple-choice ones. Similarly, questions categorized a "Profile" tend to ask information related to personal income, which some students may feel uncomfortable answering, increasing the chance of skipping and ignoring. On the other hand, questions about usability may feel more neutral when students interact with them and thus are more likely to be answered.

However, since both type and category are recorded as string values, they must be converted into numerical values, making them suitable for a machine learning algorithm. This is achieved by assigning a unique index to each individual category and type. Two columns are created to record these index representations. Once columns are created, the columns "Question Category" and "Question Type" are removed.

3.2.3 Dataset for all answered and unanswered questions

Recall that we have a data set that includes all the answers provided by the students. However, to answer our first research question - which questions are less likely to be answered by students - it is essential to not only analyze the responses, but also those unanswered ones. Therefore, our goal here is to generate a dataset that contains a record of every question, for every student, regardless of whether the question was answered or not.

Luckily, by using the available data in the answers.csv file, we can easily generate a comprehensive dataset, where:

- Each row represents the answers to the students
- Each column represents a question from the course provided to the student.

This dataset can help us identify which questions were answered and which were skipped by each student. In Figure 3.9, you can find an example of the general schema of this dataset.

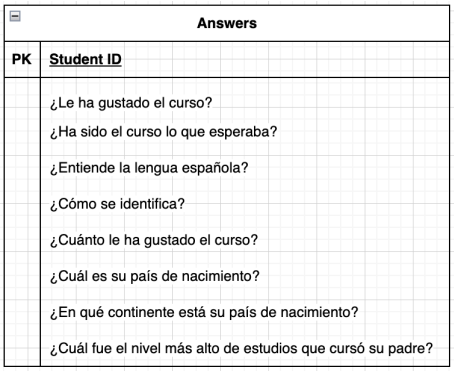


Figure 3.9: Example Schema of the Questionnaire Response Dataset (Including Unanswered Questions)

The above schema offers only a general idea of the structure of the dataset. In a real-world scenario, the dataset will contain a significantly larger number of columns corresponding to the many questions included in the course. Due to the target audience of our platform, the interface is designed to be user-friendly; hence, the questions are presented in a multiple-choice format. As a result, each cell in the dataset represents the index of the selected option for a given question. In case a student does not provide an answer to a specific question, the corresponding cell will be recorded with a value of -1. In Figure 3.10, you

can see a subset of this dataset, consisting of different responses to a question from 27 different students. We can see that the 26th student did not provide an answer to this question, which is noted as -1 in his row.

	O	P
1	Participant ID	Si quisiera usar un ordenador o portátil, ¿sería posible para usted disponer de uno?
2	885aeef7-ad0e-4752-bf9d-7f9f5bd0999c	3
3	1fae8ed8-9758-4b9c-8a02-0cb9526705ae	3
4	c4f14bc2-fc8c-442b-9e8c-6bec821a2f3c	2
5	4159f760-24ab-4ffb-8853-5b53f1e8ea77	0
6	760cad06-9e99-49a2-8a24-03ac2b963bfa	1
7	9a5d34ec-85a9-49f3-af56-3fd150410342	2
8	b5c94ca8-b665-4680-b7dd-b1043a1f9c13	0
9	139f3a02-f5e4-435e-85ba-244b6514c237	2
10	c4f2c139-2d7d-491c-8d99-73f89a059dd6	1
11	0b96e956-8a10-4722-a8cc-fa21e4f2cd78	0
12	d331769f-5765-4eb3-847f-6042354b0486	1
13	b5852481-12b4-4402-9813-a0bc0be83724	3
14	65af29f3-5774-4953-86ee-708baff6ab87	0
15	b223faf1-fb08-4556-8c02-20988e344086	2
16	9f34fa1c-6ba1-479c-bca8-96bd7b911acf	0
17	bd9a1f53-0203-4eef-95ab-01a23d6e8e22	0
18	577a159c-63ef-40ad-a558-8432cb97fafd	2
19	bb594769-ccfb-4e71-af22-609449418fcf	0
20	adfe9cb9-8855-420b-a72b-61918fd7ce52	1
21	1db57c60-f443-4488-aaf2-e0026ff4401c	1
22	0093df20-83ba-4c9c-8377-f7bffbd299c1	0
23	425783c3-aaf7-4b8e-adce-a5caf4af1ae0	3
24	d8d5ed25-5e37-4011-b66f-4911aaba89a1	0
25	9c595a29-100b-446d-bd1e-9e219b8d7159	0
26	befbbeb5-371f-4463-91ea-b7123839d3f7	-1
27	72464847-c5fb-4962-8052-352c2d57f2d2	0

Figure 3.10: A portion of the data illustrating how unanswered questions are recorded.

Now that we have described the schema of this dataset, let's focus on the algorithm behind it. Using the data from answers.csv, we begin by extracting a list of all distinct questions available on the course. By comparing this list with the record of the actual responses submitted by each student (878 participants in total), we can easily determine which questions were left unanswered. The steps to generate on this dataset consist of the following:

1. Generate the column names:
   (a) Collect a list of questions that are relevant to our analysis. This can be a subset of questions, based on the question category. For instance, in the next steps we will be applying clustering techniques based on the profile details of the participants, hence we only collect questions that are categorized as "Profile" (See section .. for further details).
   (b) Append the list with an additional column name, called "Participant ID".
2. Generate a list of unique participants.
3. Iterate through the list of participants and collect the answers to the selected list of questions from the answers.csv file. If the answer is not found

in the CSV file, it is recorded as -1. To improve the modularity of the code, a module is defined and used during each iteration. The algorithm responsible for answer collection is defined as a pseudocode (see Algorithm 1).

Algorithm 1 An algorithm generating participants row

Require: Current participant ID p, Answer dataset A, Column names C
Ensure: A dictionary R representing the responses of a participant known to exist in the dataset A

```
R ← {}                                                  ▷ Initialize empty map
id ← p["ID de participante"]
A_p ← filter rows in A where "ID de participante" equals id
for all current_column_name in C do
    if col = "Participant ID" then
        R[col] ← id
    else
        a ← find row in A_p where "Texto de la pregunta" = current_column_name
        if a exists then
            R[col] ← a["Selección en la respuesta"]    ▷ Get the answered recorded
        else
            R[col] ← −1                                ▷ Mark unanswered question
        end if
    end if
end for
return R
```

3.2.4 Time answering the questions

In Table 3.3, there is one attribute that is eye-catching. That is, the timestamp associated with each event. Furthermore, we have an entry event (when a participant begins interacting with the question) and an exit event (when the participant leaves the question). The interaction performed on a specific question by a specific participant is uniquely identified using the activity ID. Therefore, if we use the activity ID, we can accurately retrieve both the entry time and the exit time for each question interaction. If we subtract the exit time from the entry time, we can generate the time a student spends on a specific question.

This can be a valuable time-spent metric, serving as a valuable behavioral feature in our analysis. Spending more time on a question may suggest difficulty, uncertainty, or increased engagement. On the other hand, if a student spends a short time but still provides an answer, it may suggest that the question was straightforward. However, if the student spends less time and does not provide an answer, it simply implies a quick skip or disengagement. Therefore, by taking into account these time intervals, we can explore whether there's a correlation between time spent on a question and the likelihood of that question being answered.

From Table 3.3, we can also see that the column "Occurred in" is defined as a string column. In Elixir, there is a specific module called *DateTime*, which

provides a specialized data structure for handling timestamps, along with some useful functions that we can use directly to compute the time differences in seconds. To make use of these capabilities, the string values in the "Occurred in" column must first be converted into a *DateTime* data structure. This conversion involves parsing the time components (such as hours, minutes, milliseconds, etc) from the string. To achieve this, regular expressions are used (See figure 3.11 for the regular expressions). A TimeParser module is implemented to collect these components to facilitate the creation of valid DateTime objects.

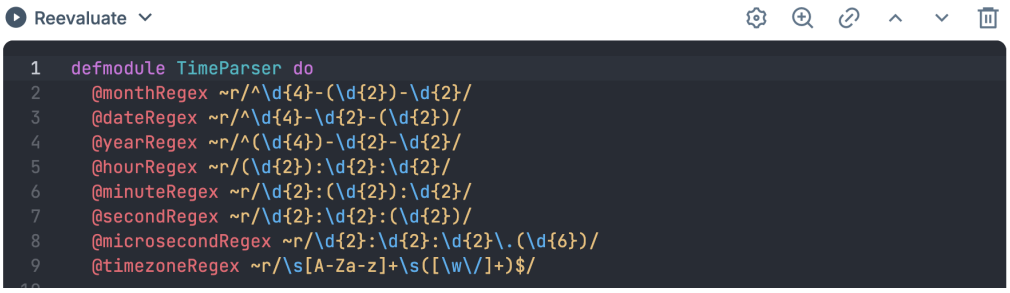


Figure 3.11: A screenshot representing the regular expressions used to extract different time components.

3.2.5 Enrollment time

Our second research question focuses on student performance, specifically aiming to predict whether a student is likely to complete the course successfully. One feature that we can take into account is the enrollment time of the student. For instance, students who have a longer enrollment time may be more motivated and have higher commitment. However, if a student didn't finish the course, while having a long enrollment period, it could also suggest a possible loss of interest or disengagement over time. As a result, by incorporating enrollment timing, together with all the other predictive features, we can enhance the accuracy of our models and potentially identify at-risk students sooner. By identifying at-risk students sooner, we can introduce timely interventions, such as personalized support or reminders, to increase students' overall course completion rates.

In Table 3.2, two columns are relevant for calculating enrollment time. Those are: "Course start timestamp" and "Last activity timestamp". By generating the differences between these two timestamps, we can compute the student's enrollment duration. However, these timestamps are recorded as strings. Therefore, *DateTime* module is used to convert them into proper datetime objects, and together with a regular expression defined in Figure 3.11 we can parse different time components directly from the string.

3.2.6 Time watching video and following the course

In Table 3.2, the time spent watching course videos and engaging with the course material is recorded as string values in the format `HH:MM:SS`. Although this in-

formation can be understood by humans using a string format, it is not ideal for predictive modeling. To facilitate numerical analysis and enable the use of time-based features during our analysis on the second research question, we convert these time values into integer representations. More specifically, the total number of seconds. The conversion is implemented using a custom TimeParser module, in combination with the *DateTime* module, which contains an embedded function that we can use directly to convert timestamps into seconds.

3.2.7 Generate target values

Another data that we are missing in the datasets available is the target values that we can use during supervised learning. To investigate our two main research questions, we need to define two distinct target variables:

1. **is_answered** flag: A boolean value indicating whether or not a student will answer a specific question. This value can be generated easily using the dataset described in section 3.2.3, since the dataset contains both answered and unanswered information.
2. **is_completed** flag: A boolean value indicating whether or not a student will finish the course. This information cannot be collected from the dataset. However, as we can see in Table 3.2, we have a column called "At least 7h30", this data can be used as the threshold to approximate course completion. Since the values are stored as Boolean (True/False), we convert them into a numerical format, with 1 representing completion and 0 representing non-completion.

3.3 Clustering

To effectively address our research questions, it is essential to gain profile information such as age, parental education, and sociodemographic background. However, this information highly depends on the student's responses via questionnaires, and some of them do not provide such details and choose to skip them. As a result, we end up with a significant portion of missing data. This is particularly challenging, as these profile characteristics are often critical for understanding and predicting participant behavior, including their likelihood of responding to specific questions and their overall performance.

To address this challenge, we propose to use clustering techniques on the available data to profile questions. The data we used follows the structure as described in section 3.2.3. After applying clustering algorithms, we will have labels, where each label represents a group that our students belong to. These labels serve as a proxy indicator of their overall profile. Our objective is to uncover a common participant profile that reflects shared characteristics.

These cluster labels will then be incorporated as an additional input feature in our decision tree, random forest, and logistic regression modeling steps in the following sections. This kind of incorporation enables us to retain profile information in a meaningful way, helps us compensate for missing profile information

when addressing our research questions.

When building the clustering model, two key questions need to be addressed:

- Which cluster algorithm should we use? As there are many, including K-means, DBSCAN, or K-mode.
- How many clusters do we need to create to meaningfully differentiate between distinct groups of students?

The answers to the above questions will be answered throughout this section.

3.3.1 K-Mode

Let's first concentrate on which clustering algorithm to use in our case. As described in section 3.2.3, the questions are designed in a user-friendly manner; hence, the questions are presented in a multiple-choice format. Therefore, each cell in the dataset represents the index of the selected option for a given question. This structure makes the dataset inherently categorical, since the values represent categories rather than quantities.

Due to this categorical format, standard clustering algorithms that rely on distance metrics like Euclidean distance (e.g., K-means) are not suitable. This is because Euclidean distance assumes continuous data and is sensitive to magnitudes that do not hold for categorical variables. If we apply this method to categorical data, we will end up with inaccurate similarity calculations during clustering and, ultimately, meaningless clusters. To address this problem, we employ the K-modes clustering algorithm [18], which is specifically designed for handling categorical data.

