POLITÉCNICO DE LEIRIA
ESCOLA SUPERIOR DE TECNOLOGIA E GESTÃO

CHARACTERIZATION OF THE LIFESTYLE AND WELL-BEING OF STUDENTS FROM THE POLYTECHNIC OF LEIRIA

DANIEL AUGUSTO BERTOLDO SANTOS

Mestrado em Ciências de Dados

Leiria, March 2024

POLITÉCNICO DE LEIRIA

ESCOLA SUPERIOR DE TECNOLOGIA E GESTÃO

CHARACTERIZATION OF THE LIFESTYLE AND WELL-BEING OF STUDENTS FROM THE POLYTECHNIC OF LEIRIA

DANIEL AUGUSTO BERTOLDO SANTOS
Number: 2210769

Project report under the supervision of Professor Rui Filipe Vargas de Sousa Santos and Professor Susana Raquel Carvalho Ferreira from the School of Technology and Management at the Polytechnic of Leiria.

Leiria, March 2024

ORIGINALITY AND COPYRIGHT

ACKNOWLEDGMENTS

I want to express my gratitude to the Polytechnic Institute of Leiria for their indispensable collaboration in this project. Specifically, I am grateful to the SAPE (Student Support Service) team, particularly its coordinator, Professor Graça Seco, and its psychologists: Ana Patrícia Pereira, Luís Filipe, and Sandra Alves. Their input and assistance, including their valuable contributions to the choice of the survey and their help with the translation of SMILE into Portuguese, were instrumental in successfully completing this project.

I also express the most sincere gratitude to my thesis advisors, Professor Rui Santos and Professor Susana Ferreira, for their guidance, mentorship, and patience. This project would not be possible without their support throughout the research process.

Furthermore, I also extend my appreciation to Professor Vicent Balanzá-Martínez of the University of Valencia for the permission to use the SMILE survey in this research project and to the Institute's presidency, especially Professors Carlos Rabadão and Carolina Henriques, for their support in disseminating the survey to all students at the Polytechnic Institute of Leiria. Lastly, I express my heartfelt thanks to the 649 students who took the time to complete the survey. Their participation and responses were essential in enabling the realization of this project.

ABSTRACT

Entering higher education marks a significant juncture in a student's life. It often involves a shift towards independence, characterized by a distancing from family and friends, increased responsibilities, and greater autonomy in decision-making. These changes can influence well-being and various aspects of lifestyle, such as dietary habits, exercise routines, alcohol and drug use, and sexual behavior. Despite the initial excitement, this transition may also induce stress and anxiety. Academic demands, including grades, exams, and deadlines, as well as the newfound responsibilities of managing one's schedule, finances, and social relationships, all affect the well-being. Hence, several studies recently conducted on college students have highlighted the importance of monitoring their well-being, especially since several reports have indicated a significant increase in mental health issues among college students, such as depression and anxiety. In particular, the Short Multidimensional Inventory Lifestyle Evaluation (SMILE), developed in 2020, is a 43-item self-rated questionnaire consisting of 7 domains, allowing a multidimensional evaluation of a (healthy) lifestyle.

Within this context, a web survey was conducted among students at the Polytechnic of Leiria. This survey collected socio-demographic data, SMILE scores and clinical variables data, including screening for depression and anxiety. The key insights gleaned from the statistical analysis of the obtained data are summarized, particularly focusing on describing lifestyle and well-being, discerning differences between categories, and validating the survey instrument. Two supervised learning classification methodologies (logistic regression and decision trees) were applied to identify depression and anxiety issues based on responses to the survey. The reliability of these classifications were carry out using confusion matrix, accuracy, sensitivity, specificity, predictive values, and the area under the ROC curve in a test sample.

The results reveal that lower SMILE scores are associated with positive screening of depression/anxiety in higher education students, despite the reliability appears insufficient to confidently recommend its use for screening depression and/or anxiety disorders. However, it enables the characterization of students' lifestyles, the assessment of their well-being levels, and, consequently, the identification of potential mental health issues.

Keywords: anxiety, classification, depression, higher education students, well-being, survey validation.

RESUMO

A entrada no ensino superior é um marco importante na vida de um estudante. Envolve usualmente uma transição para a independência, caracterizada pelo afastamento da família e dos amigos, aumento de responsabilidades e maior autonomia na tomada de decisões. Estas mudanças podem influenciar o bem-estar e vários aspectos do estilo de vida, como hábitos alimentares, rotinas de exercício, consumo de álcool e drogas e comportamento sexual. Apesar da emoção inicial, esta transição pode também induzir stress e ansiedade. As exigências acadêmicas, incluindo notas, exames e prazos, assim como as novas responsabilidades de gerir o próprio horário, finanças e relacionamentos sociais, afetam o bem-estar. Assim, estudos recentemente realizados em estudantes universitários têm destacado a relevância de monitorizar seu bem-estar, especialmente porque diversos relatórios têm indicado um aumento significativo em problemas de saúde mental, como depressão e ansiedade. Em particular, o *Short Multidimensional Inventory Lifestyle Evaluation* (SMILE), desenvolvido em 2020, é um questionário autoavaliativo com 43 itens, composto por 7 domínios, permitindo uma avaliação multidimensional de um estilo de vida (saudável).

Neste contexto, foi aplicado um inquérito *online* aos estudantes do Politécnico de Leiria, recolhendo dados sociodemográficos, pontuações do SMILE e dados para triagem de depressão e ansiedade. As principais conclusões da análise estatística dos dados obtidos são resumidas, focando sobretudo a descrição do estilo de vida e do bem-estar, avaliando diferenças entre categorias e validando instrumentos de pesquisa. Duas metodologias de classificação supervisionada (regressão logística e árvores de decisão) foram aplicadas para identificar problemas de depressão e ansiedade com base nas respostas ao questionário. A fiabilidade foi avaliada numa amostra de teste utilizando matriz de confusão, acurácia, sensibilidade, especificidade, valores preditivos e a área sob a curva ROC.

Os resultados revelam que pontuações baixas do SMILE estão associadas a presença de depressão/ansiedade nos estudantes do ensino superior, apesar da fiabilidade parecer insuficiente para recomendar seu uso para triagem de transtornos de depressão/ansiedade. No entanto, ele permite a caracterização dos estilos de vida, a avaliação dos níveis de bem-estar e, consequentemente, a identificação de possíveis problemas de saúde mental.

Palavras-chave: ansiedade, classificação, depressão, estudantes do ensino superior, bem-estar, validação de inquérito.

INDEX

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS

Ac	Accuracy.
ANOVA	Analysis of variance.
AUC	Area Under the ROC Curve.
DN	Diet and Nutrition domain of the SMILE questionnaire.
EE	Environmental Exposure domain of the SMILE questionnaire.
EFA	Exploratory Factor Analysis.
ESAD.CR	School of Arts and Design.
ESECS	School of Education and Social Sciences.
ESSLei	School of Health Sciences.
ESTG	School of Technology and Management.
ESTM	School of Tourism and Maritime Technology.
F	Female.
GAD-7	General Anxiety Disorder-7.
GIC	Image and Communication Office.
IPLeiria	Polytechnic Institute of Leiria.
KMO	Kaiser-Meyer-Olkin measure.
M	Male.

n	Sample dimention.
NB	Non-binary.
NPV	Negative predictive value.
PA	Physical Activity domain of the SMILE questionnaire.
PHQ-2	Patient Health Questionnaire-2.
PPV	Positive predictive value (Precision).
ROC	Receiver Operating Characteristic.
RS	Restorative Sleep domain of the SMILE questionnaire.
SD	Standard deviation.
Se	Sensitivity (Recall).
SM	Stress Management domain of the SMILE questionnaire.
SMILE	Short Multidimensional Inventory Lifestyle Evaluation.
SMILE-C	Short Multidimensional Inventory Lifestyle Evaluation - Confinement.
Sp	Specificity.
SS	Social Support domain of the SMILE questionnaire.
SU	Substance Use domain of the SMILE questionnaire.
TEL	Test for equality of location.
TeSP	Professional Higher Tecnhical Courses.
THV	Test of homogeneity of variances.
TI	Test of independence.
TN	Test of normality.

1

INTRODUCTION

Admission into higher education is a milestone in any young person's life, especially if the student can enroll in the desired course. The feeling of achieving a goal, coupled with the curiosity and desire to experience university life, typically translates into a significant moment in their lives. However, as the initial excitement wears off, this transition can also generate some stress, anxiety, and apprehension. Academic pressure with grades, exams, expectations, and deadlines, as well as new responsibilities concerning managing one's schedule, finances, and social relationships, can also affect the mental health or well-being of college students. College is a period of transition and adjustment, where students may face new environments, increased independence, and a shift in support structures. Commonly observed lifestyle factors among college students include irregular sleep patterns, unhealthy eating habits, lack of physical activity, and substance abuse. Social media and digital culture also play a role in this trend. Although social media provides a platform for connection and support, the constant exposure to highlighted, curated, and idealized representations of others' lives can lead to social comparison, self-doubt, and feelings of inadequacy or the fear of missing out (FOMO).

Recent research has highlighted the challenges and stressors that university students face during this transitional period (Hernández-Torrano et al., 2020). These challenges can lead to feelings of depression, anxiety, and low self-esteem and can negatively impact academic performance. In some cases, these challenges can even lead to disengagement or dropping out of the academic environment. Many higher education institutions have recognized the need to support their student's well-being and mental health and the need to provide adequate mental health support services.

In 2004, World Health Organization (WHO) defined mental health as "a state of well-being in which every individual realizes his or her own potential, can cope with the normal stresses of life, can work productively and fruitfully, and is able to make a contribution to her or his community." The term "well-being" encompasses two different philosophical orientations: the hedonic approach (which focuses on pleasure, life satisfaction) and the eudaimonia approach (which focuses on the purpose of life) (Ryff and Keyes, 1995). However, the current idea is that well-being encompasses an individual's perception of themselves and their relationship with others, life goals, and level of happiness, among others. Understanding the

factors that contribute to the well-being and mental health of college students is crucial for developing effective interventions and support systems.

In this study, the primary objective is to translate the SMILE (Short Multidimensional Inventory Lifestyle Evaluation) questionnaire to the Portuguese population while gauging its internal consistency. This adaptation seeks to evaluate the well-being of students at the Polytechnic Institute of Leiria (IPLeiria). Additionally, the PHQ-2 and GAD-7 questionnaires were employed to screen for depression and anxiety. Logistic regression and decision trees were used to screen anxiety and depression disorders using data from the SMILE questionnaire.

The structure of this study unfolds as follows: Chapter 2 outlines the survey administered to IPLeiria students, elucidating the rationale behind selecting SMILE. Chapter 3 delves into the methodologies employed for survey analysis. The focal point of the study, Chapter 4, elucidates the primary findings gleaned from the survey data analysis. This encompasses exploratory data treatment, examination of variable associations and correlations, assessment of survey internal consistency, and application of supervised learning techniques to classify whether students have anxiety and/or depression problems, leveraging SMILE survey data and sociodemographic variables. Chapter 5 engages in a concise discussion, comparing obtained results with existing literature. Finally, Chapter 6 succinctly summarizes the key conclusions. The appendix includes both English and Portuguese versions of the survey and a script detailing primary data treatment operations using the R language.

2

RELATED WORK

In recent years, there has been an increasing amount of research conducted on the mental health of higher education students (Hernández-Torrano et al., 2020). The main focus of some of these researches has been identifying the factors, like well-being levels and lifestyle behaviors, that impact their mental health and academic performance. These studies are extremely important, as they provide insights to develop strategies and interventions to promote post-secondary education students' mental health and overall well-being.

During COVID-19 lockdowns, numerous research studies were conducted to monitor the mental health and well-being of university students. The pandemic has led to a relatively high number of higher education students experiencing symptoms of anxiety and depression. The prevalence of these symptoms varies significantly across different countries (Chang et al., 2021). After this pandemic period, it is important to question whether the findings and recommendations from the studies conducted during this time are still relevant in non-pandemic situations. Additionally, we need to examine whether this period has affected the ability of higher education students to cope with daily stressors or under pressure situations.

Several surveys were considered to assess the well-being of students at the IPLeiria and explore the relationship between their lifestyle behaviors and mental health. After careful consideration and consultation with professional psychologists at SAPE (Student Support Services), we considered the Fantastic Lifestyle Assessment questionnaire (Wilson et al., 1984), Student Well-being Process questionnaire (Student WPQ), Psychological Well-being Scales (PWBS), and Short Multidimensional Inventory Lifestyle Evaluation (SMILE) (Balanzá-Martínez et al., 2021).

Completing some surveys can become fastidious for students due to their excessive length. Therefore, it is crucial to select surveys that are concise and manageable for students while still capturing relevant information about the different study domains. One of the reasons for excluding the PWBS survey was that it consisted of 84 items. Although it had already been adapted and applied to Portuguese higher education students (887), including 123 students from IPLeiria (Lopes, 2015). Nonetheless, the provided results do not discriminate between institutions.

The Fantastic Lifestyle Assessment questionnaire is a widely used tool for assessing lifestyle behaviors. It comprises 9 domains: Family, Physical Activity, Nutrition, Tobacco/toxics, Alcohol, Sleep/stress, Personality type, Insight, and Career. This questionnaire was developed in 1984 by Wilson and Ciliska. Since then, it has been used in various studies to assess the lifestyle behaviors of different populations, including Portuguese higher education students (Silva et al., 2014). Since its development in 1984, there have been significant changes in the world in terms of context and lifestyle, and the COVID-19 pandemic has exacerbated these changes. Therefore, it is reasonable to question whether this questionnaire is still the best tool for assessing lifestyle behaviors in higher education students. Recently, a new tool called the Short Multidimensional Inventory Lifestyle Evaluation (SMILE) has been developed by Balanzá-Martinez and his research team to evaluate seven lifestyle domains: Diet/Nutrition, Substance Abuse, Physical Activity, Stress Management, Restorative Sleep, Social Support, and Environmental Exposure. The evaluation is carried out through a 43-item questionnaire, where the individual has to look back at their last 30 days and choose an option for each question using a 4-point Likert-type scale (ranging from "never" to "always"). The total score is calculated by adding the scores of all the questions from 1 to 4, some of which have inverse scores, in order for higher scores to correspond to better lifestyle behaviors. This tool was adapted to the pandemic context during the first confinement period, in 2020. The adapted tool, designated by SMILE-C, is a shortened version of the previous one, consisting of 27 items (Balanzá-Martínez et al., 2021; Cervera-Martínez et al., 2021).

After evaluating different options, the SMILE questionnaire assessed the student's lifestyle behavior at IPLeiria. Although the questionnaire was available in English, Spanish, and Brazilian Portuguese, it needed to be adapted for Portuguese (Portugal) due to significant language differences. The adaptation of the SMILE questionnaire was necessary to ensure its reliability. Upon reviewing our initial Portuguese (Portugal) translation, the SAPE team recommended changes that enhanced the clarity and effectiveness of some questions.

The survey begins with five sociodemographic questions: gender, age, program, school, and displacement. Nine more questions were added to the survey to screen for depression and anxiety, including 2 items from Patient Health Questionnaire-2 (PHQ-2) and 7 items from the Generalized Anxiety Disorder (GAD-7). These surveys have been previously validated in the Portuguese population through two studies: "Validação do Patient Health Questionnaire-9 (PHQ-9) para a população portuguesa" (Cordeiro, 2022) and "Factor structure and construct validity of the Generalized Anxiety Disorder 7-item (GAD-7) among Portuguese college students" (Bártolo et al., 2017). To reduce the number of questions, we choose to apply PHQ-2 instead of PHQ-9 to screen for depression; this is common in this field of research (Havrylyuk, 2020). Balanzá-Martínez's team also uses these types

of questionnaires in some of their works, and this could be useful to compare the results (Balanzá-Martínez et al., 2021).

We contact Balanzá-Martínez to inform them of our intention to use their questionnaire, explain the purpose of our work, and request permission to use the adapted versions of the SMILE in our survey.

The corresponding survey can be found in Appendix A.1 (English version) and in Appendix A.2 (Portuguese version).

3

METHOD

3.1 STUDY DESIGN

An online survey (web survey) was conducted between March 15, 2023, and May 15, 2023. The online questionnaire was programmed in Microsoft Forms and included questions about lifestyle behaviors and demographics. The survey is available online[1]. In Appendix A, the questions are presented in English (Section A.1) and in Portuguese (Section A.2), the language in which the survey was administered. The present analysis included data from all respondents who agreed to participate in the study after reading the informed consent form.

3.2 ETHICAL ASPECTS

The Ethics Committee at the IPLeiria approved the study (CE/IPLEIRIA/19/2023). The survey was conducted anonymously, and no personal information such as name, city, or IP address was collected. Before commencing the questionnaire, participants were provided with a consent form to signify their agreement. All the responses gathered were self-reported and deemed to be truthful.

3.3 POPULATION AND SAMPLE

This study's participants were adult students from IPLeiria who had internet access and agreed to participate after reading the informed consent form. The study included individuals of all genders.

The survey was disseminated by the Image and Communication Office (GIC) of the IPLeiria through mailing lists to all IPLeiria students. For this reason, instead of defining a sample a priori, 60 days of data collection was specified. Participants were asked to complete the survey only once in the email to avoid repeated responses.

1 The survey can be viewed at https://forms.office.com/e/wCsfrYXDMh.

Of the total population of about 13500 students in the institution, 649 (4.8%) surveys were answered and included in the final sample. However, 12 of those surveys were deemed ineligible for the study due to various reasons, namely not agreeing to participate in the study (1), not being a student of IPLeiria (6), being under 18 years old (1), or belonging to a higher education course with an insignificant number of responses such as a postgraduate course (1), preparatory course for people over 23 years old (1), 60+ Program (aimed at those over 60 years old) (1), and enrolled in a single curricular unit (1).

Hence, 637 (4.7%) surveys were evaluated. A detailed description of all results is provided in Chapter 4.

3.4 VARIABLES AND MEASUREMENTS

Sociodemographic information was also gathered in addition to the SMILE questionnaire, PHQ-2, and GAD-7 surveys. This data was used as variables or to create new variables for this study.

- Gender is a categorical variable with three options: male, female, and non-binary.
- Age is a quantitative variable.
- Age group is an ordinal variable and consists of seven levels, with smaller intervals at the lower end to accommodate the higher density of responses in those age intervals.
- Course is a categorical variable with three levels of the current level of Education the respondent is pursuing. As mentioned in Section 3.3, three levels were removed from the study due to insufficient responses (a single student).
- School is a categorical variable with five levels of the School from the IPLeiria the respondent is attending.
- Displaced is a binary variable with two levels indicating whether the respondent lives with their family or not.
- SMILE is a quantitative variable representing a respondent's score on the SMILE questionnaire. Higher values indicate a healthier lifestyle.
- SMILE_DN is a quantitative variable representing the score on the Diet and Nutrition domain of the SMILE questionnaire.
- SMILE_SU is a quantitative variable representing the score on the Substance Use domain of the SMILE questionnaire.

- SMILE_PA is a quantitative variable representing the score on the Physical Activity domain of the SMILE questionnaire.
- SMILE_SM is a quantitative variable representing the score on the Stress Management domain of the SMILE questionnaire.
- SMILE_RS is a quantitative variable representing the score on the Restorative Sleep domain of the SMILE questionnaire.
- SMILE_SS is a quantitative variable representing the score on the Social Support domain of the SMILE questionnaire.
- SMILE_EE is a quantitative variable representing the score on the Environmental Exposure domain of the SMILE questionnaire.
- SMILE_Classes is an ordinal variable with four levels based on the SMILE variable separated by the quantiles.
- SMILE-C is a quantitative variable of the respondent score on the reduced Confinement version of the SMILE questionnaire.
- PHQ is a quantitative variable of the respondent score on the PHQ-2 questionnaire.
- GAD is a quantitative variable of the respondent score on the GAD-7 questionnaire.
- Anxiety is a binary variable identifying if the student has an anxiety disorder. The respondent is considered anxious if the PHQ-2 score is equal to or greater than 3.
- Depression is a binary variable identifying if the student has depression disorder. The respondent is considered depressed if the GAD-7 score is equal to or greater than 10.
- Anxiety_Depression is a binary variable based on the two previous items that indicate whether the respondent is experiencing both anxiety and depression.
- Anxiety_Depression_comp is a categorical variable with four levels that indicate whether the respondent is experiencing anxiety, depression, or both, or neither.

3.5 STATISTICAL ANALYSIS

The statistical analysis was done on R language on RStudio 2023.09.1 Build 494 (R Core Team, 2023). A script with the main computations performed on R is provided in the appendix (see Section B). It was necessary to use several packages to achieve the results (listed at the beginning of the script), among which `dplyr` and `tidyverse` for data manipulation, `ggplot2` stands out in graphical terms, `stats` and `car` for statistical analysis (including the `glm` function to fit generalized linear models, which includes logistic regression),

`ggcorrplot` for correlation representation, `psych` for internal consistency, `rpart` for trees decision, `UBL` for oversampling, `caret` for confusion matrix and associated measures of classification performance, `proc` for the ROC curves, among others.

The following statistical tools are used in the analysis.

- Significance Level:
  - A significance level of 5% ($\alpha = 0.05$) was used in all statistical hypothesis tests.
- Normality Tests:
  - Shapiro-Wilk normality test was used to check if each variable is (approximately) characterized by a normal distribution
- Tests for Homogeneity of Variance:
  - Bartlett's test for homogeneity of variances is a statistical test used to determine if the variances of different samples are equal. Its purpose is to verify the assumption of equal variances before conducting an ANOVA test. However, it should be noted that Bartlett's test is sensitive to non-normality and may produce erroneous results if applied to a non-normal distribution dataset. Therefore, the Shapiro-Wilk test was used to check the normality assumption before using Bartlett's test. Whenever normality was rejected, Levene's test was employed.
  - The Levene test is a statistical test similar to Bartlett's. It is used to check if the variances are equal for all groups. It is more robust when data does not have a normal distribution, namely if it is based on the median. It was used instead of Bartlett's test when the null hypothesis of the Shapiro-Wilk test was rejected.
- Test for equality of location (under normality):
  - The Student's t-test compares the means of two groups (paired or independent), assuming that the data is normally distributed, i.e., to test whether the difference between the group's mean responses is statistically significant.
  - The ANOVA (Analysis of Variance) was used to compare the means of three or more groups through the comparison of the variance between groups to the variance within groups.
- Non-Parametric tests for equality of location (for data not assumed to be of normal distribution):
  - The Wilcoxon rank sum test is used to compare the location of two related samples and was used to compare groups defined by binary variables in order to assess whether there is a significant difference between them.

  - Kruskal-Wallis rank sum test is used to compare the location of three or more independent groups.

- Correlation Analysis:
  - Pearson correlation is used to assess linear relations between quantitative normal variables (strength and direction).
  - The Spearman rank correlation coefficient is used to measure a monotonic relationship between variables (strength and direction). It was applied throughout the study since, in many variables, the null hypothesis of the normality test was rejected.
- Test of independence:
  - Fisher's Test of Independence was used to assess the existence of an association between two variables.
- Reliability Analysis:
  - Cronbach's alpha is a statistical measure used to assess the internal consistency and reliability of a set of related measurements in an instrument. It compares the amount of shared variance or covariance among the items to the overall variance. This study used Cronbach's alpha to validate the survey translation by comparing its results between the SMILE and SMILE-C and to the original survey value. It also was employed to assess PHQ-2 and GAD-7 surveys.
  - Kaiser-Meyer-Olkin (KMO) was used to measure the sampling adequacy for exploratory factor analysis (EFA). It measures the proportion of variance among variables that might be common variance. The higher the proportion, the higher the KMO value, and the more suitable the data is for factor analysis.
- Imbalanced Class handling:
  - Random oversampling was used to replicate instances from the minority class in the training data to address the class imbalance since the use of balanced classes may be relevant for obtaining more appropriate classification methods (with greater reliability)
- Use of train (estimation) and test (assessment) samples
  - A random 80/20 dataset split was used to divide data into training (80%) and testing (20%) sets for model evaluation. This division was used on all classification models to predict anxiety and/or depression using the SMILE or the SMILE-C scores and the sociodemographic variables.
- Classification methodologies:

  - The logistic regression (LR) can be used to predict the odds and, therefore, the probability of a specific outcome in a binary classification based on one or more predictor variables. When computing these probabilities, a classification procedure can be defined (classified into the most likely category).
  - Decision Tree classifies data points based on a tree-like structure with branching decisions at each node based on a cutoff value. With this methodology, we can define a classification scheme that is easy to apply, even for those who know little or nothing about Mathematics.

- Evaluation Metrics (for the classification models):
  - Confusion matrix is the contingency table between real and predicted classification. As such, it illustrates the count of correct (true positives and true negatives) as well as incorrect (false positives and false negatives) classifications. These numbers facilitate the calculation of various reliability measures of the classification, including accuracy, sensitivity, specificity, positive predictive value, and negative predictive value.
  - Accuracy is the proportion of correctly classified instances.
  - Sensitivity (Recall) is the proportion of true positives correctly identified.
  - Specificity is the proportion of true negatives correctly identified.
  - Positive predictive value (precision) is the proportion of positive predictions that are actually true.
  - Negative predictive value is the proportion of negative predictions that are actually true.
  - ROC (Receiver Operating Characteristic) curve plots the trade-off between sensitivity and specificity for different classification thresholds; an ideal ROC curve would follow the upper left corner of the graph, indicating that the model can perfectly distinguish between positive and negative cases.
  - AUC (Area Under the ROC Curve) is a single metric summarizing the performance of a classification model across all possible thresholds. It represents the total area underneath the ROC curve; an AUC of 1 corresponds to a perfect classifier, while an AUC of 0.5 corresponds to a random classifier. Thus, a higher AUC indicates a better performance.

- Overfitting assessment
  - To detect overfitting, statistically significant disparities existed between the AUC values in the training and test samples using DeLong's test for two ROC curves

were examined. If differences were present, it indicated that the classification quality in the test sample was notably inferior to that in the training sample. This procedure was particularly crucial for assessing the depth of classification trees ().

In utilizing these tools, we rely on references (Breiman, 2017; Hastie et al., 2009; James et al., 2021; Santos et al., 2019; Tabachnick and Fidell, 2021; Zelterman, 2015), where comprehensive information regarding all the methodologies mentioned can be found.

4

RESULTS

In this chapter, the main results of the statistical analysis are presented. In terms of organization, we divided the results into the following topics: sample description, internal consistency and sampling adequacy, depression and anxiety screening, SMILE survey, SMILE-C survey, SMILE survey domains, and screening for depression and anxiety using the SMILE well-being score.

4.1 SAMPLE DESCRIPTION

Most participants were women, with an average age of 24, and most were Bachelor students from the ESTG – School of Technology and Management. Table 1 provides more information.

The study's total number of respondents is 637 students. The largest portion of the sample comes from ESTG (n=296), followed by ESECS (n=106) and ESSLei (n=98).

The distribution of students across program levels is 442 students in Graduation programs, 112 in master's programs, and 83 students enrolled in TeSP programs.

The majority of the students identify as female (412, 64.7%). Males comprise a smaller portion (215, 33.7%), while a limited number of students identify as non-binary (10, 1.6%).

Out of all the students surveyed, 350 of them (54.9%) were found to be displaced from family. The percentage of displaced students varies across different schools, with the highest percentage found in ESTM (78.9%) and the lowest in ESECS (41.5%). Additionally, the proportion of displaced students is the highest in graduation courses, with 250 out of 442 displaced students (56.6%). This proportion is lower in TeSP courses, with 45 out of 83 students (54.2%), and in masters, with 55 out of 112 students (49.1%).

The student sample encompasses a range of ages, with the largest category falling within the 20 to 21 age group (177). Older age groups are less populated, with only 100 students above 30.

When considering both gender and age together, the data suggests that females are most prevalent within the younger age groups from 18 to 24 (65.9% in comparison with males).

Table 1: Sample Description. **F**: Females; **M**: Males; **NB**: Non-binary.

School	Level	Total	Gender			Displaced
			F	M	NB	
ESAD.CR	TeSP	6	3	3	0	2 (66.7%)
	Graduation	46	32	11	3	35 (76.1%)
	Master	9	9	0	0	8 (88.9%)
	Total	61	44	14	3	45 (73.8%)
ESECS	TeSP	19	16	2	1	9 (47.4%)
	Graduation	56	49	7	0	26 (46.4%)
	Master	31	24	7	0	9 (29.0%)
	Total	106	89	16	1	44 (41.5%)
ESSLei	TeSP	10	8	2	0	6 (60.0%)
	Graduation	73	64	7	2	46 (63.0%)
	Master	15	12	3	0	5 (33.3%)
	Total	98	84	12	2	57 (58.2%)
ESTG	TeSP	41	14	27	0	22 (53.7%)
	Graduation	213	97	113	3	98 (46.0%)
	Master	42	24	18	0	24 (57.1%)
	Total	296	135	158	3	144 (48.6%)
ESTM	TeSP	7	4	3	0	6 (85.7%)
	Graduation	54	46	7	1	45 (83.3%)
	Master	15	10	5	0	9 (60.0%)
	Total	76	60	15	1	60 (78.9%)
IPLeiria	TeSP	83	45	37	1	45 (54.2%)
	Graduation	442	288	145	9	250 (56.6%)
	Master	112	79	33	0	55 (49.1%)
	Total	637	412	215	10	350 (54.9%)

Table 2: Gender and age groups.