The steps for this K-mode algorithm can be summarized as the following steps:

1. Pick K observations, this k amount will be the number of the cluster after applying the algorithm. These K data points will act as the initial centroids of the cluster, an initial guess of what each cluster's "representative" profile looks like.
2. Calculate the dissimilarities and assign each observation to its closest cluster, with the lowest dissimilarity.
3. Define the mode of each cluster by identifying the most frequently occurring category for each feature among the data points assigned to that cluster.
4. Repeat steps 3–4 until either the cluster assignments stop changing (convergence) or a predefined maximum number of iterations is reached. In our implementation, we set the maximum number of iterations to 500. We tested the algorithm with various maximum iteration limits and found that convergence was consistently achieved within 500 iterations. As a result, we set the maximum number of iterations to 500 in our implementation to ensure stability.

In Step 3, we mentioned that we compute the dissimilarities between each data point and the current cluster centroids. In a traditional clustering algorithm,

this step typically involves calculating the Euclidean distance. However, since we are handling the K-mode clustering algorithm for categorical data, we use a different metric, called the matching dissimilarity.

Instead, we use a different metric tailored for categorical data: the matching dissimilarity. This metric simply counts the number of mismatched attributes between two categorical vectors (i.e., how many features differ). For example, if two data points differ in 3 out of 5 features, their dissimilarity score would be 3.

Matching dissimilarity

The matching dissimilarity is an effective distance metric used for comparing categorical data. It quantifies how different two data points are by counting the number of features in which they differ. For example, if two datapoints differ in 3 out of 5 features, their dissimilarity score will be 3.

The mathematical representation of such computation is defined as follows:

$$D(x, y) = \Sigma_{i=1}^{r} \delta(x_i, y_i) \tag{3.1}$$

where:

- x and y are two data points (categorical vectors)
- x_i and y_i are the values of the i-th feature in x and y, respectively
- r is the total number of features
- $\delta(x_i, y_i)$ is an indicator function defined as:

$$\delta(x_i, y_i) = \begin{cases} 1 & \text{if } x_i \neq y_i \\ 0 & \text{if } x_i = y_i \end{cases}$$

During our implementation phase, we defined a specific module for the K-modes algorithm. To ensure the clarity and reusability of the code. Every component of the algorithms, such as initialization of modes, computation of matching dissimilarity, reassignment of clusters, and mode updates, was implemented as a separate function in the module. This modular structure further ensures the overall logic remains transparent and maintainable.

To illustrate the structure and logic of our implementation, the following pseudocode presents an overview of the K-modes algorithm (see Algorithm 2).

Algorithm 2 K-Modes Clustering Algorithm

```
procedure KMODES_CLUSTER(data, k, max_iter)
    centroids = randomly selecting k data points from data
    iterate(data, centroids, k, max_iter)
end procedure

procedure ITERATE(data, centroids, k, iter)
    if iter = 0 then
        return assign_clusters(data, centroids)
    else
        assignments = assign_clusters(data, centroids)
        new_centroids = recompute_centroids(assignments, k)
        if new_centroids = centroids then
            return assignments
        else
            iterate(data, new_centroids, k, iter - 1)
        end if
    end if
end procedure

function ASSIGN_CLUSTERS(data, centroids)
    for each row in data do
        Calculate dissimilarity between row and each centroid using matching_dissimilarity
        Assign row to the cluster of the centroid with minimum dissimilarity
    end for
    return list of tuples (row, cluster_label)
end function

function RECOMPUTE_CENTROIDS(assignments, k)
    Group assignments by cluster labels
    for each cluster do
        Compute the mode
    end for
    Ensure exactly k centroids by padding if any clusters are empty
    return list of updated centroids
end function

function MATCHING_DISSIMILARITY(x, y)
    return D(x, y) = Σ_{i=1}^{r} δ(x_i, y_i)
end function
```

3.3.2 Feature Selection

Feature selection is crucial in unsupervised learning, specifically before applying clustering. Irrelevant or noisy features can distort the true structure of the data, leading clustering algorithms to form misleading or less meaningful groups. Our feature selection strategies are:

- **Remove responses** for the following question: "¿Cuál es su país de nacimiento?". One advantage of using Livebook is its ability to instantly visualize the dataframe, together with some important statistics such as the minimum, maximum, and mean values. When examining data under the column "¿Cuál es su país de nacimiento?" (What is your country of birth?), we can see that the vast majority of entries are recorded as -1, which indicates to the fact that the student did not respond to it. This observation can further be proved by the column's mean value of -1, reinforcing the fact that missing responses dominate this field (See Figure 3.12).

Figure 3.12: A screenshot displaying how the statistical values are displayed.

By ordering the data in descending order using the Dataframe interface from Livebook, we confirmed that only two students responded to this question (See Figure 3.13). Due to this extremely low response rate, this column lacks the variance and representation needed to contribute meaningfully to any downstream analysis. Therefore, we remove this column from our dataset to focus on more informative features that better represent the overall student population.

- **Merge responses** for the following questions: "¿Cuál fue el nivel más alto de estudios que cursó su madre?" (What is the highest education level completed by your mother? and "¿Cuál fue el nivel más alto de estudios que cursó su padre?" (What is the highest education level completed by your father?). These two information are related, as both reflect the parental education background, which is a possible factor that can influence a student's academic environment. Rather than treating these features separately, we merge them into a single attribute using the maximum value between the two. This will capture the highest educational value of the parent.

Figure 3.13: A screenshot showing only two students selected the first option for the multiple choice question of "¿Cuál es su país de nacimiento?".

Combining these features not only reduces noise data but also helps in simplifying the feature space, leaving n-1 features, where n is the total number of features available for the clustering algorithm. By doing so, we retain the information related to parental education while promoting a more compact data representation.

Chi-squared test

Chi-squared test is another feature selection strategy, it is used to demonstrate how strongly two features are correlated with each other. In case there is a strong correlation, it suggests that the features carry overlapping information. Keeping both in such cases does not add value and may even introduce unnecessary complexity. To address this, we adopt the following strategy: In case two features are highly correlated with each other, we retain the one with fewer missing or unanswered values, represented by -1. This ensures that we preserve the most complete source of information while improving the overall quality of the data used for clustering.

After the removal and the merging of the features as described in section 3.3.2, this statistical analysis has been applied in order to see whether we need to remove any features, as described in the above-mentioned strategy. However, we did not end up removing any features. The process of ending up to this conclusion is described throughout this sub-section.

Just like all the other statistical tests. We should define the null hypothesis and the alternative hypothesis:

- Null Hypothesis (H_0) = There is no association between two categorical values.

- Alternative Hypothesis (H_1) = There is an association between the two categorical values.

Now we proceed to generate the chi-squared test. For this, we used the *chi2_contingency* function of scipy.stats module in Python. And we visualize the generated p-value using the Matplotlib module. The figure 3.14 shows the generated matrix, where each cell represents the p-value between two different features.

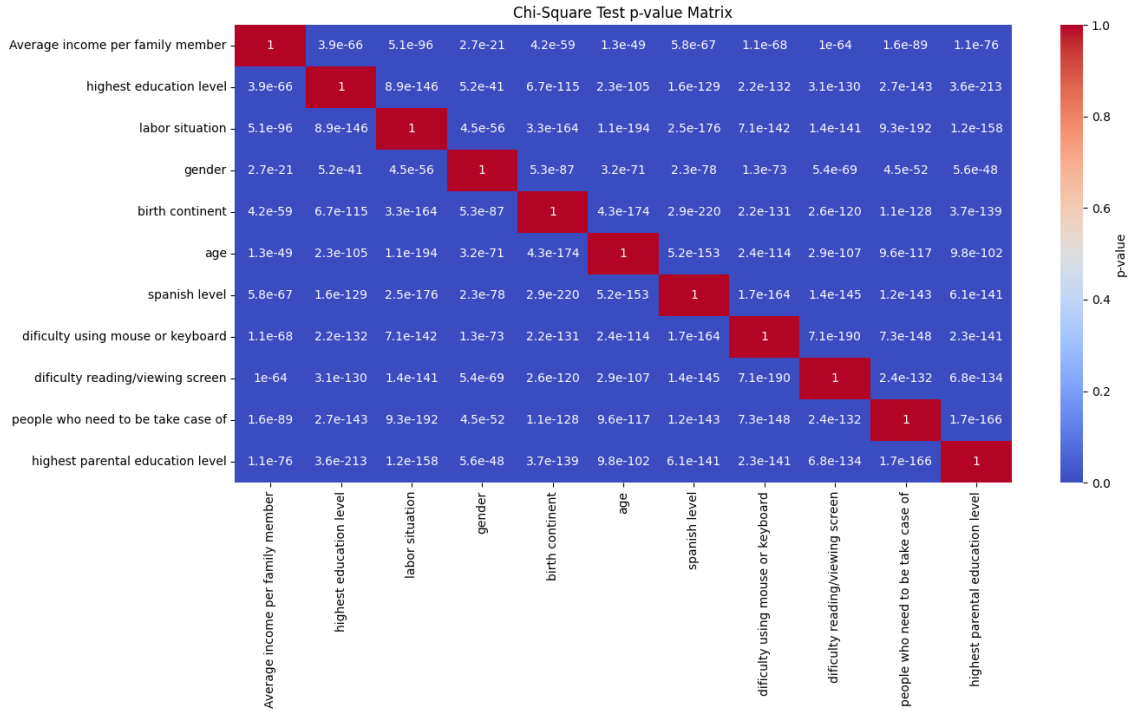


Figure 3.14: Matrix showing the p-values between different features.

From the matrix, we can see that no p-values exceed 0.05, which is a commonly used threshold. Therefore, we reject the null hypothesis. We conclude that there is a significant association between all pairs of features, suggesting they are not independent. And if we were to remove them, following the strategy mentioned before, we would end up with no data and potentially lose meaningful information. As a result, we choose to retain all features and processes without elimination.

3.3.3 Number of cluster

After describing the algorithm used and the feature selection steps, we now move on and determine the number of clusters we need to use for our algorithm. To do this, we need to perform a series of tests for different numbers of clusters and evaluate the quality of the clustering (See Table 3.4).

k (Number of cluster)	Silhouette score
2	0.45
3	0.25
4	0.15
5	0.13
6	0.12
10	0.15

Table 3.4: Table showing different k selection with its corresponding Silhoutte score.

The metric that we are using for the evaluation is the silhouette score. The value of this score ranges from -1 to 1.

- -1 = Incorrect clustering
- 0 = Overlapping clusters
- 1 = Excellent clustering - the clusters are well separated.

To test what the optimum number of clusters is, we have tried to execute the clustering algorithm with different k-values ($k = 2$ to $k = 6$, as well as $k = 10$). The reason for selecting these k values is that we observed that the silhouette score starts to drop after $k = 2$. We also include $K = 10$ to see whether a higher number of clusters might still produce a meaningful structure.

From Table 3.4, we observe that as the number of clusters increases, the lower the silhouette score. And we encounter a better silhouette score when we have 2 clusters. Therefore, we select 2 as the optimal number of clusters.

3.4 Sub-Clustering

From section 3.3.3, the initial clustering reveals only two distinct groups. If we look closely at the data, we have divided the students into 2 groups:

- **Inactive students**: This group represents those who are inactive and answered only one or two profile questions.
- **Active students**: This group represents those who are active and answered almost all profile questions.

However, this clustering does not provide meaningful insight into the characteristics of active students. Therefore, we filter out the inactive group and apply the same clustering algorithm again, but this time we focus on the active students. We called this technique sub-clustering.

To determine the number of clusters, we perform again with the same procedure as described in section 3.3.3. Where a different number of clusters are selected and their corresponding silhouette score are recorded.

From Table 3.5, we observe again that as the number of clusters increases, the

k (Number of cluster)	Silhouette score
2	0.28
3	0.15
4	0.10
5	0.08
6	0.07
10	0.07

Table 3.5: Table showing different k selection with its corresponding Silhoutte score.

lower the silhouette score. We select 2 as the optimal number of clusters again, due to its highest silhouette score among others. After performing these sub-clustering, we managed to obtain 3 clusters in total: one representing inactive students, and the other two groups are generated from this sub-clustering. The description of the resulting profile for these groups of students will be presented in Chapter 4.

In general, from both Table 3.5 and Table 3.4, we can see a low silhouette score, which is due to the categorical nature of our input data. Categorical variables have limited combinations, leading to many duplicate or near-duplicate points

3.5 Decision Tree

To answer our first questions highlighted in Section 1.3, we have chosen to use a decision tree. The input data are derived from the dataset described in section 3.2.3. We transform the dataset so that each row corresponds to a specific question presented to a student. The binary target value indicating whether a question was answered by the student is also derived from that dataset. As described in section 3.3, the labels generated after applying the clustering algorithm are one of our input features for our model. This allows us to get an insight into how different student profiles influence the chance of answering the question. Additionally, other important characteristics are the category and type of questions. Hence, these two features are also considered part of the input features for our decision tree model. Apart from these, another metric that we find interesting is the time a student spends on questions, which can be extracted from the events.csv file. In summary, we have the following input features:

- Type of questions: Some question types may be more approachable for certain students. For example, some people might find it difficult to read text, hence they find it easier to answer Likert-style questions.
- Category of questions: The category of the questions might also affect the chance of not answering. For instance, questions about personal background or profile information tend to be skipped more often due to privacy concerns.
- Time spent on video in seconds: A student who spends a significant amount

of time on the video but fails to answer the question may not have fully understood the question.