Gender	Age groups							Total
	[18,20)	[20,22)	[22,25)	[25,30)	[30,40)	[40,50)	[50,60]	
Female	95	120	101	30	33	24	9	412
Male	58	53	44	27	14	12	7	215
Non-binary	2	4	2	1	1	0	0	10
Total	155	177	147	58	48	36	16	637

The only age groups with a somewhat balanced division of males and females are the 25 to 29 (51.72% females) and 50 to 59 (56.25% females) age groups. All non-binary students in the sample are under 40 years old (Table 2).

The left side of Figure 1 shows a pie chart that summarizes the distribution of gender within the sample population and shows the prevalence of female students in the sample. On the right side, the image displays the gender and age group distribution within the sample population in a histogram format.

The graph confirms a higher prevalence of females across all age groups. The largest proportion of students falls within the 20 to 21 age group for both females and males, though the number of females is nearly double that of males in this category. The distribution of females gradually tapers off across the older age groups, while the distribution of males shows a steeper decline after the 25 to 29 age group.

The sample data includes a small number of non-binary students. However, it is worth noting that all of them are below 40.

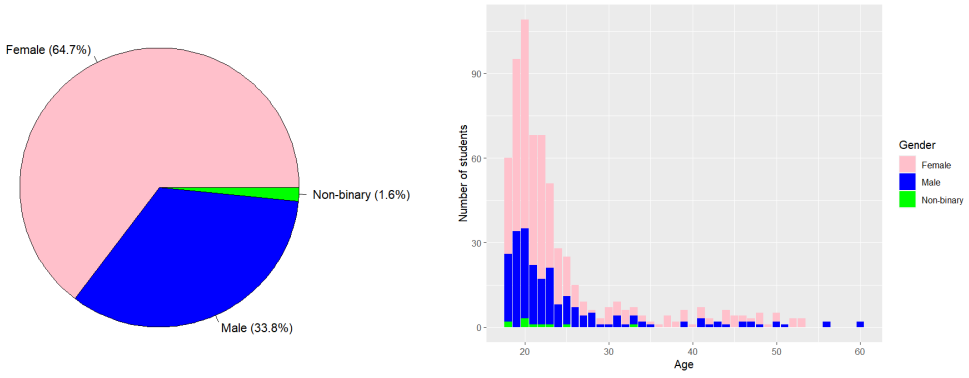

Figure 1: Gender and age distribution

The largest student population is from ESTG (296, 46.5%), followed by ESECS (106, 16.6%). The school with the fewest students is ESAD.CR (9.6%).

Graduation students make up the largest group (69.39%), followed by TeSP (18.98%) and Master (11.63%).

Of the total sample, ESECS has the highest proportion of Master students (29.0%) as well as TESP students, while ESAD.CR has the highest proportion of Graduation students (76.1%). However, ESTG has more TeSP students (41, 49.40%), cf. Figure 2.

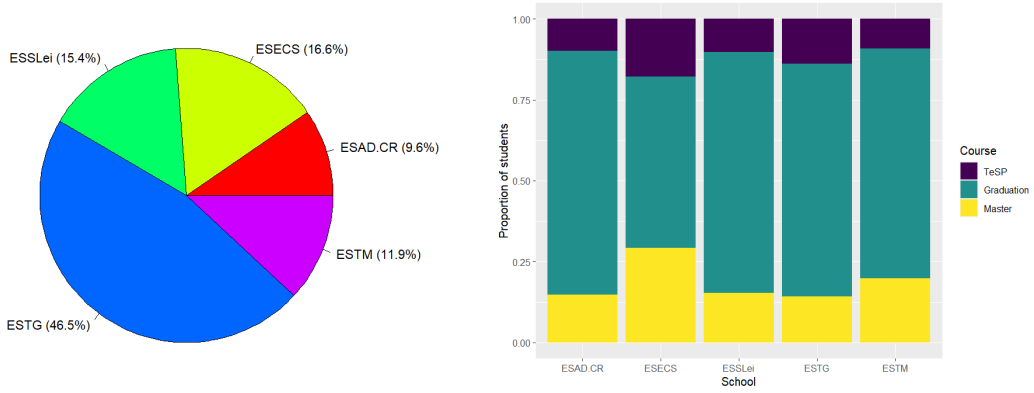

Figure 2: School and education level distribution

There is almost no difference in the gender distribution of displaced and non-displaced students (Figure 3).

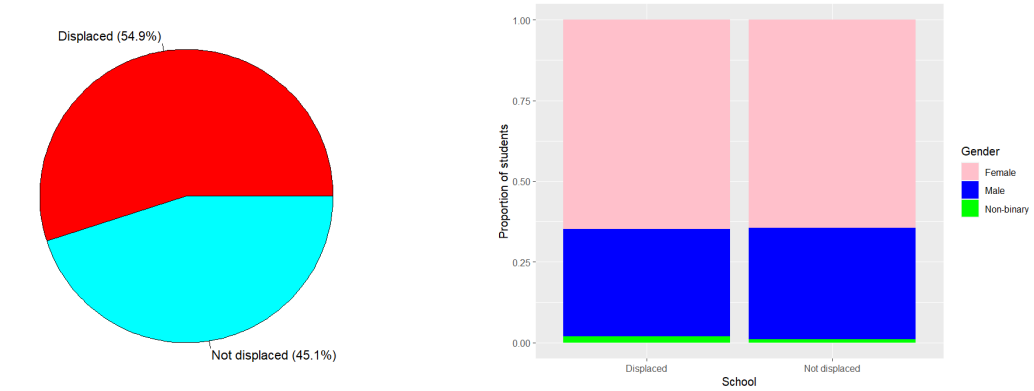

Figure 3: Displacement and gender distribution

While there is almost no difference in the gender distribution between displaced and non-displaced students (as shown in Figure 3), the data reveals a significant variation in displacement rates across schools and programs.

ESTM has the highest percentage of displaced students (78.9%), followed by ESAD.CR (73.8%) and ESSLei (58.2%). In contrast, ESECS has the lowest proportion of displaced students (41.5%) (Table 1).

Table 3: Survey internal consistency and reliability measures. α: Cronbach's alpha; **KMO**: Kaiser-Meyer-Olkin measure; **Quest.**: Questions

Survey	SMILE			SMILE-C		
	Quest.	α	**KMO**	**Quest.**	α	**KMO**
Diet and nutrition	Q01:Q07	0.61	0.68	Q03:Q07	0.51	0.58
Substance Use	Q08:Q11	0.64	0.67	Q08:Q11	0.64	0.67
Physical activity	Q12:Q15	0.63	0.63	Q12	–	–
Stress management	Q16:Q24	0.65	0.70	Q16:Q19 + Q23:Q24	0.55	0.58
Restorative sleep	Q25:Q29	0.54	0.67	Q25:Q26 + Q28:Q29	0.64	0.67
Social support	Q30:Q39	0.85	0.88	Q30:Q33 + Q36:Q37	0.76	0.82
Environment exposures	Q40:Q43	0.38	0.50	Q42	–	–
Entire survey	43 quests.	0.86	0.84	27 quest.	0.80	0.80
PHQ-2 & GAD-7	PHQ-2			GAD-7		
	2 quest.	0.83	–	7 quest.	0.92	0.92

Looking at program levels, TeSP courses show the highest displacement rate (85.7%), followed by Graduation programs (56.6%) and Master's programs (49.1%). This suggests that TeSP students are more likely to be displaced from their families compared to students in other programs.

Overall, there are valuable insights about the student population at the institution. A significant portion of the students are female, young adults, and displaced from their families. The data also highlights variations in these characteristics across schools and program levels. This information can be useful for the institution to develop targeted support services for different student groups.

4.2 INTERNAL CONSISTENCY AND SAMPLING ADEQUACY

The internal consistency and sampling adequacy of the SMILE, SMILE-C, PHQ-2, and GAD-7 surveys were assessed (Table 3). Internal consistency was evaluated using Cronbach's alpha coefficient (α), while the Kaiser-Meyer-Olkin (KMO) measure assessed sampling adequacy for factor analysis.

When considering the whole SMILE (α = 0.86, KMO = 0.84) and SMILE-C (α = 0.80, KMO = 0.80) surveys, both have excellent internal consistency and sampling adequacy. However, the analysis also revealed mixed results for the SMILE and SMILE-C domain scores. Several domains showed acceptable internal consistency (α > 0.60) and sampling adequacy (KMO > 0.60), indicating that the questions within those domains effectively measured a single underlying concept. These domains included Diet and Nutrition (α = 0.61, KMO = 0.68), Substance Use (α = 0.64, KMO = 0.67), Physical Activity (SMILE: α = 0.63, KMO = 0.63), Stress Management (SMILE: α = 0.65, KMO = 0.70; SMILE-C: α = 0.55, KMO = 0.58), and Restorative Sleep (SMILE-C: α = 0.64, KMO = 0.67). Notably, the Social Support domain demonstrated good internal consistency (α > 0.80) and sampling adequacy (KMO > 0.80) in both SMILE (α = 0.85, KMO = 0.88) and SMILE-C (α = 0.76, KMO = 0.82).

However, some domains in the SMILE and SMILE-C surveys had concerning results. The Environment Exposures domain (SMILE: α = 0.38, KMO = 0.50) showed poor internal consistency and inadequate sampling. This suggests that the questions within this domain may not be measuring a single concept effectively, and the sample size might be insufficient to capture the underlying factors. Additionally, the Physical Activity domain in SMILE-C was excluded because there was only one question.

The PHQ-2 and GAD-7 surveys demonstrated excellent internal consistency (α > 0.80) for depression and anxiety screening, respectively.

Overall, the findings suggest that the SMILE, SMILE-C, PHQ-2, and GAD-7 surveys provide generally good data for assessing student lifestyles. However, some domains within the SMILE and SMILE-C surveys require further refinement, particularly the Environment Exposures domain.

4.3 DEPRESSION AND ANXIETY SCREENING

The Patient Health Questionnaire-2 (PHQ-2) (Kroenke et al., 2003) was used to screen for current depression with a cut-off score of $\geq$ 3, while the Generalized Anxiety Disorder 7-item (GAD-7) (Spitzer et al., 2006) was used to screen for current anxiety with a cut-off score of $\geq$ 10. Then, a composite variable was created to identify participants who screened positive for both depression and anxiety disorders.

4.3.1 *Depression*

As mentioned, the Patient Health Questionnaire-2 (PHQ-2) was applied to screen for current depression. The left-hand side of Figure 4 shows the distribution of PHQ-2 scores among the students surveyed. The most frequent score was 2 (some depression symptoms), which corresponds to 34.2% of the students. The right-hand side of the figure depicts the distribution of depression screening results. According to the PHQ-2 criteria, 39.7% of the students screened positive for depression.

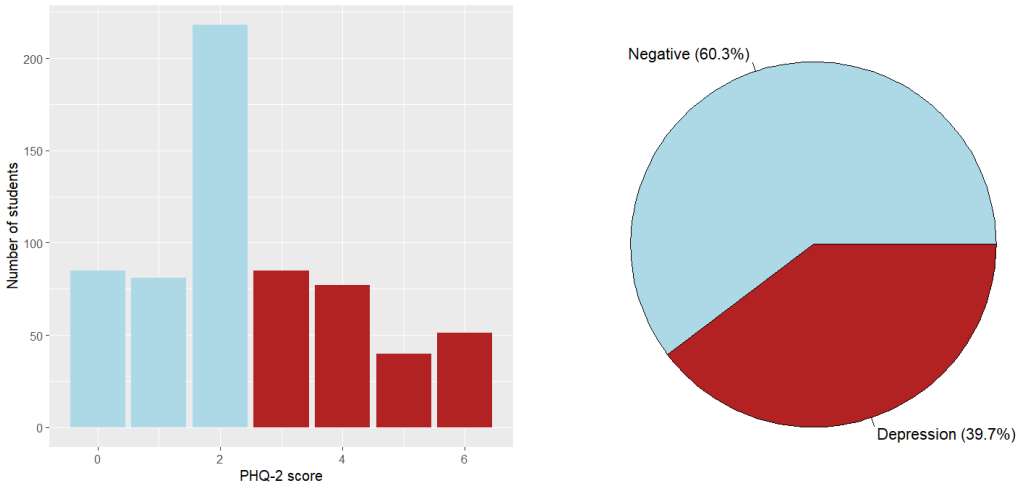

Figure 4: PHQ score and depression distribution

The left side of Figure 5 depicts the internal correlation between the two PHQ-2 questions at 0.7. This indicates a substantial positive correlation, meaning that scores on the two questions tend to move together. The right side of Figure 5 shows a Likert bar plot for the distribution of responses to each question. The most frequent response for both questions is 2 (several days). However, for question Q44 ("Little interest or pleasure in doing things"), a slightly higher percentage of students endorsed this response compared to question Q45 ("Feeling down, depressed, or hopeless"). Notably, question Q45 also has a higher proportion of students selecting response 4 (nearly every day) compared to question Q44.

These findings suggest that while students experience similar levels of symptom frequency across both questions, feelings of depression may be slightly more intense than a lack of interest or pleasure in activities.

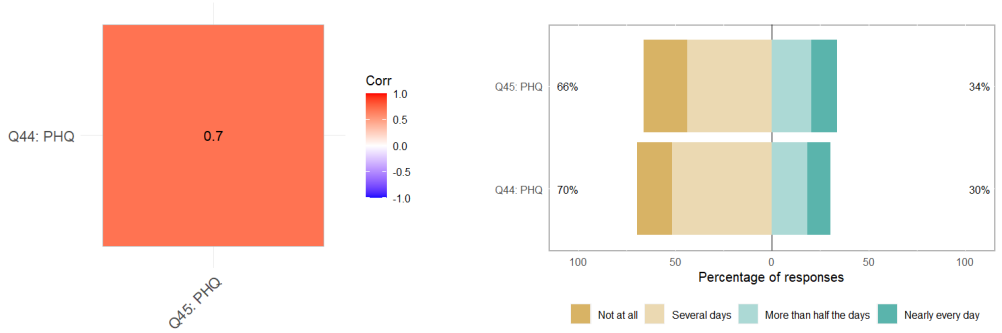

Figure 5: PHQ questions: correlation and answers

4.3.2 *Anxiety*

The General Anxiety Disorder-7 (GAD-7) was used to screen for current anxiety symptoms in the student population. The left-hand side of Figure 6 shows the distribution of GAD-7 scores among those surveyed. The most frequent score falls within the range of 5-9, indicating mild anxiety symptoms. The right-hand side of Figure 6 depicts the distribution of anxiety screening results. Hence, 44.9% of the students screened positive for anxiety based on the GAD-7 criteria.

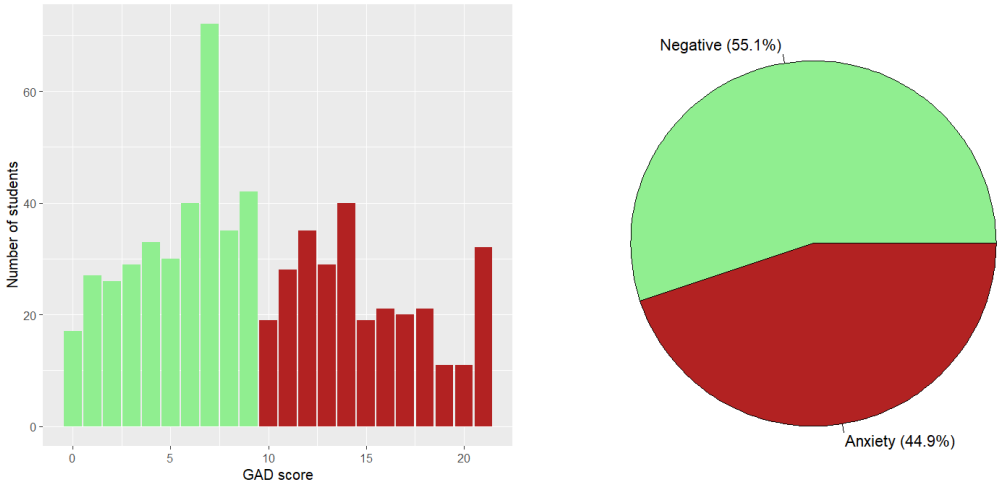

Figure 6: GAD score and anxiety distribution

Figure 7 explores the internal consistency of the GAD-7 instrument and the distribution of responses to each question. The left side of the figure shows the correlation coefficients between each of the GAD-7 questions. The correlation coefficients range from 0.58 to 0.72, indicating moderate to substantial positive correlations, which suggests good internal consistency for the GAD-7 in this sample. The right side of Figure 7 presents a Likert bar plot for the distribution of responses to each GAD-7 question. The most frequent response

for all questions was 2 (several days). Notably, question Q51 ("Becoming easily annoyed or irritable") had a slightly higher percentage of students endorsing this response compared to the other questions, while questions Q46 ("Feeling nervous, anxious or on edge") and Q48 ("Worrying too much about different things") has a higher proportion of students selecting response 4 (nearly every day).

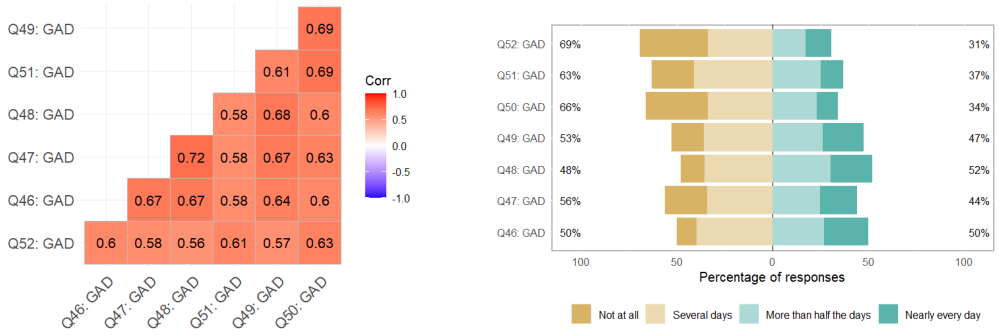

Figure 7: GAD questions: correlation and answers

This might suggest that while students experience general anxiety symptoms for several days throughout the week, feeling nervous or worrying excessively might be even more prevalent concerns.

4.3.3 *Anxiety and depression*

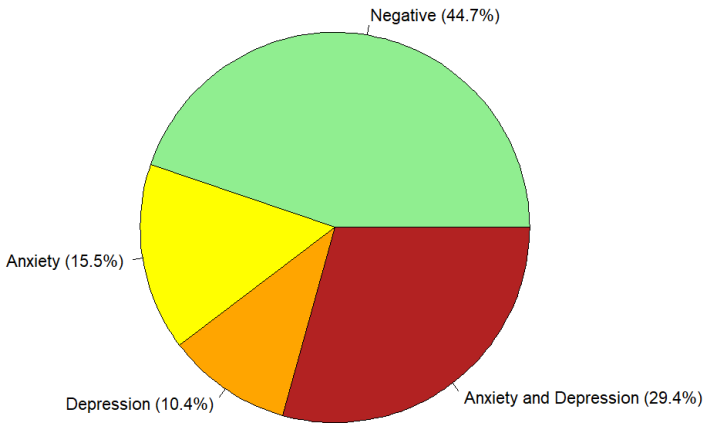

Figure 8: Anxiety and depression distribution

Figure 8 is a pie chart that shows the distribution of anxiety and depression scores among the study participants. The chart reveals that the largest proportion of students, 44.7%, of the students screened negative for both anxiety and depression. While 10.4% of the

students screened positive for depression only, and 15.5% screened positive for anxiety only. The remaining (29.4%) fell into the category of both anxiety and depression, indicating a significant comorbidity between the two conditions.

4.3.4 *Anxiety and depression per socio-demographic features*

Figure 9 explores the co-occurrence of anxiety and depression symptoms across different student subgroups. The analysis reveals several key findings:

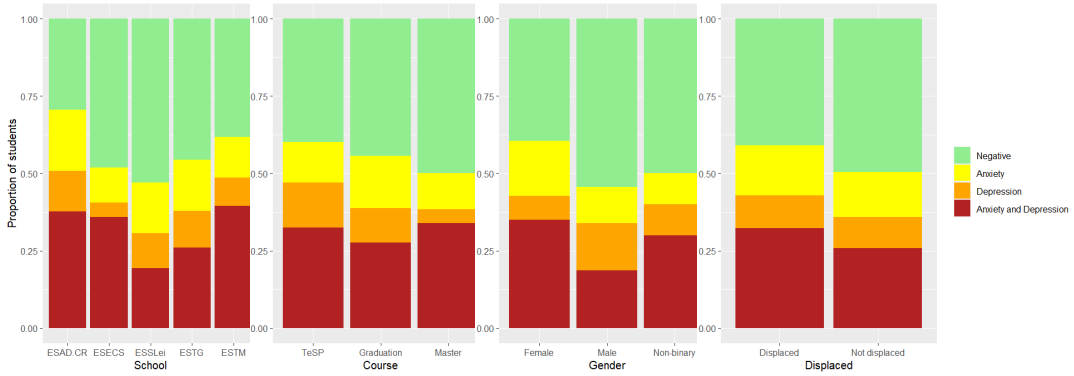

Figure 9: Anxiety and depression per socio-demographic features

- School: Students in ESAD.CR reported the highest overall prevalence of anxiety and depression, while ESTM students showed the greatest comorbidity. ESSLei students had the lowest overall rates.
- Course Level: TeSP students exhibited the highest overall prevalence of anxiety and depression. Master's students had the highest comorbidity rates.
- Gender: Females reported higher overall prevalence and comorbidity of anxiety and depression compared to males. Non-binary students fell between the two groups.
- Displacement Status: Displaced students reported a higher prevalence of anxiety and depression and a greater likelihood of comorbidity compared to non-displaced students.

The analysis of the occurrence of depression and anxiety symptoms across student subgroups is presented in Table 4 using the PHQ-2 scores and in Table 5 using the GAD-7 scores. The tables show the results of the statistical tests done to assess the normality of data distribution (TN), equality of location across groups (TEL), homogeneity of variances (THV), and independence between the socio-demographic feature and PHQ-2 or GAD-7 scores (TI). Moreover, it summarizes the PHQ-2 and GAD-7 scores for each category within

Table 4: PHQ-2 score per socio-demographic features. **SD**: Standard deviation; **TN**: test of normality; **TEL**: test for equality of location; **THV**: test of homogeneity of variances; **TI**: test of independence.

					p-value		
Feature	**Categories**	**Mean**	**SD**	**TN**	**TEL**	**THV**	**TI**
School	ESAD.CR	2.934	1.548	0.0006	0.0220	0.0665	0.0445
	ESECS	2.406	1.871	$< 10^{-4}$			
	ESSLei	2.122	1.431	$< 10^{-4}$			
	ESTG	2.483	1.733	$< 10^{-4}$			
	ESTM	2.750	1.674	0.0008			
Course	TeSP	2.602	1.689	0.0003	0.7025	0.3712	0.4458
	Graduation	2.466	1.670	$< 10^{-4}$			
	Master	2.500	1.831	$< 10^{-4}$			
Gender	Female	2.580	1.658	$< 10^{-4}$	0.0940	0.2553	0.0680
	Male	2.307	1.732	$< 10^{-4}$			
	Non-binary	2.700	2.452	0.0418			
Displaced	Displaced	2.603	1.670	$< 10^{-4}$	0.0489	0.9869	0.0825
	Not Disp.	2.352	1.727	$< 10^{-4}$			
All sample		2.490	1.700	$< 10^{-4}$	—	—	—

the socio-demographic features. The table also reports the mean score, standard deviation (SD), and the p-values for the above-mentioned statistical tests.

When considering depression in the PHQ-2 scores, the analysis is summarized in Table 4. Therefore, we can draw the following conclusions:

- School: The results indicate a statistically significant difference in PHQ-2 scores between schools (TEL p-value = 0.0220). However, Levene's test for homogeneity of variances (THV p-value = 0.0665) suggests equal variances across schools. The Shapiro-Wilk test (TN p-values) reveals non-normal data distribution for all schools. Fisher TI reveals an association between depression and schools (TI p-value = 0.0445).
- Course Level: The p-value for the test of equality of location (TEL p-value = 0.7025) seems to reveal that the locations of the populations being compared are equal, suggesting no significant difference in PHQ-2 scores between course levels (TeSP, Graduation, Master). Additionally, the tests for homogeneity of variances (THV p-value = 0.3712) and independence (TI p-value = 0.4458) do not show evidence for rejecting the null hypothesis.
- Gender: While the mean PHQ-2 score is higher for females compared to males, the test of equality of location (TEL) does not yield a significant p-value (p-value = 0.0940). The tests for homogeneity of variances (THV p-value = 0.2553) and independence (TI p-value = 0.0680) do not provide evidence against the null hypothesis.
- Displacement Status: Displaced students reported a higher mean PHQ-2 score than non-displaced students. The test of equality of location (TEL) suggests a statistically significant difference (p-value = 0.0489). However, the Shapiro-Wilk test (TN p-value = 0.0489) indicates non-normal data distribution.

When considering anxiety, the GAD-7 scores are examined in Table 5. Similar to the PHQ-2 results, the same statistics were analyzed.

- School: The results indicate a statistically significant difference in GAD-7 scores between schools (TEL p-value = 0.0081). The test for homogeneity of variances (THV p-value = 0.1120) suggests no strong evidence of unequal variances. However, the Shapiro-Wilk test (TN p-values) reveals non-normal data distribution for most schools, only ESAD.CR (TN p-value = 0.0920) and ESTM (TN p-value = 0.2215) seem to have normal data distribution. The TI reveals no association between variables (TI p-value = 0.3538).
- Course Level: The test of equality of location (TEL) does not yield a statistically significant difference in GAD-7 scores between course levels (TeSP, Graduation, Master) (p-value = 0.7537). The tests for homogeneity of variances (THV p-value =

Table 5: GAD-7 score per socio-demographic features. **SD**: Standard deviation; **TN**: test of normality; **TEL**: test for equality of location; **THV**: test of homogeneity of variances; **TI**: test of independence.

					p-value		
Feature	**Categories**	**Mean**	**SD**	**TN**	**TEL**	**THV**	**TI**
School	ESAD.CR	11.311	5.274	0.0920	0.0081	0.1120	0.3538
	ESECS	10.179	6.196	0.0010			
	ESSLei	9.469	5.251	0.0001			
	ESTG	8.945	5.803	$< 10^{-4}$			
	ESTM	10.526	5.257	0.2215			
Course	TeSP	9.980	5.996	0.0032	0.7537	0.4766	0.7806
	Graduation	9.509	5.640	$< 10^{-4}$			
	Master	9.946	5.840	0.0003			
Gender	Female	10.694	5.481	$< 10^{-4}$	$< 10^{-4}$	0.4035	0.0005
	Male	7.631	5.593	$< 10^{-4}$			
	Non-binary	9.800	6.844	0.2053			
Displaced	Displaced	10.109	5.608	$< 10^{-4}$	0.0231	0.6627	0.3608
	Not Disp.	9.084	5.807	$< 10^{-4}$			
All sample		9.647	5.717	$< 10^{-4}$	—	—	—

0.4766) and independence (TI p-value = 0.7806) also do not provide evidence against the null hypothesis.

- Gender: Females reported a significantly higher mean GAD-7 score compared to males. Hence, the TEL reveals a significant difference in the location (TEL p-value $< 10^{-4}$). The test for homogeneity of variances (THV p-value = 0.4035) does not provide evidence against the null hypothesis, but the TI test (p-value = 0.0005) indicates a statistically significant association between gender and GAD-7 scores. Only Non-binaries have a normal distribution (TN p-value = 0.2053).
- Displacement Status: Displaced students reported a slightly higher mean GAD-7 score compared to non-displaced students. The test of equality of location (TEL) suggests a statistically significant difference (p-value = 0.0231). However, there is a lack of normality in GAD-7 scores (as indicated by the TN p-values). Moreover, no significant variation difference (THV p-value = 0.6627) or association between variables was revealed (TI p-value = 0.3608).