- Cluster label: Different profiles of the student might affect the performance when answering the questionnaires. For instance, a non-Spanish speaker might skip the questions more often due to language barriers.

Since the majority of our features are categorical, we chose Decision Tree as one of our predictive models. Decision trees natively support categorical features without the need to convert them. Furthermore, decision trees do not assume a numerical or linear relationship between feature values and the outcome, which makes them more suitable for our categorical data. Another reason for selecting a decision tree is the low dimensionality of input data. As we can see from the above bullet points, we have in total of 4 input features. Decision trees do not require complex modeling or transformations to learn meaningful patterns from these 4 input features.

3.5.1 One-hot encoding

One-hot encoding is a method for converting categorical variables into a binary format, where each unique category is represented by a binary column with a value of 1 indicating its presence and 0 indicating its absence. At first, this technique was not applied, and we got the following results from Figure 3.15 and Figure 3.16.

From Figure 3.15, we can see that the decision tree performs exceptionally well on this dataset. The model correctly classified 99% of all samples. It detects class 0 perfectly (recall 1.00), with very few false positives (precision 0.98). For class 1, it's nearly perfect with precision 1.00 and recall 0.99. Furthermore, the weighted average score is about 0.99 for all metrics. This reflects the fact that there is balanced performance across classes, with no major signs of overfitting toward one class.

	precision	recall	f1-score	support
0	0.98	1.00	0.99	3848
1	1.00	0.99	1.00	7446
accuracy			0.99	11294
macro avg	0.99	1.00	0.99	11294
weighted avg	0.99	0.99	0.99	11294

Figure 3.15: Precision, Recall, F1-score of decision tree model without applying one-hot encoding.

From Figure 3.16, we see that the most influential factor is the time a student spends answering the questions. Generally, if a student spends less time on the questions, there is a higher chance of skipping. While the other characteristics, such as type, category, and cluster label, are less influential factors. But this is not enough; we also want to understand how different question types, question

categories, and groups of people affect the chance of unanswered responses. To achieve this, we use one-hot encoding, which allows the model to assess the impact of each category separately, rather than treating the entire column as a single combined index.

```
Category_Index          0.001042
Type_Index              0.001156
final_total_time        0.996745
cluster                 0.001058
```

Figure 3.16: Feature importance scores of the decision tree model without applying one-hot encoding

The resulting dataset after applying one-hot encoding can be seen in Figure 3.17, Keep in mind that this screenshot only displays a portion of the dataset.

is_access	is_feedback	is_inactive	is_likert	is_mcq	is_older	is_profile	is_satisfaction	is_selfpre_selfpost	is_usability	is_younger
0	1	0	0	1	1	0	0	0	0	0
0	0	1	0	1	0	1	0	0	0	0
0	0	0	1	0	0	0	0	0	0	1
0	0	0	1	0	1	0	0	1	0	0
0	0	0	0	1	1	1	0	0	0	0
0	0	0	1	0	1	0	0	0	0	0
0	0	0	1	0	1	0	1	0	0	0
0	0	0	1	0	1	0	0	1	0	0
0	0	1	1	0	0	0	0	1	0	0
0	0	0	1	0	1	0	0	1	0	0
0	0	0	1	0	1	0	0	1	0	0

Figure 3.17: Screenshot displaying a part of the dataset after applying one-hot encoding.

3.5.2 Pythonx

Unlike Python, Elixir is still evolving in the area of machine learning and data analytics. Therefore, some useful functions are still missing. In the case of developing a Decision Tree model, Elixir has an *Evision* module that can be used to train the model and compute some basic metrics such as testing errors and the model's accuracy. However, it lacks built-in support for more advanced functionalities. For example, if we want to determine how much each feature contributes to the model's predictions and to visualize the entire tree structure. On the other hand, Python has the *sklearn.tree* module, where the scoring of feature importance is calculated directly during the model's training phase. It also has a *plot_tree* function that can be used to directly visualize the tree structure.

One of the advantages of using Elixir Livebook is the possibility of executing Python code using the *Pythonx* module. This feature allowed us to leverage

Python's powerful libraries without having to implement these capabilities from scratch in Elixir. Specifically, we were able to use the *sklearn* module in Python to calculate feature importance and visualize the decision tree directly.

3.6 Random Forest

Random Forest is another model we chose to investigate for our first research question. As we highlighted in Section 2.2. Random Tree Forest is an ensemble learning method. This type of model combines the predictions of multiple decision trees to produce a more accurate and stable result (see Figure 3.18). Random Forest model is often chosen to improve accuracy and avoid overfitting, although our decision tree does not exhibit low accuracy or overfitting, we still want to test it to see whether any improvement is still possible.

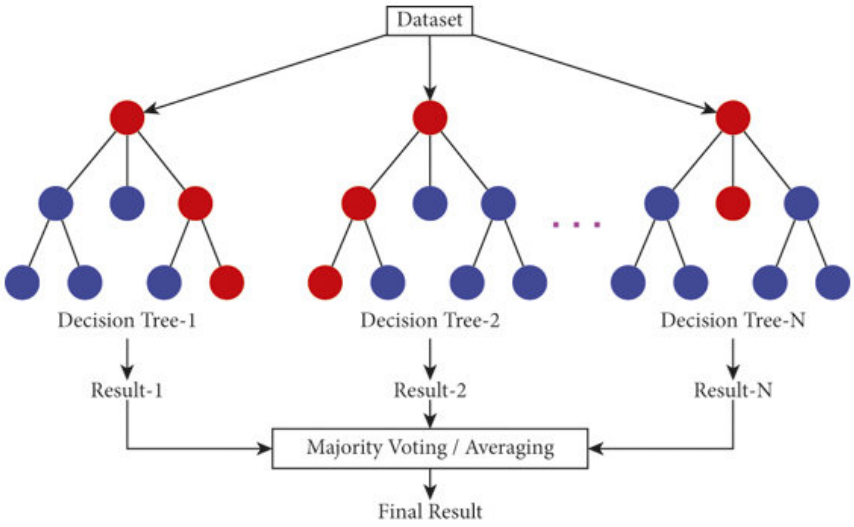


Figure 3.18: Simple diagram explaining how Random Forest works. [19]

Both Random Forest and Decision Trees are used to answer our first research question. Furthermore, we will compare the results of both to determine whether Random Forest brings any improvements compared to our decision tree model. To ensure a fair comparison, the input dataset is the one for our decision tree model, as described in Section 3.5, including the use of one-hot encoding.

3.6.1 One-hot encoding

As described in Section 3.5.1, One-hot encoding is also used to gain an understanding of how different question types, question categories, and groups of people affect the chance of unanswered responses. The resulting dataset is the same as displayed in Figure 3.17 since we use the same dataset compared to the one used for the decision tree.

3.6.2 Pythonx

Pythonx module is also used here to ensure the execution of Python code in Elixir Livebook. The same reasons are used as described in section 3.5.2, where we can easily use the $Sklearn$ module in Python to calculate the importance of the features after the model has trained.

3.7 Logistic Regression

Logistic regression is used to explore our second research question. More specifically, we will be using a Logistic Regression model to determine whether the student will finish the course or not.

$$Y = \begin{cases} 1 & \text{if } finish \\ 0 & \text{if } not\_finished \end{cases}$$

Due to this binary nature of the predictive value Y, the Logistic regression model is chosen. This type of model is developed from the Linear Regression model, where the output values are continuous. However, unlike Linear Regression, the output value of Logistic Regression is binary, due to the use of the sigmoid function.

$$\sigma(z) = \frac{1}{1 + e^{-z}} \tag{3.2}$$

where:

- z = $w \cdot X + b$, the linear regression formula. where:
  - w = the weight determined by the linear regression model.
  - X = input feature matrix
  - b = intercept

This sigmoid function (see Equation 3.2) will map any set of independent variables of real value to a value between 0 and 1 (See Figure 3.19), which represents the likelihood that the dependent variable is either 0 or 1 (See Equation 3.4 and 3.3).

$$P(Y = 1) = \sigma(z) \tag{3.3}$$

$$P(Y = 0) = 1 - \sigma(z) \tag{3.4}$$

In terms of the implementation of the model, we use *Scholar.Linear.LogisticRegression* module, which contains built-in functions for direct training and testing.

3.7.1 Dataset

The input data is derived from the Participation.csv file, including other metrics such as the enrollment time of the student, which is derived from the Events.csv file. As described in section 3.3, the labels generated after applying the clustering algorithm are one of our input features for our model. This allows us to get

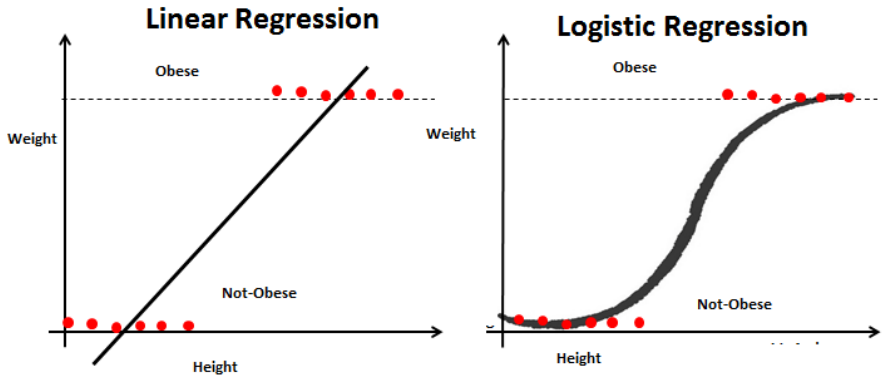


Figure 3.19: Simple diagram explaining the differences between linear regression and logistic regression. [20]

an insight into how different student profiles influence the chance of finishing the course. To summarize, we have the following input features:

- Number of activity: The number of activities performed by the students. These include answering questions or pausing videos. This metric serves as an indication of student engagement.
- Enrollment duration in seconds: The total time a student has been enrolled. A student who has been enrolled for a long time but fails to complete the course is considered an inactive student.
- Number of answers provided: A high number of responses provided to the questionnaire reflects a high level of participation.
- Number of videos reproduced: The number of videos reproduced by the student also reflects the level of student participation.
- Last topic finished: This is a flag indicating whether the most recent topic is finished by the student. If a student finishes the last topic, it shows they're actively progressing.
- Time following the course in seconds: The amount of time also reflects on the level of the student's engagement.
- Cluster label: Different profiles of the student might affect the chance of finishing the course. For instance, a young student may drop out if the content is perceived as too easy.

The majority of our features are numerical, and only two features are categorical (Last topic finished and cluster label). Logistic regression works well with numerical inputs, so having mostly numerical features makes it a good fit. Furthermore, we have a small dataset, consisting of only 875 rows. Logistic regression is a simple, low-variance model that tends to perform well on smaller datasets, reducing the risk of overfitting.

3.7.2 Standardized data

By observing the data of our input features, we observe that our dataset contains variable values that are different in scale. For instance, "Last topic finished" consists of 0 or 1 values; on the other hand, the "number of activity" of the student ranges from 10 to 5099. As a result, our model will be biased toward the large-scale features, even if they're not more important. To overcome this challenge, standardization is performed on the dataset before feeding it to the model. To mitigate this issue, we apply standardization to the dataset before feeding it to the model for training. We use the *Scholar.Preprocessing* module, which includes built-in functions responsible for performing this standardization.

3.7.3 One-hot encoding

After learning the importance of one-hot encoding as described in section 3.5.1. We have chosen to apply one-hot encoding to the cluster labels. This allows us to assess how different student profiles influence the likelihood of course completion.

3.7.4 Evaluation Metrics

Since the application is written in Elixir, it is better to incorporate the code of this study into the project. To efficiently evaluate the quality of the model, we decided to generate the confusion matrix, precision, recall, F1 score, and accuracy. A custom module called *Metrics* is defined to generate all these evaluation metrics.

3.7.5 Number of iterations

The performance of the model is highly dependent on the number of training iterations. Each iteration updates the model's weights w using some version of gradient descent (See Figure 3.20). If you train the model with a few iterations, the model may not have converged, and the weights w are still far from optimal. Therefore, to determine the optimal number of iterations, a series of tests was performed. To evaluate the model's performance in each iteration, the accuracy of the model is generated and computed.

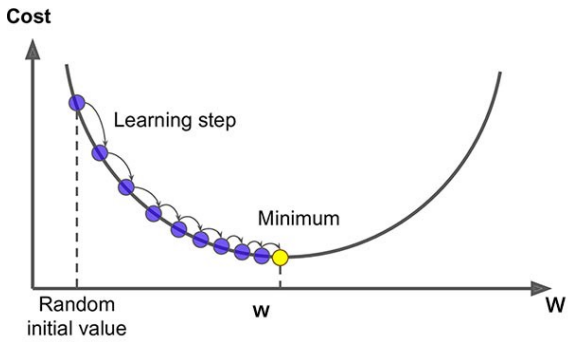


Figure 3.20: Simple diagram explaining the process of gradient descent. [21]

Number of iterations	Accuracy (%)
100	56
200	46
300	56
400	74
500	44
600	56
700	57
800	44
900	44
1000	44

Table 3.6: Table showing different accuracy scores for different numbers of training iterations.

From Table 3.6, we can see that the best accuracy occurs at 400 iterations, reaching 74%. And we also observe that the model already converges when it reaches 400 iterations, since the accuracy does not improve with more iterations. In fact, after 400 iterations, accuracy tends to decline. This decrease in accuracy can be explained by overfitting, where the model learns the training data too well and fails to generalize to the testing set. It would be beneficial to plot the loss against iterations to see if the model is actually diverging, and most importantly, to visualize the overfitting pattern. However, since the primary goal of this experiment is to identify the optimal number of training iterations, we conclude that 400 iterations is the most suitable choice and proceed with this configuration.

Chapter 4

Result and Analysis

In Chapter 3, we described all the methodology used throughout the study. We are now going to analyze the results that we have obtained using the methodologies described. Since our study focuses on answering two research questions described in Section 1.3, we will use the analysis of our results to answer our research questions.

4.1 Clustering Result

We first focus on the clustering results. As described in Section 3.3, after applying the clustering algorithm for the first time, we successfully divided the less active students from the active students. Then we applied the clustering algorithm again to these active students. So in the end, we ended up performing sub-clustering, where we have managed to divide the active students into two different groups. During this session, we will describe the result of this sub-clustering and what profile each of the sub-clusters has.

4.1.1 Sub-cluster 1

In this session, we will describe the overall profile for this sub-cluster labeled as 1. To do so, several bar charts have been generated, each illustrating the distribution of a specific profile characteristic within this cluster.

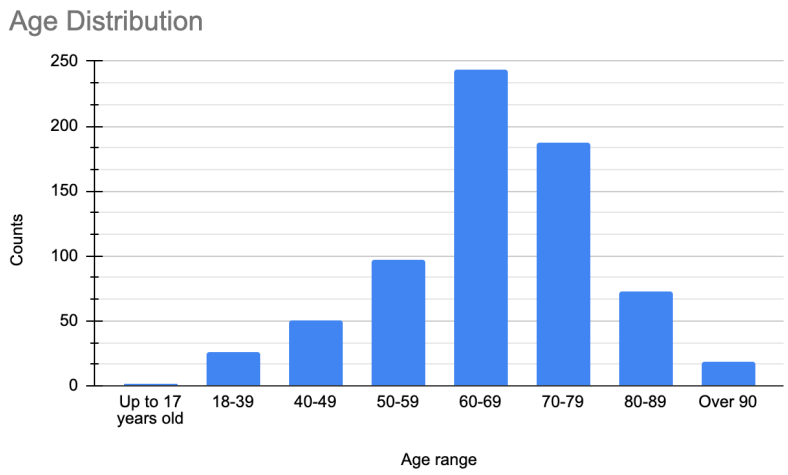


Figure 4.1: Age distribution within sub-cluster labeled as 1

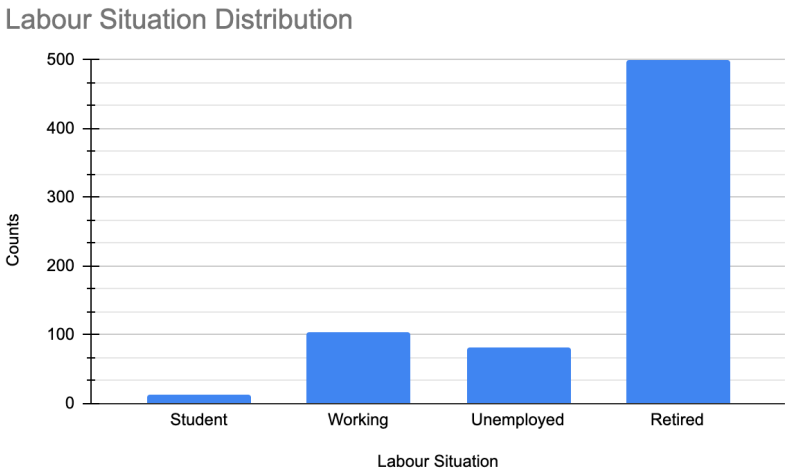


Figure 4.2: Distribution of students within sub-cluster 1 based on labor situation.

From Figure 4.1, we can see that the plot is skewed to the right, showing that most people in this group are over 50 years old. In terms of the labor situation, we can see that most people are retired and only a few of them are working (see Figure 4.2). This further corresponds to the age range belonging to this group of people.

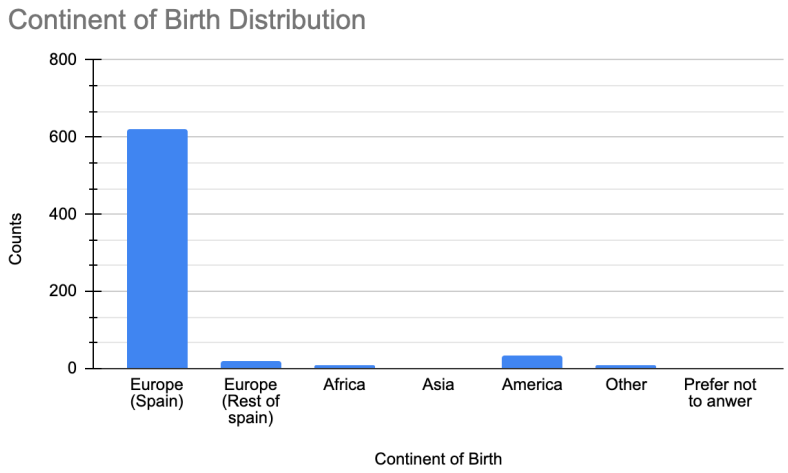


Figure 4.3: Continent of birth distribution within sub-cluster labeled as 1

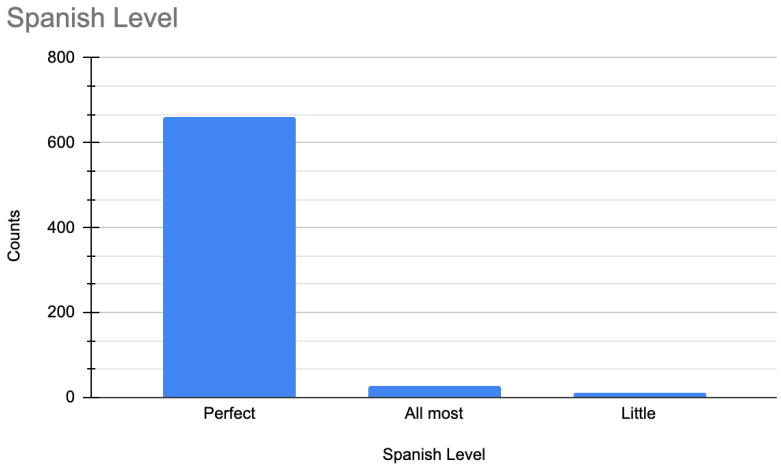


Figure 4.4: Distribution of students within sub-cluster 1 based on Spanish level.

From Figure 4.3, there is a high number of students from Spain. This further corresponds to the fact that most students have perfect knowledge of Spanish (see Figure 4.4).

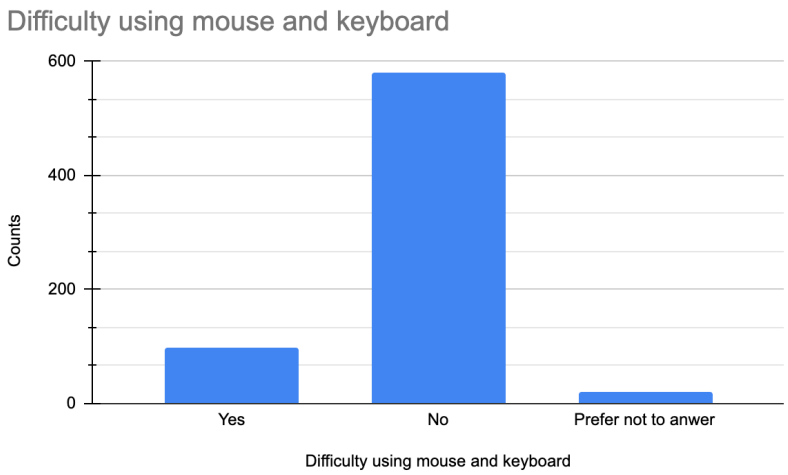


Figure 4.5: Distribution of students within sub-cluster 1 based on reported difficulties using a mouse or keyboard.

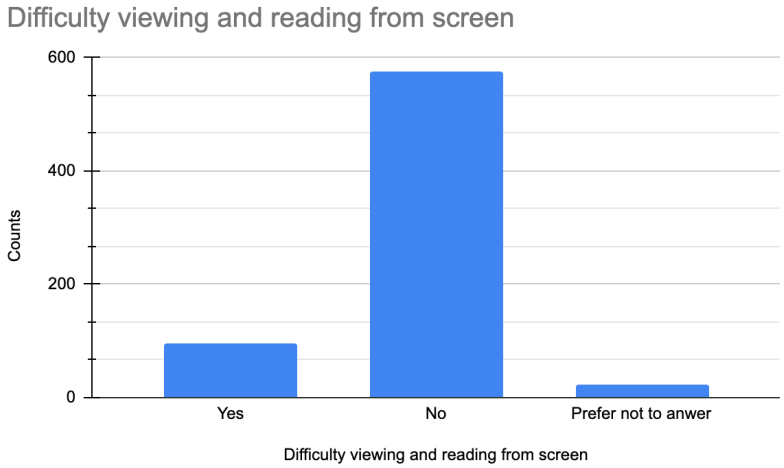


Figure 4.6: Distribution of students within sub-cluster 1 based on reported difficulties reading or viewing from the screen.

From Figure 4.5, we observe that only a few students have difficulty using the mouse and keyboard; most students do not experience any issues with them. Furthermore, from Figure 4.6, we can also observe that most students do not experience difficulty viewing and reading from the screen.

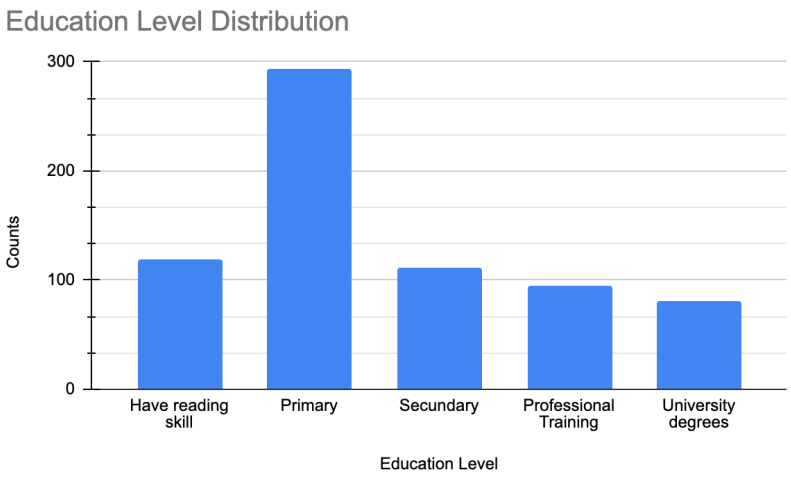


Figure 4.7: Distribution of students within sub-cluster 1 based on the education level.

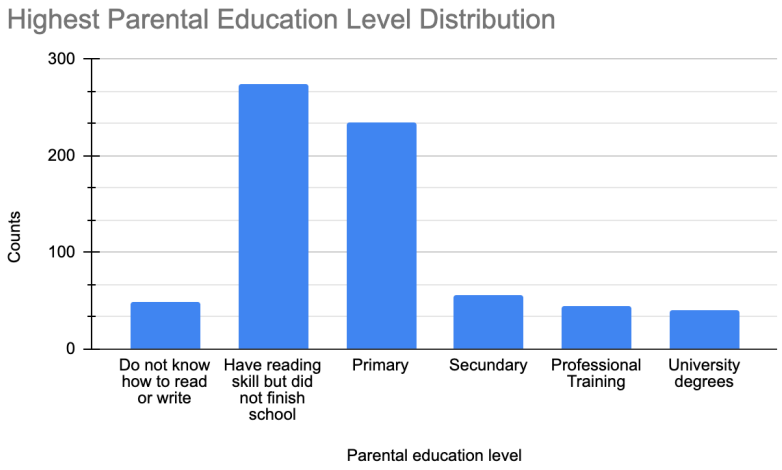


Figure 4.8: Distribution of students within sub-cluster 1 based on the highest parental education level.

From Figure 4.7. We see that most students in this group completed Primary education. In terms of the highest education level of their parent, we can see that most of their parent did not finish their study (see Figure 4.8). This pattern is consistent with the educational background commonly found in older generations.

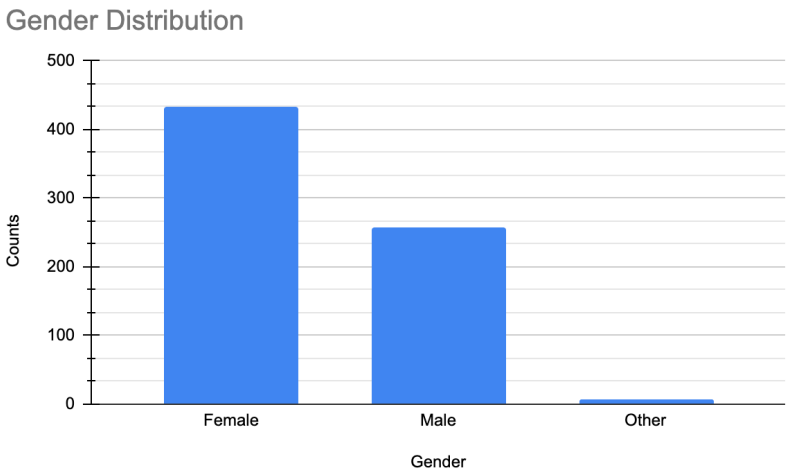


Figure 4.9: Distribution of students within sub-cluster 1 based on gender.