Overall, these findings highlight potential variations in depression and anxiety symptoms across different student subgroups. Notably, the data exhibits a lack of normality in most variables, especially when considering depression. The only normally distributed data were observed in the GAD-7 scores of students from ESAD.CR (TN p-value = 0.0920) and ESTM (TN p-value = 0.2215), and for non-binary students (TN p-value = 0.2053).

The test for equality of location revealed no significant difference between course levels for both PHQ-2 (TEL p-value = 0.7025) and GAD-7 (TEL p-value = 0.7537). However, gender showed a significant difference only for GAD-7 (TEL p-value < 0.0001). In all other comparisons, the null hypothesis of equal means was rejected.

The test of homogeneity of variances did not suggest strong evidence of unequal variances in both PHQ-2 and GAD-7 scores.

The test of independence only indicated a possible association between school and PHQ-2 scores (TI p-value = 0.0445) and between gender and GAD-7 scores (TI p-value = 0.0005).

4.4 SMILE SURVEY

This section summarizes the scores obtained on the Short Multidimensional Inventory Lifestyle Evaluation (SMILE) survey.

Considering the complete sample, the SMILE score appears to be characterized by a normal distribution (TN p-value = 0.0702), presenting an approximately symmetrical distribution around its median (117.56), cf. Figure 10 and Table 6.

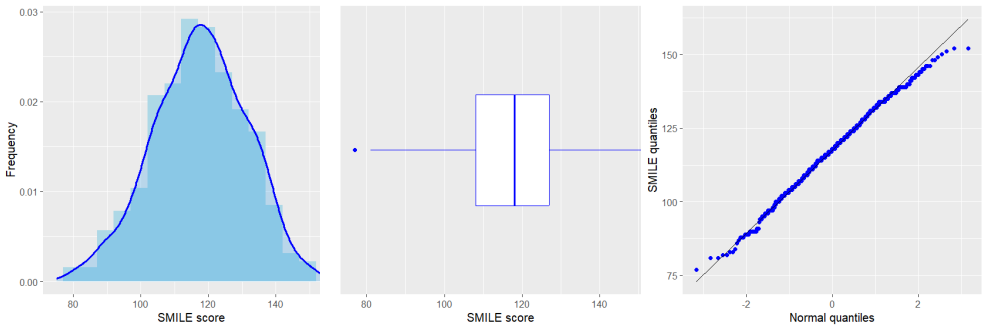

Figure 10: SMILE scores

The sample's mean and standard deviation for the SMILE scale is 117.6 ±13.80. Table 6 outlines the sociodemographic and clinical features of the sample, their corresponding SMILE averages, and their correlation. Male students (116.75 ±14.03), enrolment in the ESAD.CR School (111.48 ±12.23), pursuing a graduate education (116.90 ±13.68), and displacement from family (116.31 ±13.81) had lower SMILE scores on average, indicating

Table 6: SMILE score per socio-demographic features. **SD**: Standard deviation; **TN**: test of normality; **TEL**: test for equality of location; **THV**: test of homogeneity of variances; **TI**: test of independence.

					p-values		
Feature	**Categories**	**Mean**	**SD**	**TN**	**TEL**	**THV**	**TI**
School	ESAD.CR	111.56	12.14	0.5393	$< 10^{-4}$	0.0564	0.3553
	ESECS	121.01	14.57	0.0755			
	ESSLei	117.91	12.45	0.0133			
	ESTG	116.48	13.58	0.5062			
	ESTM	121.34	14.34	0.6815			
Course	TeSP	119.54	13.34	0.1539	0.1072	0.7608	0.5377
	Graduation	116.80	13.69	0.1217			
	Master	119.11	14.33	0.7787			
Gender	Female	117.97	13.38	0.09323	0.4530	0.0194	0.0665
	Male	116.60	14.01	0.6008			
	Non-binary	121.50	23.26	0.4373			
Displaced	Displaced	116.18	13.81	0.1720	0.0052	0.7857	0.0665
	Not Disp.	119.25	13.60	0.0623			
All sample		117.56	13.78	0.0702	—	—	—

an unhealthy lifestyle. Furthermore, having a positive depression (110.48 ±13.64), anxiety (111.89 ±13.65), or both depression and anxiety (108.99 ±13.90) screening was the strongest predictor of a lower SMILE score (see Figure 11, and Table 6).

After analyzing the scholarly level data, the population distribution was normal for all groups based on Shapiro-Wilk tests. TeSP had a p-value of 0.1539, Graduation had a p-value of 0.1217, and Master had a p-value of 0.7787. Bartlett's test was used to determine the homogeneity of variances. The p-value of 0.7608 was not significant. Conducting a one-way analysis of variance (ANOVA) and the non-parametric Kruskal-Wallis test to determine if there were significant differences in SMILE scores among the three academic courses, neither of these tests yielded statistically significant results, with p-values of 0.1072 for ANOVA and 0.1401 for the Kruskal-Wallis test. Therefore, these findings suggest no significant differences in lifestyle habits, as measured by SMILE scores, among students enrolled in TeSP, Graduation, or Master's programs when considering the entire dataset.

After analyzing data from the five schools, namely ESAD.CR, ESECS, ESSLei, ESTG, and ESTM, it was found that all schools, except ESSLei, had normally distributed SMILE scores. ESSLei showed a deviation from normality based on a p-value of 0.01332, and Levene's test suggested potential (but not significant) heterogeneity of variances among schools with a p-value of 0.05636. As a result, a non-parametric approach was used to determine significant differences in SMILE scores. Therefore, the Kruskal-Wallis test was chosen over a one-way analysis of variance (ANOVA). The Kruskal-Wallis test revealed a highly significant p-value of 0.00003, indicating notable variations in lifestyle habits among students from different schools, as measured by SMILE scores.

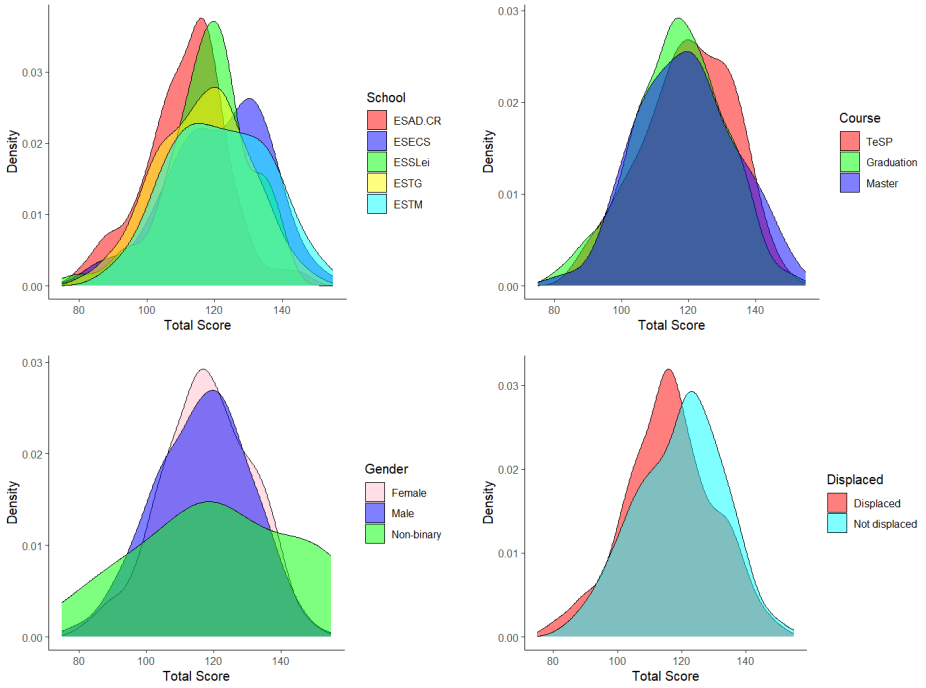

Figure 11: SMILE scores distribution by socio-demographic features

Considering the Shapiro-Wilk test, the SMILE scores were normally distributed in both displacement groups. The p-value for the "Displaced" group was 0.172, and for the "Not displaced" group, it was 0.06232, slightly above the common significance level of 0.05. Although normality was not rejected in either group, the deviation observed in the "Not displaced" group requires some caution. On the equality of variances test, a non-significant p-value of 0.7857, indicating that the assumption of equal variances was met. Parametric and non-parametric tests were performed to assess significant differences in SMILE scores between the two groups. The parametric t-test, assuming equal variances, showed a significant difference in SMILE scores between the "Displaced" and "Not displaced" groups with a p-value of 0.005169. The non-parametric Wilcoxon test produced a highly significant p-value of 0.003136, confirming significant differences in lifestyle habits between students who have

experienced family displacement and those who have not. As such, the analysis highlights notable differences in lifestyle habits, as measured by SMILE scores, between students who have experienced family displacement and those who have not. Hence, parametric and non-parametric tests consistently indicated significant differences, emphasizing the importance of considering family displacement status.

When examining the differences between age groups and analyzing the normality and equality of variances within each group, the results of the Shapiro-Wilk test reveal that normality is rejected in the "[18,20)" age group, with a p-value of 0.01518. However, normality is not rejected in the other groups, although some have p-values slightly above the common significance level of 0.05. Levene's test for equality of variances produces a non-significant p-value of 0.1381, indicating that the assumption of equal variances is met. To assume equal variances to investigate whether there are significant differences in SMILE scores among the age groups, a non-parametric Kruskal-Wallis test was used, which yielded a significant p-value of 0.02201. This result suggests that there are indeed significant differences in lifestyle habits, as measured by SMILE scores, among the different age groups.

After analyzing the anxiety and depression measurements obtained through GAD-7 and PHQ-2 screenings, it was found that the SMILE scores were normally distributed in all groups, with p-values ranging from 0.1921 to 0.9286. However, Bartlett's test showed a significant p-value of 0.02184, indicating that the assumption of equal variances among the screening result groups was unmet. This violated the equal variances assumption while conducting parametric and non-parametric tests to determine significant differences in SMILE scores among groups. The Kruskal-Wallis test showed an extremely significant p-value ($< 2.2 \times 10^{-16}$), indicating significant differences in lifestyle habits among the various anxiety and depression screening result groups.

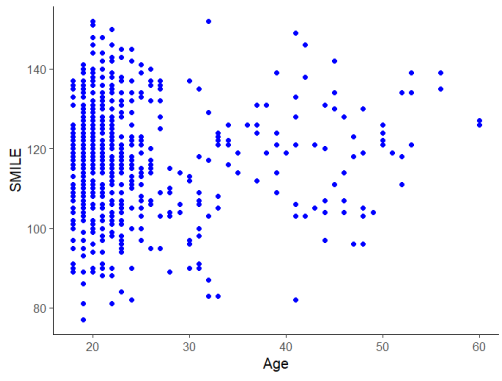

Figure 12: Scatter diagram of SMILE scores and age

The correlation Spearman coefficient between SMILE score and age is -0.0391 and, therefore, there is no evidence of a monotonic relation between these features (p-value = 0.3245), cf. Figure 12 illustrates.

4.4.1 *SMILE-C survey*

As discussed in Section 2, the Short Multidimensional Inventory Lifestyle Evaluation-Confinement (SMILE-C) survey was designed with the lockdowns of the COVID-19 pandemic in mind, and the questions considered non-relevant to the situation removed.

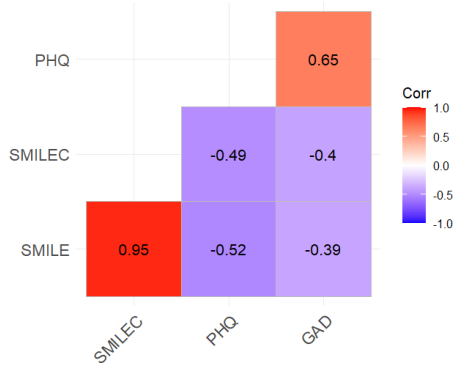

Figure 13: Correlation between surveys scores

Figure 13 shows a Spearman correlation matrix summarizing the associations between the questionnaires used in the study. Strong positive correlations (values close to 1) are observed between the SMILE and SMILE-C (0.95), indicating that these two instruments provide similar results. The PHQ-2 has moderate negative correlations with both SMILE (-0.52) and SMILE-C (-0.49), suggesting that higher scores on the PHQ-2 are associated with lower scores on the SMILE and SMILE-C. The GAD-7 (anxiety) scores show weaker negative correlations with SMILE (-0.39) and SMILE-C (-0.40) compared to the PHQ-2. Finally, PHQ-2 and GAD-7 present a significant positive relationship (0.65).

4.5 SMILE SURVEY DOMAINS

The SMILE and SMILE-C questionnaires have seven domains, each assessing lifestyle habits that the original study considered important pillars of well-being (see 2).

- Diet and Nutrition: This domain looks at how often the student has eaten home-cooked meals, checks food labels, consumes processed foods, manages stress eating, includes healthy options, and maintains a regular eating schedule.

- Substance Use: It inquires about the frequency of binge drinking, smoking tobacco products, using marijuana, and employing other illicit drugs.
- Physical Activity: The questionnaire asks about meeting recommended exercise guidelines, participating in team sports, taking the stairs or walking for errands, and feeling good after physical activity.
- Stress Management: It covers incorporating relaxation techniques, using psychological or physical strategies to cope with stress, having faith or religion, achieving work-life balance, feeling overwhelmed by tasks, commuting satisfaction, finding meaning in life, and practicing gratitude.
- Restorative Sleep: This domain concerns nightly sleep duration, feeling rested with your sleep amount, napping habits, maintaining a consistent sleep schedule, and reliance on sleeping pills.
- Social Support: It explores the interaction with friends and family, a sense of belonging, having a confidant, receiving help with chores, having someone for leisure activities, participating in social gatherings, enjoying free time, offering support to loved ones, and feeling loved.
- Environment Exposures (screen time/outdoor time): This domain investigates screen time, computer games, and internet use exceeding 2 hours daily, using electronic devices close to bedtime, spending time outdoors, and valuing connection with nature.

4.5.1 *Diet and nutrition domain*

In general, as shown in Figure 14, the questions in the diet and nutrition domain have a weak positive Spearman correlation with each other. The highest correlation is between checking labels for ingredients (Q2) and healthy food intake (Q5) (0.39), indicating that those who frequently check labels are also more likely to report consuming healthy foods. The lowest correlation is between processed food consumption (Q3) and sharing meals with family/friends (Q7), with a null correlation.

Table 3 (presented on page 19) shows that the domain has moderate internal consistency and reliability measures (α = 0.61, KMO = 0.68) on the SMILE. However, when transposed to the SMILE-C, those values still show moderate consistency and reliability measures (α = 0.51, KMO = 0.58) but at a lower value.