Within this sub-cluster, we can also find more female participants compared to male participants (see Figure 4.9).

We found that most students in this sub-cluster are not responsible for taking care of someone, such as a child, elderly family member, or someone with a disability (see Figure 4.10). Notably, these students also report a personal income per family member between 400 and 800 euros (see Figure 4.11).

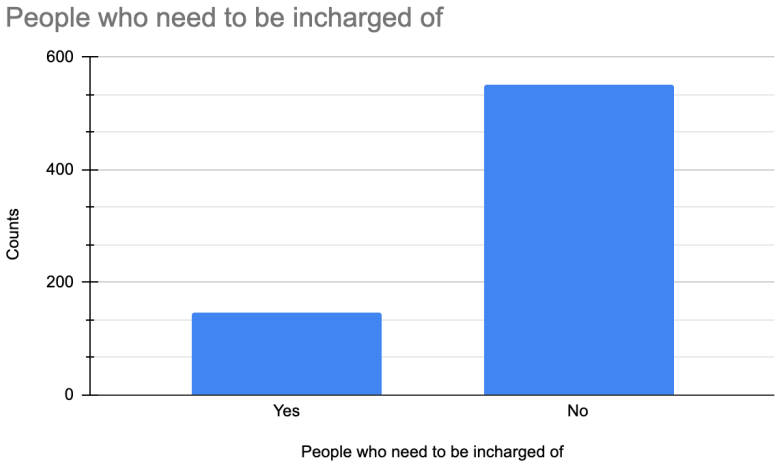


Figure 4.10: Distribution of students within sub-cluster 1 based on whether they need to be responsible for someone.

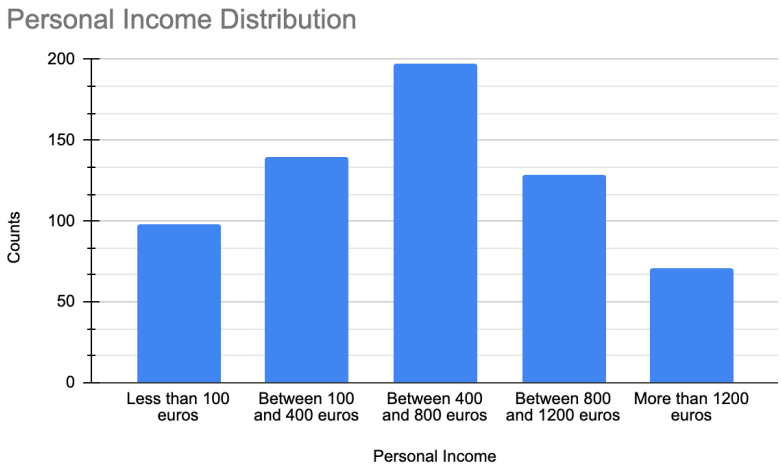


Figure 4.11: Distribution of students within sub-cluster 1 based on personal income per family member.

Summary

To summarize, students who belong to this category have some outstanding characteristics, such as those who are over 50 years old, retired people, and native Spanish speakers with fluent Spanish proficiency. Furthermore, these students tend not to face difficulties when using a mouse or keyboard, nor do they struggle with reading from a screen. Both their own and their parents' education levels are predominantly low.

4.1.2 Sub-cluster 2

In this session, we will describe the overall profile for this sub-cluster labeled as 2. The same as in Section 4.1.1, several bar charts have been generated, each illustrating the distribution of a specific profile characteristic within this cluster.

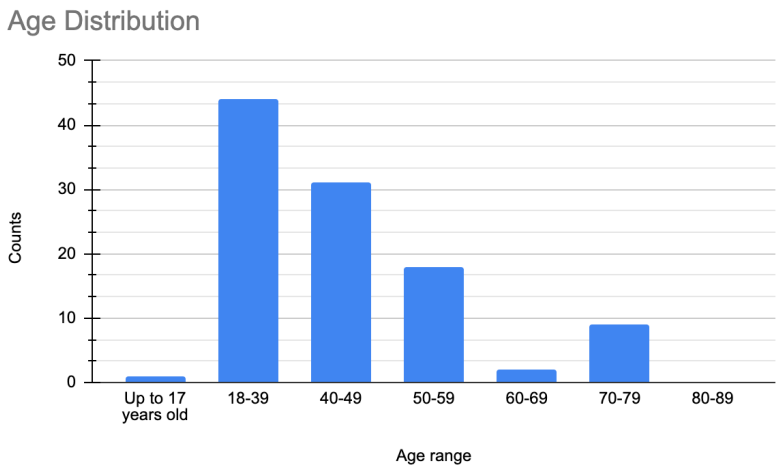


Figure 4.12: Age distribution within sub-cluster labeled as 2

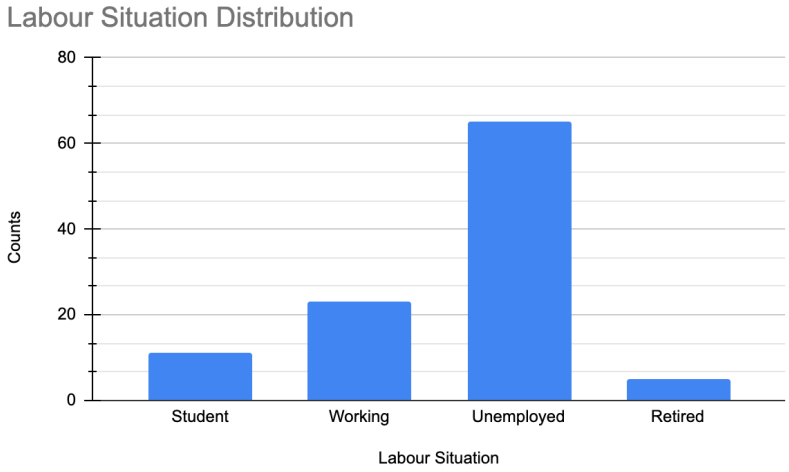


Figure 4.13: Distribution of students within sub-cluster 2 based on labor situation.

From Figure 4.12, we can see that the plot is skewed to the left, showing that most people in this group are under 50 years old. In terms of the labor situation, we can see that most people are unemployed and only a few of them are retired (see Figure 4.13). This further corresponds to the age range belonging to this group of people.

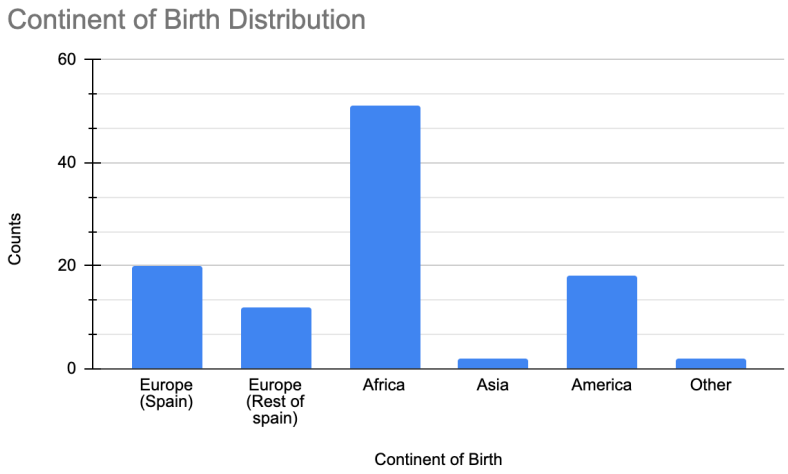


Figure 4.14: Continent of birth distribution within sub-cluster labeled as 2

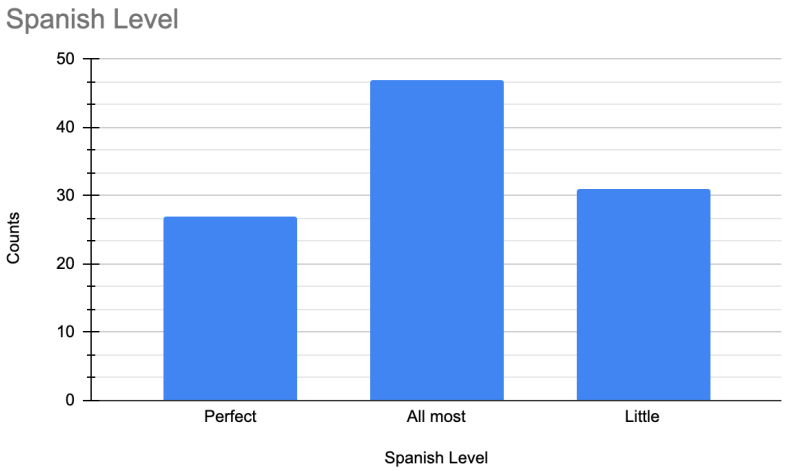


Figure 4.15: Distribution of students within sub-cluster 2 based on Spanish level.

From Figure 4.14, there is a high number of students from Africa. This further corresponds to the fact that most students do not have perfect knowledge of Spanish (see Figure 4.15).

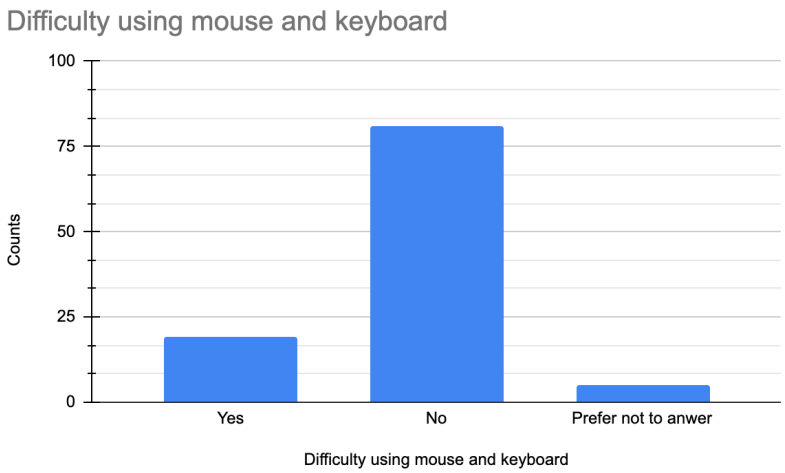


Figure 4.16: Distribution of students within sub-cluster 2 based on reported difficulties using a mouse or keyboard.

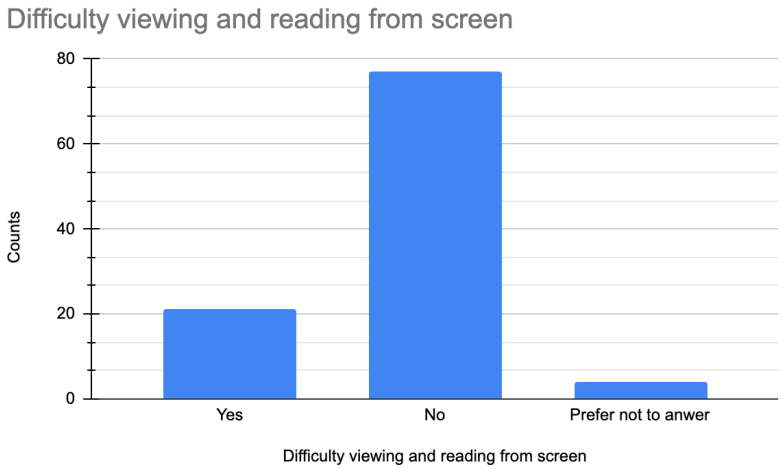


Figure 4.17: Distribution of students within sub-cluster 2 based on reported difficulties reading or viewing from the screen.

From Figure 4.16, we observe that only a few students have difficulty using the mouse and keyboard; most students do not experience any issues with them. Furthermore, from Figure 4.17, we can also observe that most students do not experience difficulty viewing and reading from the screen.

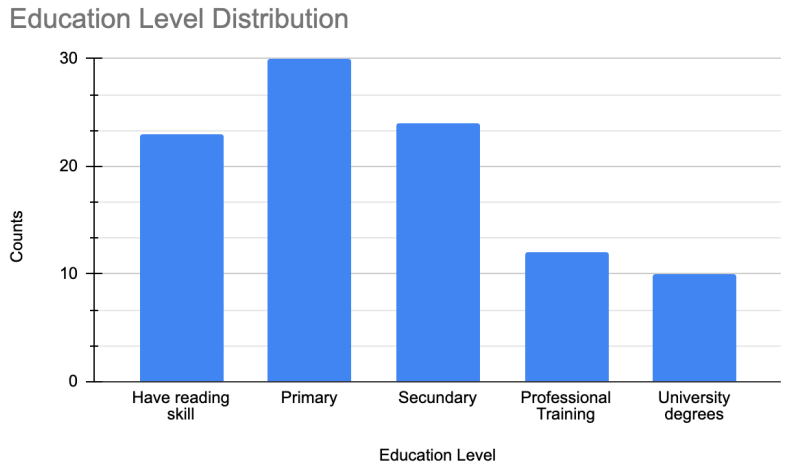


Figure 4.18: Distribution of students within sub-cluster 2 based on the education level.