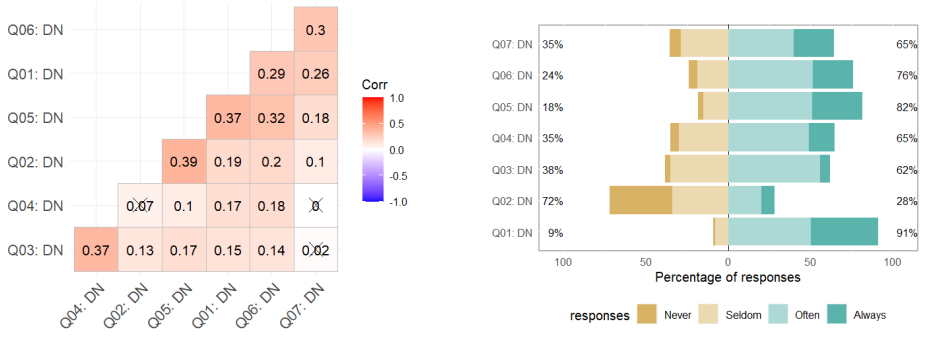

Figure 14: Diet and nutrition domain

4.5.2 *Substance use domain*

As shown in Figure 15, the questions in the substance use domain generally have weak positive Spearman correlations with each other. However, there is a moderate positive correlation (0.48) between smoking tobacco (Q9) and using marijuana or hashish (Q10).

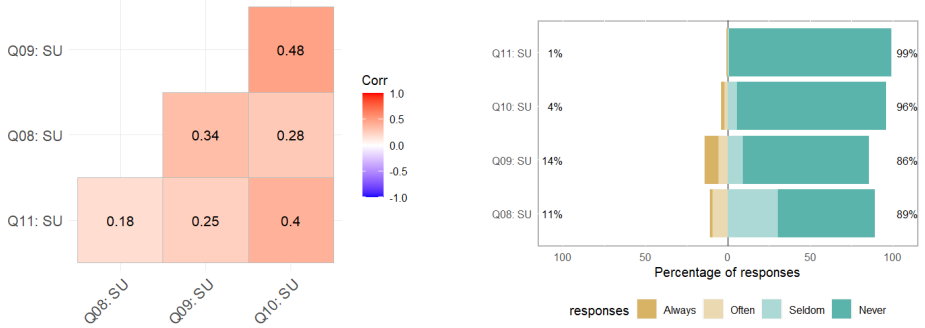

Figure 15: Substance use domain

Table 3 shows that the domain has moderate internal consistency and reliability measures(α = 0.64, KMO = 0.67) on the SMILE and SMILE-C since this is the only domain where no questions were removed.

4.5.3 *Physical activity domain*

As shown in Figure 16, the questions in the physical activity domain generally have weak positive Spearman correlations with each other. However, there is a moderate positive correlation (0.58) between exercising daily (Q12) and good feeling after performing physical activity (Q15). There is almost no correlation (0.03) between team sports (Q13) and using stairs/walking routines (Q14).

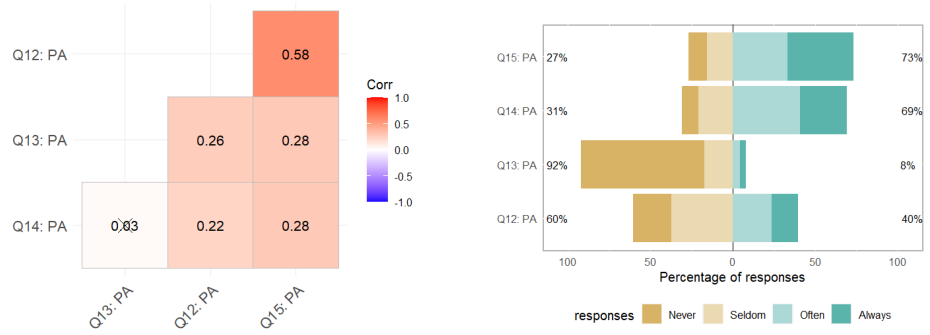

Figure 16: Physical activity domain

Table 3 shows that the domain has moderate internal consistency and reliability measures (α = 0.63, KMO = 0.63) on the SMILE. However, those measures could not be taken for the SMILE-C since there is only one question left in the domain.

4.5.4 *Stress management domain*

As shown in Figure 17, the questions in the stress management domain generally have weak positive Spearman correlations with each other. However, a moderate positive correlation is evident between the sentiment of leading a meaningful life (Q23) and the queries relating to gratitude for one's life (Q24), with a value of 0.66, as well as a 0.45 correlation between maintaining a healthy work-life balance (Q20). The questions regarding the utilization of meditation, mindfulness, or psychotherapy (Q17) and practicing faith or religion (Q19) demonstrate a relatively lower positive Spearman correlation and, at times, no correlation with the remaining questions.

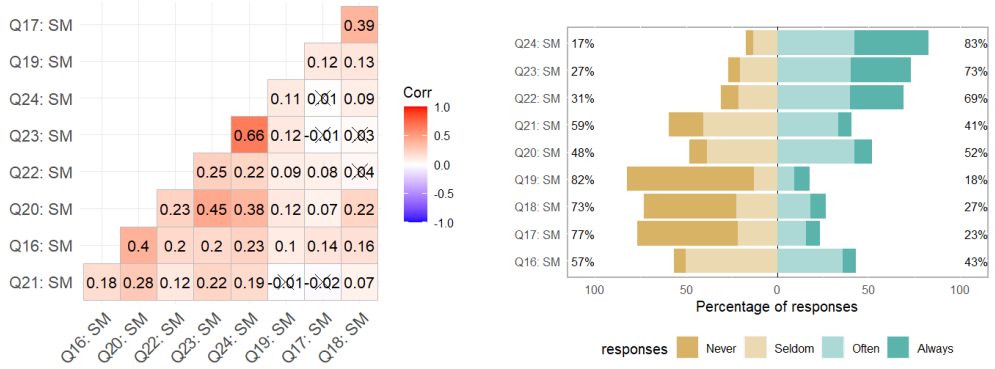

Figure 17: Stress management domain

Table 3 shows that the domain has moderate internal consistency and reliability measures(α = 0.65, KMO = 0.70) on the SMILE. However, when transposed to the SMILE-C, those

values still show moderate consistency and reliability measures (α = 0.55, KMO = 0.58) but at a lower value.

4.5.5 *Restorative sleep domain*

In the restorative sleep domain, Figure 18 indicates that there is a moderate positive correlation between having 7 to 9 hours of sleep (Q25) and the queries relating to a regular sleep schedule (Q28), with a value of 0.55, as well as a 0.48 correlation between feeling rested after sleep (Q26). On the other hand, the use of sleeping pills (Q29) shows a relatively lower positive Spearman correlation, while taking a siesta (Q27) exhibits no significant correlation with the remaining questions or even a weak negative Spearman correlation with the use of sleeping pills (Q29).

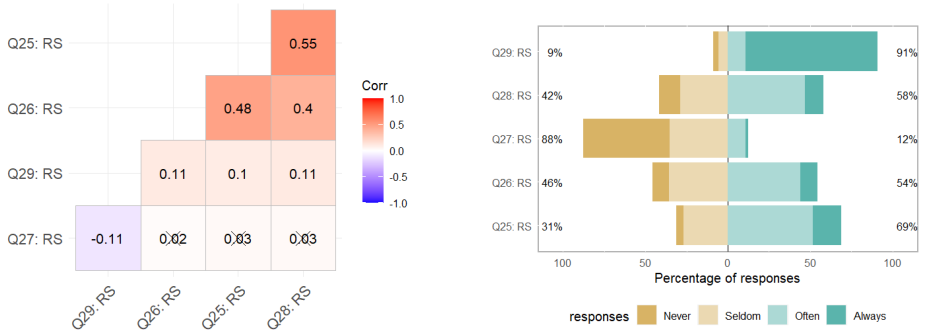

Figure 18: Restorative sleep domain

Table 3 shows that the domain has moderate internal consistency and reliability measures(α = 0.54, KMO = 0.67) on the SMILE. However, when transposed to the SMILE-C, those values still show moderate consistency and reliability measures (α = 0.64, KMO = 0.67) but at a higher value. This improvement is probably due to the removal of question 27 (rest after lunch).

4.5.6 *Social support domain*

As shown in Figure 19, the questions in the social support domain generally have weak to moderate positive Spearman correlations with each other. The most significant positive correlation (0.59) is evident between having someone in your life (Q34) and taking part in activities with significant others (Q35), as well as the correlation (0.58) between satisfaction with sexual life (Q38) and feeling of being loved (Q39). The questions regarding having

someone to help with everyday chores (Q33) and making yourself available to support your significant ones (Q37) demonstrate the lowest positive Spearman correlation (0.17).

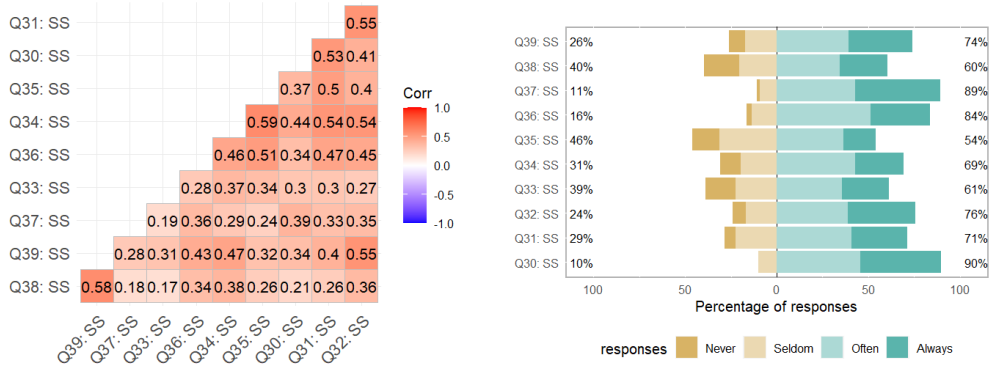

Figure 19: Social support domain

Table 3 shows that the domain has high internal consistency and reliability measures(α = 0.85, KMO = 0.88) on the SMILE. However, when transposed to the SMILE-C, those values still show high consistency and reliability measures (α = 0.76, KMO = 0.82) but at a lower value.

4.5.7 *Environment exposures domain*

As shown in Figure 19, the questions in the environment exposures domain generally have a non-significant Spearman correlation with each other. The exception is a weak positive correlation (0.40) between screen time (Q40) and screen time before sleep (Q41), as well as the correlation (0.39) between exposure to nature (Q42) and feeling relationship to nature (Q43). The remaining correlations are not significant.

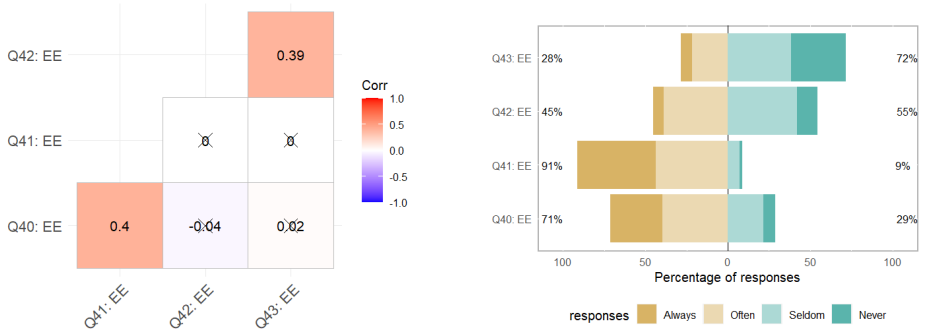

Figure 20: Environment exposures domain

Table 3 shows that the domain has weak internal consistency and moderate reliability measures(α = 0.38, KMO = 0.50) on the SMILE. Those measures could not be taken for the

SMILE-C since only one question is left in the domain. This problem is probably the reason why these questions were removed from the survey when SMILE-C was created.

4.5.8 *Correlation between domains*

The overall Spearman correlation between domains, as shown in Figure 21, is generally a weak Spearman correlation with each other. The exception is the stress management domain, which has a moderate positive correlation with all domains other than the environment exposures domain, which has a weak correlation (0.33). The substance use domain has a non-significant Spearman correlation with all other domains, but the diet and nutrition domain shows a weak correlation (0.15).

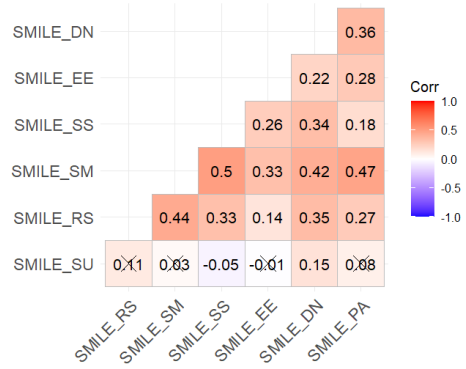

Figure 21: Correlation between domains scores

4.6 SCREENING FOR DEPRESSION AND ANXIETY USING THE SMILE WELL-BEING SCORE

To predict anxiety and depression, statistical models were created using the SMILE score and data from various variables. Two specific models, Logistic Regression LR) and Decision Tree (DT), were chosen. To ensure reproducibility, a seed was established, and the data was split into training and testing sets with a ratio of 80% to 20%. This approach was used in all the models. The models were ranked in the table based on their fiability (accuracy, sensitivity, specificity, positive predicted value, negative predicted value, F_1 score, and area under the ROC curve).

The various models are identified by an acronym that includes the model type and the predicted variable. The first letter corresponds to the target screening, “A” for anxiety, “D” for depression, and “AD” for both; after that, there is the acronym for the prediction model

type, “DT” for decision trees, and “LR” for logistical regression. The third corresponds to a specific variable, such as “S” for the SMILE scale, “SC” for the SMILE-C scale, and “SD” for the SMILE Scale divided into its domains. The included variables may vary in each model to optimize performance in order to adhere to the principle of parsimony[1]. Finally, the “O” is used to indicate the use of oversampling in the training sample.

4.6.1 *Anxiety*

This section delves into anxiety screening by utilizing SMILE(-C) scores, their domains, and other socio-demographic variables gathered during the study. The prediction models of logistic regression and decision trees were employed to achieve this. In cases where only one variable was utilized, the decision tree model was preferred.

4.6.1.1 *Screening for anxiety through a single variable*

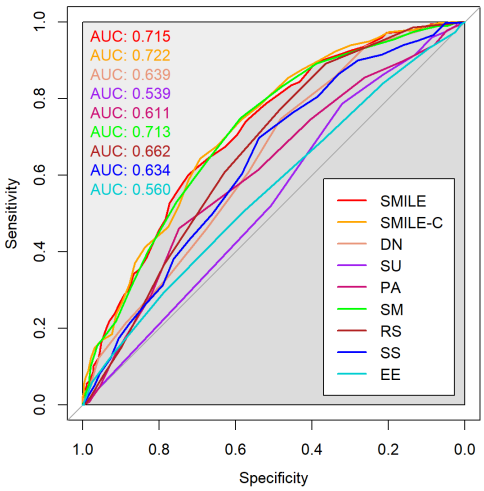

Figure 22: ROC curves of screening for anxiety (entire sample)

Considering the ROC curve and the area under the curve (AUC) as shown in Figure 22, the best model to predict the screening of anxiety is the SMILE-C score (AUC: 0.722), followed by the SMILE score (AUC: 0.715). Considering only a domain, the best is stress management (AUC: 0.713), and the worst is the environment exposures domain (AUC: 0.560), which is almost random (see Table 7).

1 The principle of parsimony suggests that when multiple theories fit the data equally well, scientists should opt for the simplest theory. Hence, it’s preferable to favor simple explanations over complex ones, resulting in models with minimal parameters.

Table 7: One feature to screen for anxiety. **Cut**: cut-off point (an individual is classified with anxiety if the feature is lower than the cut-off point); **Ac**: accuracy; **Se**: sensitivity; **Sp**: specificity; **PPV**: positive predicted value; **NPV**: negative predicted value; F_1: F_1 score; **AUC**: area under the ROC curve.

Feature	Cut-off	Ac	Se	Sp	PPV	NPV	F_1	AUC
SMILE	108	0.690	0.429	0.900	0.774	0.663	0.552	0.664
SMILE-C	73	0.683	0.464	0.857	0.722	0.667	0.565	0.661
DN	17	0.595	0.214	0.900	0.632	0.589	0.320	0.557
SU	15	0.635	0.393	0.829	0.647	0.630	0.489	0.611
PA	11	0.611	0.750	0.500	0.545	0.714	0.632	0.625
SM	19	0.635	0.339	0.871	0.679	0.622	0.452	0.605
RS	12	0.675	0.393	0.900	0.759	0.649	0.518	0.646
SS	26	0.635	0.304	0.900	0.708	0.618	0.425	0.602
EE	—	0.556	0	1	—	0.556	—	0.500

The decision tree model provides excellent visualization of the cut-off point for the prediction, as seen in Figure 23 where when the SMILE score is equal or greater than 108, there is a 76% chance of a negative screening for anxiety. Otherwise, there is a 24% chance of being a student with anxiety issues.

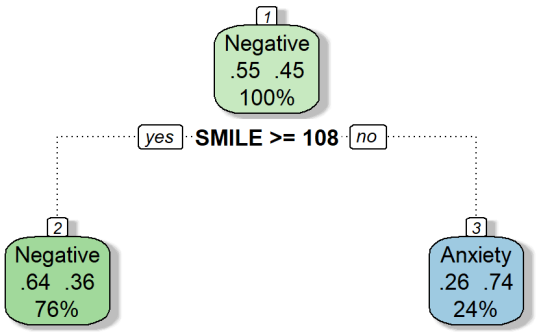

Figure 23: Decision tree for screening for anxiety through SMILE score

The results for the SMILE, SMILE-C scores, and its domains are summarized in Table 7.

- Accuracy (Ac): All models except EE have an accuracy between 0.595 and 0.690, indicating a moderate ability to classify individuals with or without anxiety correctly.
- Sensitivity (Se): Ranges from 0.214 (DN) to 0.750 (PA), indicating some models miss a significant portion of individuals with anxiety (low sensitivity).

- Specificity (Sp): Ranges from 0.500 (PA) to 0.900 (DN, SM, RS, SS), indicating some models might misclassify healthy individuals as anxious (low specificity).
- Positive Predictive Value (PPV): Ranges from 0.545 (PA) to 0.774 (SMILE), indicating a moderate probability that a positive test result reflects true anxiety.
- Negative Predictive Value (NPV): Ranges from 0.556 (EE, all negative) to 0.714 (PA), indicating a moderate probability that a negative test result reflects a truly healthy individual.
- F_1 Score: Ranges from 0.320 (DN) to 0.632 (PA), summarizing precision and recall into a single metric.
- AUC: Ranges from 0.500 (EE, random guess) to 0.664 (SMILE).

Overall, the SMILE score and PA domain show the strongest performance, with accuracy (SMILE: 0.690, PA:0.611), F_1 score (SMILE:0.552, PA:0.632), and AUC (SMILE:0.664, PA:0.625). However, all models have limitations, and more variables are needed for a better result. For example, if it is crucial to avoid missing anxious individuals (high sensitivity), and for that reason, when using only one variable, the PA domain model might be preferred despite its lower accuracy. The worst model was the EE domain model, classified as negative for anxiety screening in all the population with an AUC of 0.5, indicating it does not provide any useful information for anxiety prediction.

4.6.1.2 *Decision tree for screening for anxiety through multiple variables*

In this section, the screening analyses are not restricted to one explanatory feature. Hence, the following models use divergent variables to screen for anxiety. The first model (A_DT_S) uses decision trees for screening for anxiety through SMILE and socio-demographic features.

Without restrictions, the decision tree can be complex (see Figure 24) and have a high depth or height (number of tree branches from the root to the leaves, i.e., questions needed to perform the intended classification). Therefore, we may encounter overfitting problems, which can be evaluated by the difference in the ROC curve obtained in the training sample and the test sample. If the difference is significant, then the model is not generalizing to data that was not used in the model estimation, indicating overfitting issues. Therefore, we should reduce the complexity of the trees by imposing restrictions on the tree's depth. The tree shown in Figure 24 has a depth of 10, as to perform the intended classification, we may need to ask ten questions. Figure 25 represents the ROC curve obtained in the training sample and the test sample, with their respective areas under the ROC curve being 0.832 and 0.705. Applying the DeLong test to compare the area under two ROC curves, we obtain a p-value of 0.0118, indicating a significant difference between the two curves. Thus,

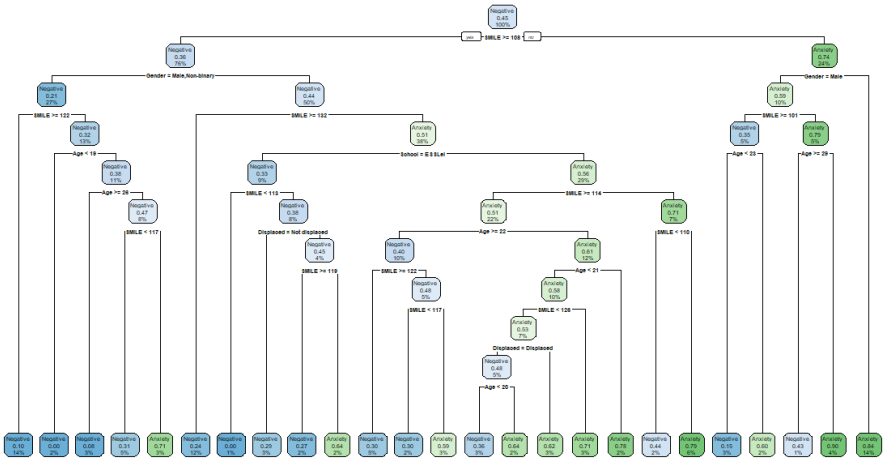

Figure 24: Decision tree for screening for anxiety through SMILE and socio-demographic features (with overfitting)

we restrict the tree's depth, starting by setting the maximum limit to 9 and analyzing for overfitting problems. The process is iterative until no significant differences are detected between train and test samples.

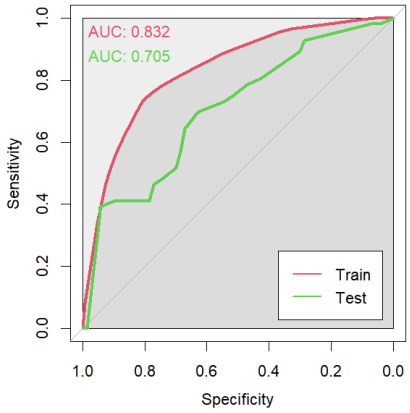

Figure 25: ROC curve for train versus test samples with overfitting

The difference between the training and test ROC curves ceased to be significant only when the maximum depth was set to 4 (with a p-value of 0.1721), resulting in the tree and ROC curves shown in Figure 26. It is worth noting that, even in this case, there is some difference in reliability between the training and test samples, but the DeLong test no longer deems it significant. The remaining reliability measures of this classification are presented in Table 8.

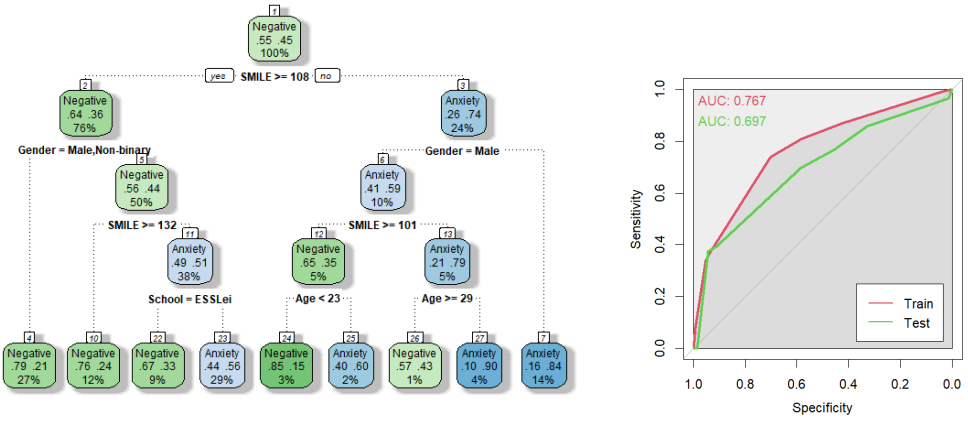

Figure 26: Decision tree and ROC curve for screening for anxiety through SMILE and socio-demographic features (controlled for overfitting)

Table 8: Multivariate analysis to screen for anxiety. **NV**: number of variables used in the screening procedure; **Ac**: accuracy; **Se**: sensitivity; **Sp**: specificity; **PPV**: positive predicted value; **NPV**: negative predicted value; F_1: F_1 score; **AUC**: area under the ROC curve.

Model	NV	Ac	Se	Sp	PPV	NPV	F_1	AUC
A_DT_S	4	0.635	0.696	0.586	0.574	0.707	0.629	0.697
A_DT_SC	2	0.690	0.446	0.886	0.758	0.667	0.562	0.668
A_DT_SD	7	0.698	0.571	0.800	0.696	0.700	0.627	0.713
A_DT_S_O	5	0.627	0.768	0.514	0.558	0.735	0.647	0.727
A_DT_SC_O	3	0.619	0.786	0.486	0.550	0.739	0.647	0.717
A_DT_SD_O	3	0.667	0.500	0.800	0.667	0.667	0.571	0.644
A_LR_S	3	0.706	0.661	0.743	0.673	0.732	0.667	0.770
A_LR_SC	3	0.683	0.643	0.714	0.643	0.714	0.643	0.762
A_LR_SD	5	0.730	0.679	0.771	0.704	0.750	0.691	0.762
A_LR_S_O	3	0.667	0.679	0.657	0.613	0.719	0.644	0.770
A_LR_SC_O	4	0.683	0.714	0.657	0.625	0.742	0.667	0.768
A_LR_SD_O	5	0.698	0.732	0.671	0.641	0.758	0.683	0.765

The performance of these multi-variable models (Table 8) when compared to the single-variable models (Table 7), the decision tree model A_DT_S_O (5 variables) achieved an AUC of 0.727, which is better than the best single-variable model (SMILE, AUC: 0.664). This also reveals that with the application of oversampling techniques, the model achieves better results in the decision tree models, with the only decrease being observed in the A_DT_SD (AUC: 0.713) to the A_DT_SD_O (AUC: 0.644).

The second model (A_DT_SC) is similar to the first but uses the SMILE-C score instead of the SMILE score. This model, after limiting the depth to avoid overfitting, it requires the fewest number of variables (only two) to perform the classification: SMILE-C score and gender (see Figure 27).

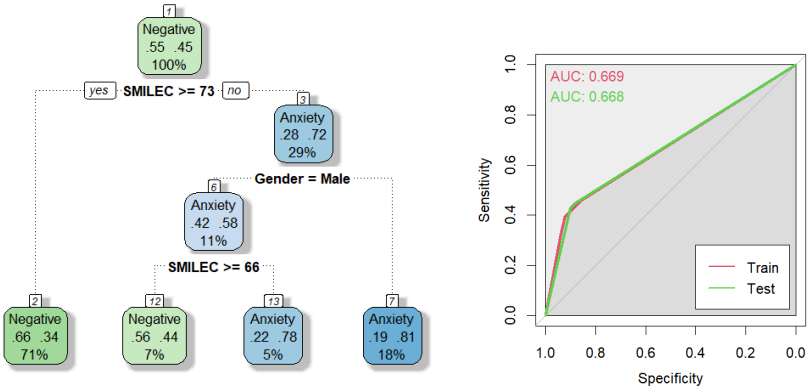

Figure 27: Decision tree and ROC curve for screening for anxiety through SMILE-C and socio-demographic features (controlled for overfitting)

The third model (A_DT_SD) uses the seven domain scores (DN, SU, PA, SM, RS, SS, and EE) instead of the SMILE score. This model, in its simplified version, still uses seven features to perform the classification: DN, SM, SS, gender, age, school, and displacement status (see Figures 28 and 29).

The fourth model (A_DT_S_O) is similar to the first but uses oversampling. This model, as shown in Table 8, has the best AUC (0.727) and F_1 score (0.647) among the decision tree model, and uses 5 features to perform the classification: gender, age, school, course, and SMILE score (see Figure 30).

The fifth model (A_DT_SC_O) is similar to the second but uses oversampling. This model uses 3 features to perform the classification: gender, school, and SMILE-C score (see Figure 31).