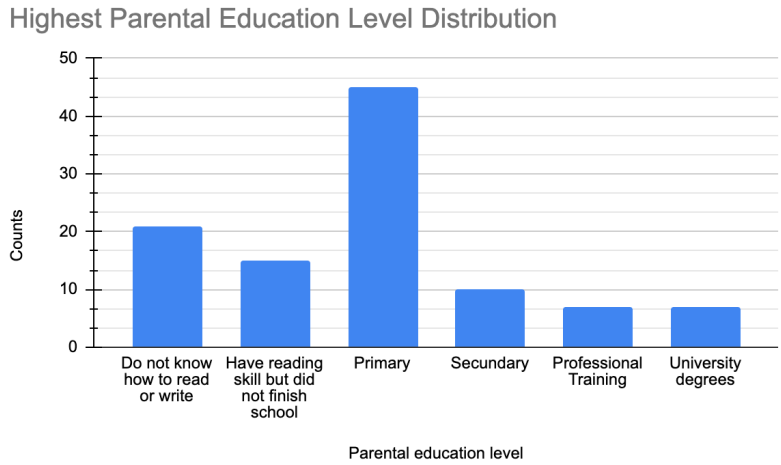


Figure 4.19: Distribution of students within sub-cluster 2 based on the highest parental education level.

The education level is almost evenly distributed (see Figure 4.18). However, we still have a significant number of students who completed Primary. Furthermore, we can see a decent number of students who completed secondary education. In terms of the highest education level of their parent, we can see that most of their parent also completed primary education Figure 4.19).

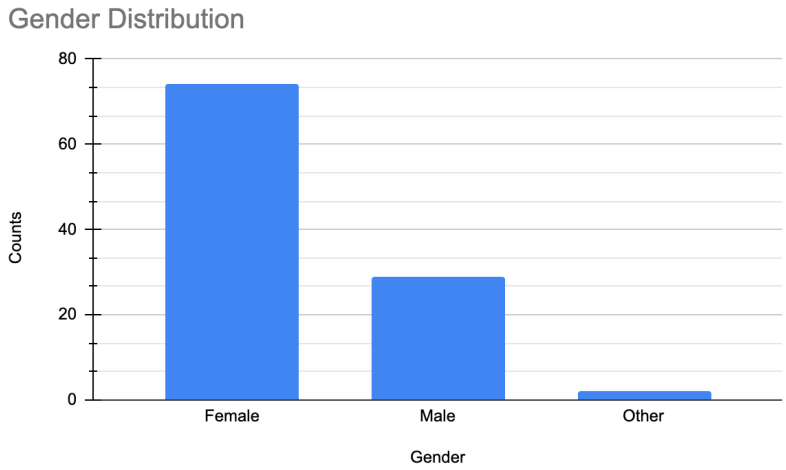


Figure 4.20: Distribution of students within sub-cluster 2 based on gender.

Within this sub-cluster, we can also find more female participants compared to male participants (see Figure 4.20).

We found that approximately 45% of the students in this sub-cluster are still responsible for taking care of someone, such as a child, elderly family member, or someone with a disability (see Figure 4.21). Notably, these students also report a personal income per family member below 400 (see Figure 4.22). This observation further indicates a generally low quality of life among these students.

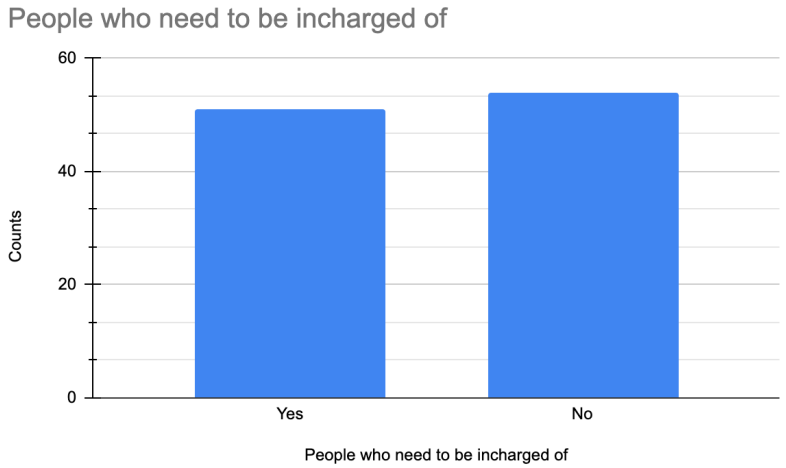


Figure 4.21: Distribution of students within sub-cluster 2 based on whether they need to be responsible for someone.

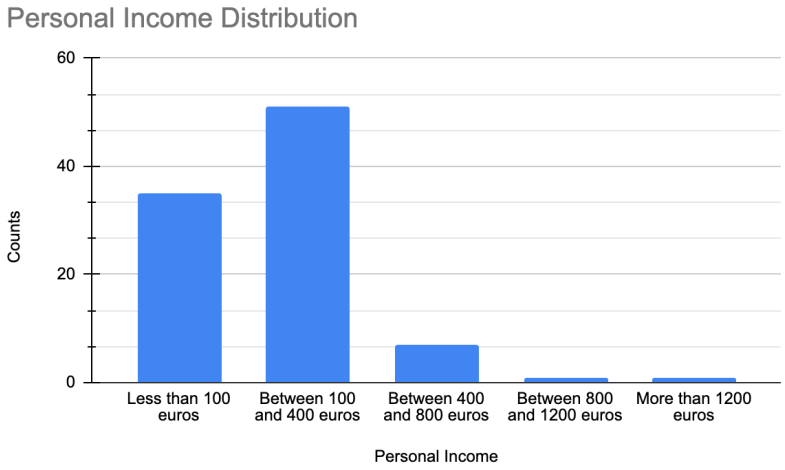


Figure 4.22: Distribution of students within sub-cluster 2 based on personal income per family member.

Summary

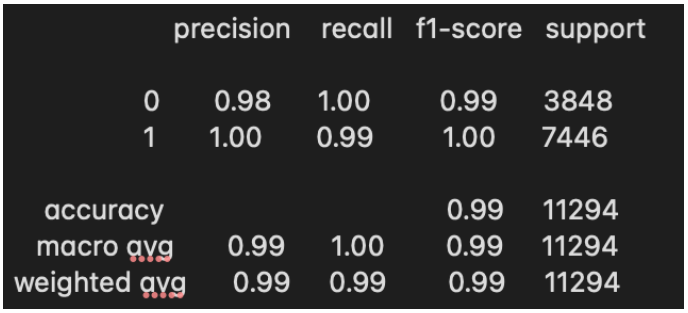
To summarize, students who belong to this category have some outstanding characteristics, such as those who are under 50 years old, non-retired people, and non-Spanish people who do not speak Spanish at a perfect level. Furthermore, these students tend not to face difficulties when using a mouse or keyboard, nor do they struggle with reading from a screen.

Compared to the students in Section 4.1.1, these students achieve a higher education level due to a moderate amount of students who completed their secondary education level. However, they have a low personal income that reflects a low quality of life, especially when many of them need to be in charge of someone else.

4.2 Will the students answer the questions in the questionnaire?

4.2.1 Decision Tree Result

From Session 3.5, we describe the methodology used and the final decision to use one-hot encoding to facilitate further detailed analysis. Throughout this sub-session, we will analyze the result of our final decision tree model.



	precision	recall	f1-score	support
0	0.98	1.00	0.99	3848
1	1.00	0.99	1.00	7446
accuracy			0.99	11294
macro avg	0.99	1.00	0.99	11294
weighted avg	0.99	0.99	0.99	11294

Figure 4.23: Result metrics of Decision Tree model

From Figure 4.23, the final accuracy of our model is 99%. Overall, the model performs well, detecting questions that are not answered perfectly, with very few false positives (precision 0.98). When detecting questions that are answered, it has a precision score of 1.00 and a recall of 0.99, indicating good performance. Furthermore, the weighted average score is about 0.99 for all metrics. This reflects the fact that there is balanced performance across classes, with no signs of overfitting toward one class.

Features	Importance Score
Total time spent (in seconds)	0.996745
Likert Type questions	0.001604
Access questions	0.000842
Profile questions	0.000233
Feedback questions	0.000175
sub-cluster 1	0.000141
MCQ type question	0.000127
selfpre and selfpose question	0.000096
Usability questions	0.000015
Inactive student	0.000012
Satisfaction questions	0.000010
sub-cluster 2	0.000000

Table 4.1: Table showing feature importance, ordered from most important to least important features

Now we concentrate on the features. From Table 4.1, we see that the most influential factor is the time a student spends answering the questions. Generally speaking, if a student spends more time answering the questions, there is a higher chance that the questions will be answered.

In terms of the type of questions (Table 4.1), those that are Likert-type questions are proven to be more important compared to multiple-choice questions. In other words, the model assigns higher importance to Likert-type questions (0.001604) than to multiple-choice questions (0.000127), indicating that the presence of Likert scale questions may have a greater impact on whether a student responds to the questionnaire. Furthermore, if we look closely at the decision tree struc-

ture, we can see that whenever is_likert is greater than 0.5, the result is that the student does not respond (class = 0) (see Figure 4.24). This further proves that Likert scale questions have a negative impact on the likelihood of answering the questions. Since our students are mostly elderly people, this negative impact can be explained by unfamiliarity with emojis.

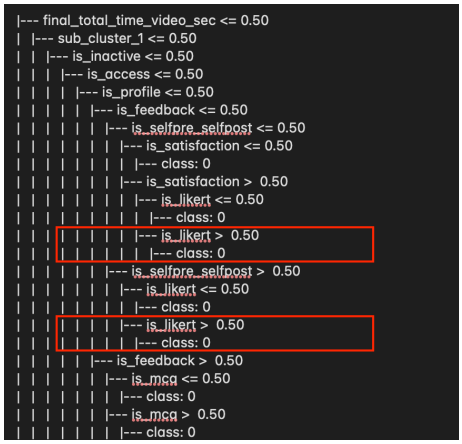


Figure 4.24: A small session of the Decision Tree, it shows the node where is_likert is greater than 0.5, leading to a class = 0

Now we focus on different question categories. The top 3 most influential categories are: Access (0.000842), Profile (0.000233), and Feedback (0.000175) (Table 4.1). They all play a minor role. By looking at the decision tree structure, whenever is_access is greater than 0.5, it will be classified as answered (class = 1) (see Figure 4.25). Therefore, Access category questions are more likely to be answered by students. The answers to access questions consist of yes or no answers; therefore, they are straightforward and easy-to-answer questions, which explains why elderly people are more responsive. Unlike the access category, the profile category has a slightly negative impact on the likelihood of answering the questionnaire (see Figure 4.26). This can be explained by the fact that our students may be more hesitant to answer questions about personal details. In the decision tree, we observe that in all branches where is_feedback appears, it does not lead to a response in the decision paths (see Figure 4.27). Thus, their presence is associated only with non-response paths in the model. This behavior can be explained by looking closely at the content of feedback questions, we observe that most questions are Likert-scale and are queried at the end of the course. The elderly students may be unfamiliar with Likert-scale questions and may already feel disengaged after the course finishes.

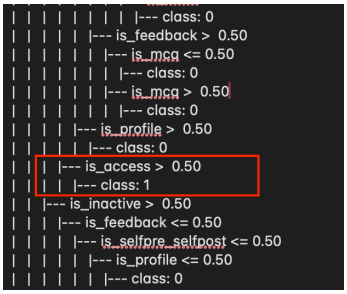


Figure 4.25: A small session of the Decision Tree, it shows the node where is_access is greater than 0.5, leading to a class = 1

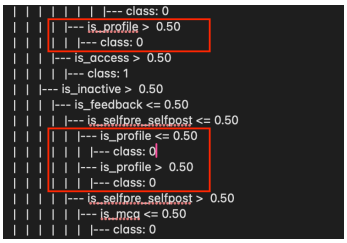


Figure 4.26: A small session of the Decision Tree, it shows the node where is_profile is greater than 0.5, leading to a class = 0

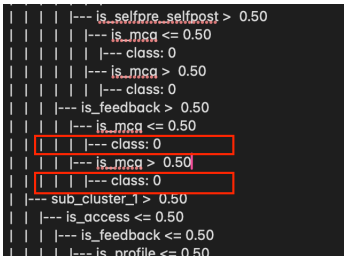


Figure 4.27: A small session of the Decision Tree, it shows that the node where is_feedback is greater than 0.5, will lead to a class = 0 in any way.

Among the 3 groups of people, those who are classified under sub-cluster 1 have a greater impact on the response rate. In other words, those who follow the characteristics highlighted in Section 4.1.1 have a greater impact on the likelihood of responding to the questions. And if we look at the decision tree data structure, we can see that whenever sub_cluster_1 is greater than 0.5, the model always predicts non-response (class = 0) (see Figure 4.28). Therefore, we conclude that this group of students harms the likelihood of the response rate. In other words, older students tend not to respond to the questionnaire; this can be due to several reasons, such as cognitive, survey fatigue, etc.

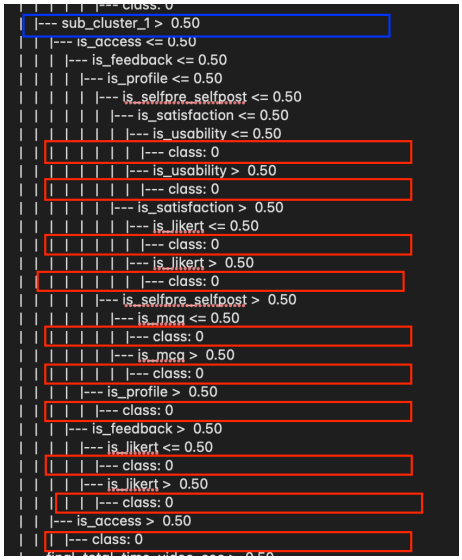


Figure 4.28: A small session of the Decision Tree, it shows that the node where sub_cluster_1 is greater than 0.5, will lead to a class = 0 in any way.

4.2.2 Random Forest Result

Random Forest is another selected model to address our first research question. Throughout this sub-session, we will go through the results obtained.

	precision	recall	f1-score	support
0	0.98	1.00	0.99	3848
1	1.00	0.99	1.00	7446
accuracy			0.99	11294
macro avg	0.99	1.00	0.99	11294
weighted avg	0.99	0.99	0.99	11294

Figure 4.29: Result metrics of Random Forest model.

From Figure 4.29, the final accuracy of our model is 99%. The overall performance is similar to our decision tree model as described in session 4.2.1. The

weighted average score also shows the fact that there is balanced performance across both classes, with no signs of overfitting.

Features	Importance Score
Total time spent (in seconds)	0.862752
Inactive student	0.037136
Feedback questions	0.021552
Profile questions	0.014927
is_older (sub-cluster 1)	0.014864
Satisfaction questions	0.013155
MCQ type question	0.011138
Likert type questions	0.009742
selfpre and selfpose question	0.006562
Access questions	0.003867
is_younger (sub-cluster 2)	0.003081
Usability questions	0.001225

Table 4.2: Table showing feature importance, ordered from most important to least important features

From Table 4.2, we can see that the time spent answering the question is still the most important factor. The overall feature importance scoring is higher compared to our result of Decision Tree (see Table 4.1) Based on the computed SHAP value, we can see that when the time spent on questions is higher (indicated by red color), it will have a positive impact on the model outcome (see Figure 4.30).

Among the two different types of questions, MCQ has a greater impact on the likelihood of responding to the questions. And if we look closely at the computed SHAP value, there are more red dots located to the right(see Figure 4.30). This means that when the question is MCQ type, the model is more likely to predict that a question to be answered. In case of Likert scale questions, it will have a more neutral impact, since the dots are mixed and close to 0. This conclusion reflects on the results of the decision tree, where we highlighted a stronger negative impact when the questions are Likert scale.

Additionally, the model identifies inactive students who contribute negatively to the model outcome. This means that those students who are inactive will for sure not answer the questions.

Those questions that are categorized as Feedback and Profile are shown to be positively impacted based on SHAP values (see Figure 4.30). This conclusion is different compared to the decision tree results. The result of Random Forest reflects that in certain scenarios, users do respond to feedback questions, especially if they are otherwise active. This supports the broader insight that maintaining student engagement throughout the course plays a critical role in increasing response rates. Furthermore, we can also see from the corresponding SHAP value of is_access features, it shows a greater positive impart on the chance of responding, this reflects the same observation from the decision tree results.

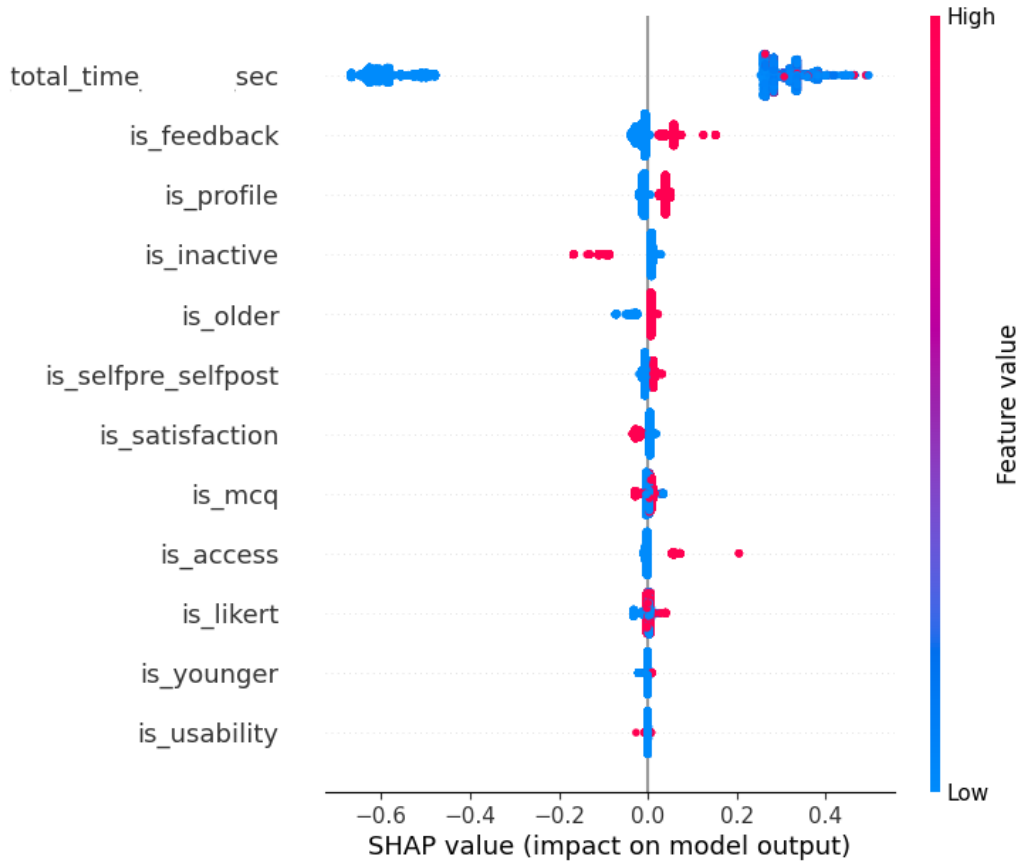


Figure 4.30: Graph showing the SHAP values for each feature. This plot displays both feature importance and the direction of impact.

4.2.3 Summary

To address our first research question - which questions will be answered by students - we developed two different tree-structured models: The Decision Tree model and the Random Forest model. Both of these models gained good performance. After combining the results of both models, we came up with several conclusions and possible strategies for improving the response rate of the questionnaire.

First of all, we realized that inactive students are those who will highly affect on the response rate. Therefore, it is important to keep students engaged throughout the online course.

Furthermore, we have figured out that Likert scale questions are less likely to be answered by students, especially among our students who are mostly over 60 years old (Figure 3.3). This type of question consists of emojis, which is an unfamiliar and abstract format for this group of audience. Therefore, it can create a comprehension challenge and might not be seen as straightforward to

answer, compared to multiple-choice questions. Furthermore, questions from access category are more likely to be answered by students due to their easy-to-answer questions.

In the previous sections, we have mentioned that time spent on questions might also be an important factor, since it may reflect on those students who have difficulties understanding the questions. However, both models showed that longer video time is positively associated with answering; hence, we can exclude the situation where the students spent a long time on questions but failed to answer them.

In summary, to improve the questionnaire response rate, we need to focus on re-engaging those who are inactive. To do this, a reminder can be set. Furthermore, Likert scale questions can be removed, especially for elderly people. It's also important to consider that our students are more likely to respond to simple, straightforward questions, such as true or false items, especially in the questions that fall in access category.

4.3 Will the students be able to complete the course?

During this session, we will describe the result for our second research question highlighted in Session 1.3. To address this question, we have decided to use a logistic regression model, due to the reasons highlighted in section 3.7.1.

4.3.1 Logistic Regression

The result metrics are shown in Table 4.3. Our logistic regression model ends up with an accuracy of 74%. This accuracy shows a moderate accuracy, possibly due to a small amount of data (955 rows of data in total). Our model has a good precision score, reflecting the fact that whenever the model predicts a student completes the course with success, it is usually correct. However, it has a relatively low recall, indicating that the number of false negatives is relatively high. In other words, students who would complete the course, but the model predicts they would not. Our F1 score of 0.72 suggests a fairly balanced and useful model.

Metrics	Score
Precision	0.78
Recall	0.67
F1_Score	0.72
Accuracy	0.74

Table 4.3: Table showing Precision, Recall, F1_Score and Accuracy of our Logistic Regression model.

Now we focus on the input features of course completion. After training a logistic regression model, we receive a set of coefficients. Each coefficient in logistic

regression represents the influence of a feature on the log-odds of different outcomes. As you can see in Figure 4.31, we have two sets of coefficients, one set corresponding to the outcome of a student not completing a course, and the other set corresponding to the outcome of a student completing a course. For example, we have the feature related to the time following the course, and we have a coefficient of 6.56. It can be expressed as follows: For each 1-unit increase in time following the course, the log-odds of completing the course increase by 5.56. If we think about this in terms of odds, it will be around 259.82 ($e^{5.56}$). Recall that we have the following conversion from odds to probability:

$$P = \frac{odds}{1 + odds} \tag{4.1}$$

By substituting the odds value, we have the following probability values.

$$P = \frac{259.82}{1 + 259.82} = 0.99 \tag{4.2}$$

Therefore, 1 more second following the course will increase the predicted probability of completing the course to 99%

Figure 4.31: Screenshot showing the coefficients after training a logistic regression model.

To gain an overall summary of how influential each input feature is on the predicted outcome. The net coefficient is computed, representing the fact that the relative influence of each feature on predicting class 1 (e.g., student completing the course), compared to class 0 (student not completing the course).

From Figure 4.32, we can conclude that we have two very strong indicator for the prediction of course prediction. The time following the course and the enrollment duration. More specifically, students who spent more time following the course have 68062 times higher odds of completing. Furthermore, students enrolled longer have 245 times higher odds of completing. Then we have the number of events triggered by the students, which is a moderate positive indicator of success. Each additional activity event increases the odds of completing the course by around 21.6%.

DataFrame 9 entries

Show 10 Prev 1 of 1 Next

	Feature names	net coefficient	odd_ratio
	unique: 9	min: 0.0121666789054878	min: 1.0122410058975
	top: Actividad (#eventos)	max: 11.128178596496582	max: 68062.2890625
	top freq: 1	mean: 1.88	mean: 7590.51
	nulls: 0	nulls: 0	nulls: 0
1	Actividad (#eventos)	0.19555377960205078	1.2159842252731323
2	Enrolment duration (sec)	5.501346588134766	245.0216522216797
3	Numero de respuestas	0.02608799934387207	1.0264313220977783
4	Numero de videos reproducidos	0.02882683277130127	1.0292463302612305
5	is_inactive	0.01216667890548706	1.012241005897522
6	is_older	0.01230931282043457	1.012385368347168
7	is_younger	0.012201070785522461	1.0122758150100708
8	last topic finished	0.012326359748840332	1.0124026536941528
9	time_following_course_second	11.128178596496582	68062.2890625

Figure 4.32: Screenshot of the Dataframe showing the net coefficient for each input features.

Now, let us focus on the 3 different groups of people (is_inactive, is_older refers to subcluster 2 and is_younger refers to subcluster 1). If we were to rank the net coefficient of these groups of people from lower to highest, we end up with the following ranking:

1. is_inactive
2. is_younger (sub-cluster 1, see Section 4.1.1)
3. is_older (sub-cluster 2, see Section 4.1.2)

From this ranking, we can conclude that being inactive has the worst impact on student success among the three. Then we have people from sub-cluster 2 who has the more impact on the probability of success and are more likely to complete the course among other groups of people. This is because they are finding this course useful, and they are actually learning with this online course. Unlike younger people (students categorized as sub-cluster 1) who might think that this course is teaching something that they might have already known and find it too easy, leading them to drop the course halfway through. In general, this observation is expected, and it is within our hypothesis.

In summary, we found the top 3 most important factors affecting the likelihood of students completing the course. These factors are long-term factors. Therefore,

these findings do not directly inform how we can improve the early detection of at-risk students. In other words, those who are a high risk of dropping out.

Chapter 5

Limitations and Future Work

During this study, we have managed to look into both of our research questions. Despite the insights gained, we have faced certain limitations. Therefore, throughout this chapter, we will describe these limitations and their corresponding solutions. These solutions were not carried out during our study due to time constraints.

In section 3.3, we described that we used the K-mode clustering algorithm. Despite the perfect result that we have obtained in this algorithm, we encountered certain challenges during its execution. That is, to get an optimal result after clustering where we obtain a good silhouette score, we needed to execute the cluster algorithm multiple times. This is mainly due to the first step of the algorithm. As you can see in Section 3.3.1, we decided to pick k data points randomly as initial cluster centroids, which can lead to variability in the final clustering result. One possible way to optimize this algorithm is to introduce novel algorithms [22]. Where the selection of initial centroids is based on 3 factors:

- Weighted average density: This factor will help us select the representative points.
- Distance Outlier Factor: This factor helps us to avoid a noisy center.
- Distance to existing centers: This factor ensures well-separated initial centers.

After taking these 3 factors in when performing the first initialization of centroids, we will ideally end up with an ideal result at once (see Figure 5.1 for an example of an ideal result).

From the results of Logistic Regression, we have figured out that the top 3 most influential factors that have a positive influence on the course completion are: Total time following the course, total number of activities triggered by the students, and the enrollment duration until the latest activity performed by the students. However, these variables reflect behavior over time, so they're not useful early on for identifying at-risk students. One straightforward workaround will be to remove these 3 input features when training the models and see how the rest

Figure 5.1: The ideal result after performing the clustering algorithm. [23]

of the input features contribute to the outcomes. Furthermore, we can record the student activities on the first day of registration, and use this data to train our logistic regression model. Furthermore, recall that in the answers.csv file, we have the self-pre category questions, which contain information on the initial knowledge level related to technology before starting the course. This piece of information can also be included in our model, as it might reveal pre-existing differences in technical knowledge, which could affect course completion likelihood. Including these variables may help capture baseline preparedness, providing valuable insight for early identification of at-risk students.

To address our second research question, only the Logistic Regression model was trained. The accuracy of our model is moderate. Therefore, it would be beneficial to introduce another model, such as a gradient boosting model, such as XGBoost. This type of model can also work for a small set of non-categorical data, and it has also been broadly used in the online learning domain [24].

Chapter 6

Conclusions

This thesis aimed to carry out a set of learning analysis (LA) for a specific online learning environment, RETOMadrID. This online environment offers digital literacy training, with a particular focus on women in vulnerable situations. To understand the direction of our LA, we formulated two main research questions: the likelihood of students answering the questions in the questionnaire and the possibility of course completion. To answer these research questions effectively, we introduced a specific branch of artificial intelligence into action, called machine learning (ML).

A review of the relevant literature highlights the importance and applicability of both supervised and unsupervised learning in the context of online education. Therefore, both types of ML approaches were considered.

To carry out this research appropriately, our methodology followed a specific workflow (see Figure 3.1): Data understanding, Data pre-processing, Feature Selection, Model Creation, Model evaluation. Throughout this workflow, we began by exploring and understanding the available data from three different CSV files (answers.csv, events.csv, and participation.csv), followed by preprocessing to ensure quality and consistency for model training (see Section 3.2). When the models were developed, we evaluated their performance. The insights generated through this evaluation process directly addressed the proposed research questions. The RETOMadrID platform is built using Elixir; therefore, the entire coding session is written in Elixir. We used Livebook to annotate our code, an interactive Elixir code notebook, due to its strong alignment with Elixir and its well-defined functionalities that facilitate the data analysis procedure.

The first ML algorithm that we have chosen is the k-mode clustering algorithm, a type of unsupervised learning specifically designed for categorical data. This algorithm is applied to the profile data of the students (such as age, Spanish level, etc), in order to gain an overview of our student population. Furthermore, due to a significant amount of missing profile data, we could not directly use these variables as input features in our future ML models. This is because the data is limited, and if we were to discard all of the missing data, we would have a small training and testing sample, thereby weakening the model's training and

evaluation capacity. To address this issue, we used the cluster labels generated by the clustering algorithm and then incorporated them as an additional input feature in our subsequent modeling steps. This approach preserved valuable insights from the profile data while avoiding data loss. Based on the clustering results, we identified three distinct groups of students with the help of a sub-clustering algorithm:

- Inactive students: Those who do not answer most of the questions in the questionnaire.
- Students who are over 50 years old, retired people, and native Spanish speakers. (see Section 4.1.1 for further details)
- Students who are under 50 years old, non-retired people, and non-Spanish people. (see Section 4.1.2 for further details)

Taking into account the three identified student groups, we prepared models for supervised learning. We developed both a decision tree model and a random forest model to address our first research question: whether a student is likely to respond to the questionnaire. The selection of these models is made based on the fact that the input features and the predicted value are categorical. Furthermore, these models do not require complex modeling to learn meaningful patterns from a low dimensionality of input data. Unlike the decision tree, our random forest is an ensemble learning method; therefore, it combines multiple decision tree models to produce more accurate and stable results. In Python, there is a well-built Sklearn module that can generate the importance of the input features for both models. In order to execute Python in Livebook directly, the Pythonx module is imported and used.

Throughout the implementation, we achieved a high prediction accuracy of 99% with both models. From the feature importance score, we also gathered some important insights. We realized that to improve the overall interaction level, we need to keep those inactive students engaged. Moreover, those questions that are easy to answer, such as true/false questions or multiple choice questions, are more popular compared to Likert scale questions, given that Likert scale questions are unfamiliar to people over 60 years old. Furthermore, both models showed that longer video time is positively associated with answering. This finding rules out the possibility that students are engaging with the material but failing to respond.

Based on these insights, we can conclude that to improve the questionnaire response rate, we need to focus on re-engaging those who are inactive. Furthermore, Likert scale questions can be removed, especially for elderly people.

After both the decision tree and the random forest were built, we built a logistic regression model, another supervised learning method, to address our second research question: whether a student is likely to complete the course. For this model, we again considered the three identified student groups. This selection was made because the majority of our features are non-categorical, but at the same time, our predicted values are binary. Logistic regression is a simple model that works well with a small dataset consisting of only 875 rows of data.

Throughout the implementation, we achieved a moderate prediction accuracy of 74%. The accuracy is moderate since we have a small amount of dataset, leading to a limited number of training samples and testing samples. If there were more time, it would be beneficial to try out other models, such as the XGBoost model. In theory, this type of model can also work for a small set of non-categorical data.

From the result of our analysis, we have identified 3 different factors affecting the likelihood of course completion: The total time following the course, the total number of activities triggered by the students, and the enrollment duration until the latest activity performed by the students. However, these factors are long-term factors. Therefore, this finding does not facilitate the early detection of at-risk students. Different workarounds were mentioned, such as the re-training of the model discarding these 3 input features, or using the data related to the student activities on their first day of registration. Due to the time limitation, these workarounds have not yet been carried out. We also found that inactive students have the worst impact on student success among the three different groups of students.

Overall, we have successfully explored both research questions with the help of ML and data analysis methodologies. Although the finding in our second research question does not help us generate an early detection of at-risk students, suggestions have been described for future work. Furthermore, despite the perfect result that we have obtained in the K-mode clustering algorithm, we encountered certain challenges during its execution. That is, we needed to execute the cluster algorithm multiple times to obtain good clustering results. One possible way to optimize this algorithm is to introduce novel algorithms [22]. Finally, this research also facilitated hands-on learning with Elixir in the Livebook tool and demonstrated the integration of Python into Elixir-based data workflows.

Bibliography

[1] J. Wood, *These 3 charts show the global growth in online learning*, Nov. 2022. [Online]. Available: https://www.weforum.org/stories/2022/01/online-learning-courses-reskill-skills-gap/#:~:text=People%20are%20increasingly%20accessing%20online%20courses%20to%20help,new%20learner%20growth%20online%20came%20from%20emerging%20economies. (visited on 05/05/2025).

[2] ETSIINF. "Retomadrid | reequilibrio territorial en madrid con inclusión digital." (), [Online]. Available: https://blogs.upm.es/retomadrid/ (visited on 05/07/2025).

[3] "The elixir programming language." (), [Online]. Available: https://elixir-lang.org/ (visited on 05/11/2025).

[4] "Home - livebook.dev." (), [Online]. Available: https://livebook.dev/ (visited on 05/11/2025).

[5] S. M. Dol and P. M. Jawandhiya, "Classification technique and its combination with clustering and association rule mining in educational data mining — a survey," *Engineering Applications of Artificial Intelligence*, vol. 122, p. 106 071, 2023, ISSN: 0952-1976. DOI: https://doi.org/10.1016/j.engappai.2023.106071. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0952197623002555.

[6] W.-B. Xie, Y.-L. Lee, C. Wang, D.-B. Chen, and T. Zhou, "Hierarchical clustering supported by reciprocal nearest neighbors," *Information Sciences*, vol. 527, pp. 279–292, 2020, ISSN: 0020-0255. DOI: https://doi.org/10.1016/j.ins.2020.04.016. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0020025520303157.

[7] G. J. Oyewole and G. A. Thopil, "Data clustering: Application and trends," *Artificial Intelligence Review*, vol. 56, no. 7, pp. 6439–6475, Jul. 2023, ISSN: 1573-7462. DOI: 10.1007/s10462-022-10325-y. [Online]. Available: https://doi.org/10.1007/s10462-022-10325-y.

[8] S. Preidys and L. Sakalauskas, "Analysis of students' study activities in virtual learning environments using data mining methods," *Baltic Journal on Sustainability*, vol. 16, pp. 94–108, Mar. 2010. DOI: 10.3846/tede.2010.06.

[9] A. Navarro and P. Moreno Ger, "Comparison of clustering algorithms for learning analytics with educational datasets," *International Journal of Interactive Multimedia and Artificial Intelligence*, vol. inPress, p. 1, Sep. 2018. DOI: 10.9781/ijimai.2018.02.003.

[10] N. Dewi and I. B. G. Dwidasmara, "Implementation of k-modes algorithm for clustering of stress causes in university students," *JELIKU (Jurnal Elektronik Ilmu Komputer Udayana)*, vol. 9, p. 419, Feb. 2021. DOI: 10.24843/JLK.2021.v09.i03.p17.

[11] S. Rizvi, B. Rienties, and S. A. Khoja, "The role of demographics in online learning; a decision tree based approach," *Computers Education*, vol. 137, pp. 32–47, 2019, ISSN: 0360-1315. DOI: https://doi.org/10.1016/j.compedu.2019.04.001. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0360131519300818.

[12] C. Wang, L. Chang, and T. Liu, "Predicting student performance in online learning using a highly efficient gradient boosting decision tree," in *Intelligent Information Processing XI*, Z. Shi, J.-D. Zucker, and B. An, Eds., Cham: Springer International Publishing, 2022, pp. 508–521, ISBN: 978-3-031-03948-5.

[13] S. Jia, "Logistic regression analysis of online course click rate," in *2021 International Conference on Machine Learning and Intelligent Systems Engineering (MLISE)*, 2021, pp. 388–391. DOI: 10.1109/MLISE54096.2021.00081.

[14] Y. Guo, "Logistic regression analysis of college students' learning behavior considering mooc data in online learning environment," *Journal of Combinatorial Mathematics and Combinatorial Computing*, vol. 127a, pp. 1177–1190, Apr. 2025. DOI: 10.61091/jcmcc127a-068.

[15] a. mutawa a.m, "Perspective chapter: Moocs at higher education: Current state and future trends," in Mar. 2023, ISBN: 978-1-83769-523-2. DOI: 10.5772/intechopen.1001367.

[16] "10 e-learning modeling technique and convolution neural networks in online education," in *IoT-enabled Convolutional Neural Networks: Techniques and Applications*. 2022, pp. 261–296.

[17] J. L. Lobo, J. Del Ser, A. Bifet, and N. Kasabov, "Spiking neural networks and online learning: An overview and perspectives," *Neural Networks*, vol. 121, pp. 88–100, 2020, ISSN: 0893-6080. DOI: https://doi.org/10.1016/j.neunet.2019.09.004. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0893608019302655.

[18] Aditya. "K-Modes Clustering Algorithm With Numerical Example." (), [Online]. Available: https://codinginfinite.com/k-modes-clustering-algorithm-with-numerical-example/ (visited on 06/08/2025).

[19] M. Y. Khan, A. Qayoom, M. Nizami, M. S. Siddiqui, S. Wasi, and K.-U.-R. R. Syed, "Automated prediction of good dictionary examples (gdex): A comprehensive experiment with distant supervision, machine learning, and word embedding-based deep learning techniques," *Complexity*, Sep. 2021. DOI: 10.1155/2021/2553199.

[20] Sourav. "Linear Regression vs Logistic Regression: Difference — analyticsvidhya.com." (), [Online]. Available: https://www.analyticsvidhya.com/blog/2020/12/beginners-take-how-logistic-regression-is-related-to-linear-regression/ (visited on 06/13/2025).

[21] D. Andrés. "Gradient Descent - ML Pills — mlpills.dev." (2022), [Online]. Available: https://mlpills.dev/machine-learning/gradient-descent/ (visited on 06/14/2025).

[22] Y. Sha, J. Du, Z. Yang, and F. Jiang, "Cluster center initialization for fuzzy k-modes clustering using outlier detection technique," in *Pattern Recognition and Computer Vision*, Z. Lin, M.-M. Cheng, R. He, *et al.*, Eds., Singapore: Springer Nature Singapore, 2025, pp. 3–18, ISBN: 978-981-97-8487-5.

[23] "K-means++ Algorithm - ML - GeeksforGeeks — geeksforgeeks.org." (), [Online]. Available: https://www.geeksforgeeks.org/machine-learning/ml-k-means-algorithm/ (visited on 06/19/2025).

[24] S. Hakkal and A. A. Lahcen, "Xgboost to enhance learner performance prediction," *Computers and Education: Artificial Intelligence*, vol. 7, p. 100 254, 2024, ISSN: 2666-920X. DOI: https://doi.org/10.1016/j.caeai.2024.100254. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2666920X24000572.