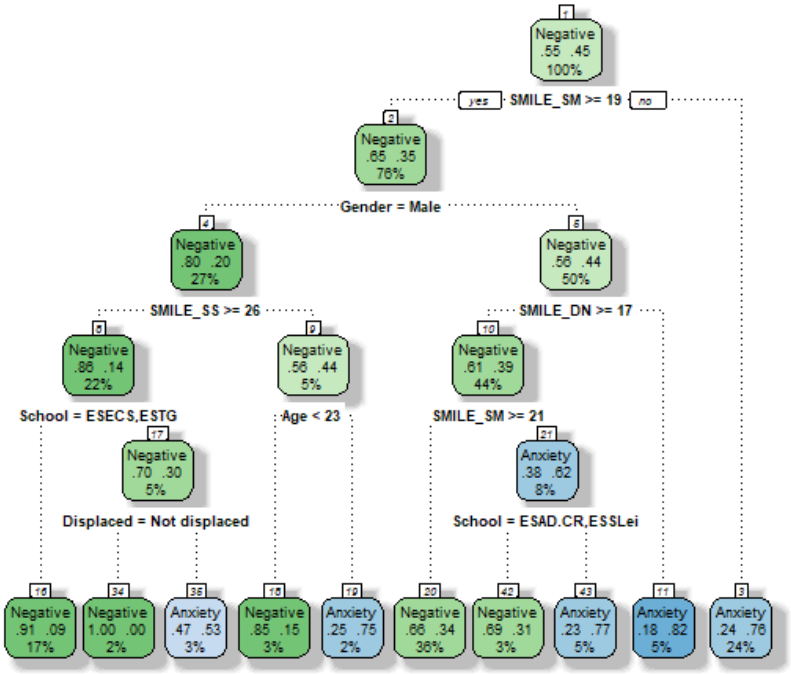

Figure 28: Decision tree for screening for anxiety through SMILE domains and socio-demographic features (controlled for overfitting)

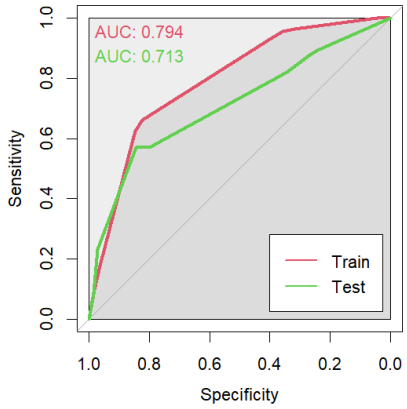

Figure 29: ROC curve for screening for anxiety through SMILE domains and socio-demographic features (controlled for overfitting)

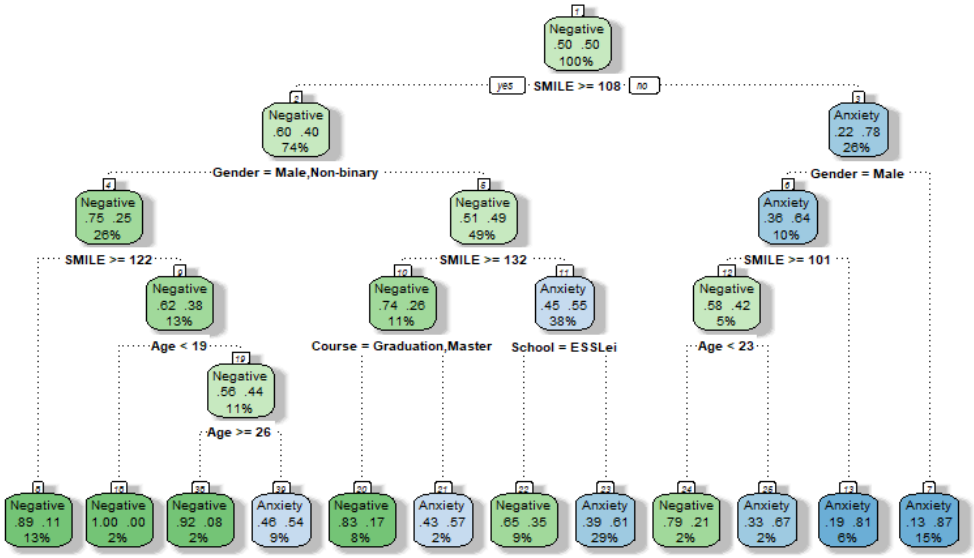

Figure 30: Decision tree for screening for anxiety through SMILE and socio-demographic features (with oversampling)

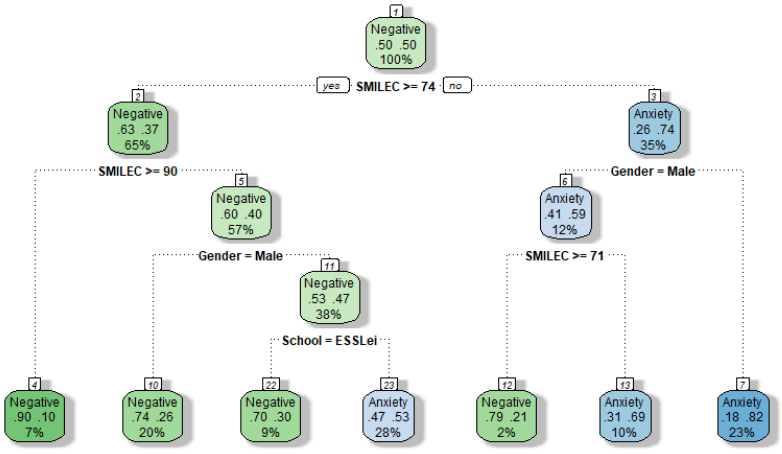

Figure 31: Decision tree for screening for anxiety through SMILE-C and socio-demographic features (with oversampling)

The sixth model (A_DT_SD_O) is similar to the third, using the domain scores instead of the SMILE score but using oversampling. This model uses three features to classify: DN, SM, and course (see Figure 32).

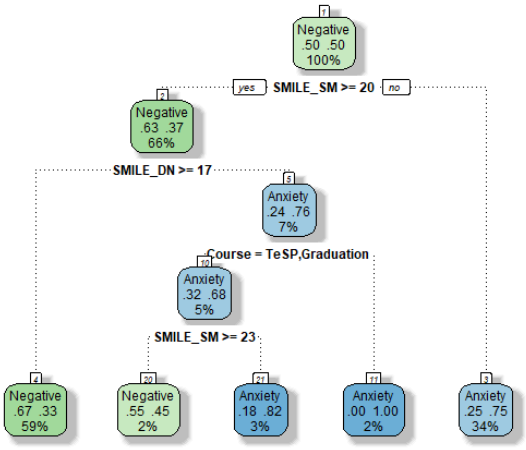

Figure 32: Decision tree for screening for anxiety through SMILE domains and socio-demographic features (with oversampling)

As described in Table 8 and Figure 33, the model A_DT_S_O has the better AUC (0.727), and A_DT_SD_O has the worst (AUC: 0.644), demonstrating that oversampling has different effects depending on if you use the SMILE score or the scores of its domains.

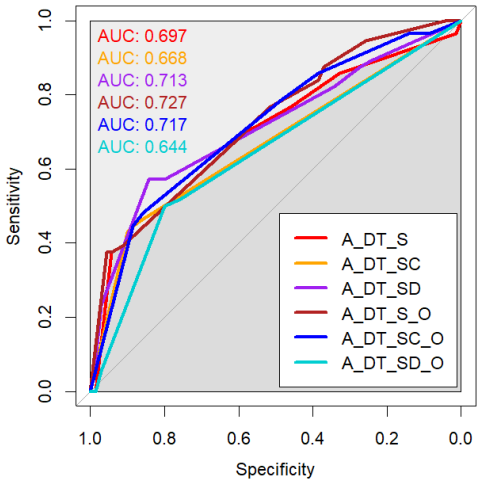

Figure 33: ROC curves of screening for anxiety of the six multivariate models analyzed through decision trees

In what concerns feature importance in each model, Figure 34 shows that the scores are always the most important variable when trying to predict anxiety screening. From the socio-demographic variable, the most important one is gender, followed by school.

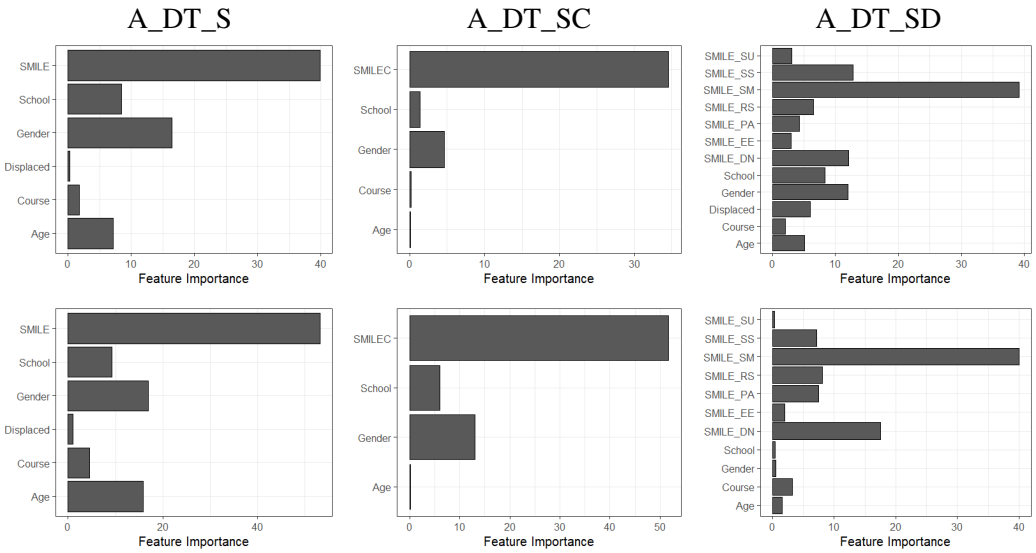

Figure 34: Features importance in each decision tree model for screening anxiety, without (top) and with (bottom) oversampling

4.6.1.3 *Logistic regression for screening for anxiety through multiple variables*

Logistic regression (LR) was applied to screening for anxiety through multiple variables. Hence, as in the DT analysis, in the first model (A_LR_S), the SMILE score and all socio-demographic features are used as explanatory variables. Subsequently, adhering to the principle of parsimony, features that were found to be non-significant (according to the Wald test) were sequentially eliminated. Thus, in the LR parsimonious model for the first model (A_LR_S), only SMILE, gender, and School variables are significant. The estimated model is given by

$$\begin{aligned}\widehat{\text{logit}\,(p_A)} &= 8.5793 - 0.0720\,\text{SMILE} - 1.2885\,G_{\text{Male}} + 0.0859\,G_{\text{Non-binary}} \\ &\quad +0.2812 S_{\text{ESECS}} - 0.5745 S_{\text{ESSLei}} + 0.0853 S_{\text{ESTG}} + 0.7088 S_{\text{ESTM}},\end{aligned}$$

where $\text{logit}(p) = \ln(\text{odds}(p))$, with $\text{odds}(p) = p\ (1-p)^{-1}$, is the inverse of the logistic function given by $\text{logistic}(\alpha) = \exp(\alpha)\,(1+\exp(\alpha))^{-1}$ and p_A is the probability of the student suffering from anxiety (James et al., 2021). In the explanatory features, SMILE is a quantitative variable (score in the SMILE survey), while the others are binary dummy variables, i.e.,

- G_{Male} equal to 1 if the student is a male and is equal to 0 otherwise (Gender feature);

- $G_{\text{Non-binary}}$ equal to 1 if the student is non-binary and is equal to 0 otherwise (Gender feature);
- S_{ESECS} equal to 1 if the student is from the ESECS school and is equal to 0 otherwise (School feature);
- S_{ESSLei} equal to 1 if the student is from the ESSLei school and is equal to 0 otherwise (School feature);
- S_{ESTG} equal to 1 if the student is from the ESTG school and is equal to 0 otherwise (School feature);
- S_{ESTM} equal to 1 if the student is from the ESTM school and is equal to 0 otherwise (School feature).

According to the Hosmer and Lemeshow goodness of fit test (p-value = 0.1443), the estimated probabilities of anxiety are close to the true probabilities. For the classification procedure, a student is classified as suffering from anxiety if $p_A > 0.5$, i.e., if the probability of suffering from anxiety is greater than the probability of not suffering from anxiety. The classification performance measures of this model are presented in Table 8, and the ROC curve is displayed in Figure 35.

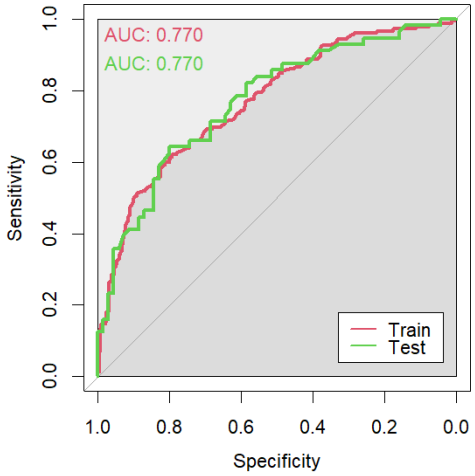

Figure 35: ROC curves of screening for anxiety through SMILE and socio-demographic features (logistic regression)

The second model (A_LR_SC) uses the SMILE-C score instead of SMILE and, in its parsimonious form, is given by

$$\begin{aligned}\widehat{\text{logit}\,(p_A)} &= 9.5534 - 0.1138\,\text{SMILE} - \text{C} - 0.0257\text{Age} \\ &\quad -1.1680\,G_{\text{Male}} + 0.0189\,G_{\text{Non-binary}}\end{aligned}$$

where SMILE – C is the score of the SMILE-C survey and Age corresponds to the age of the student (numerical variable). The Hosmer and Lemeshow goodness of fit test reveals no problems of adjustment (p-value =0.281). The classification measures are available in Table 8 and Figure 36.

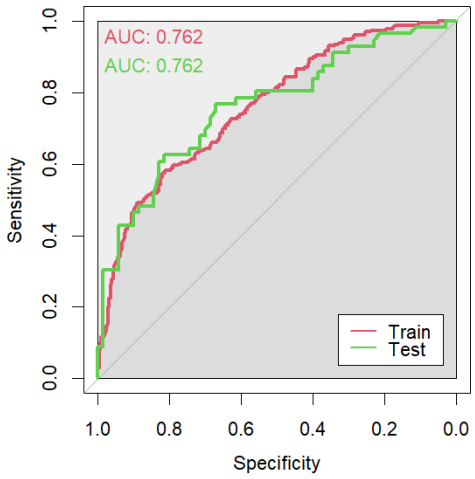

Figure 36: ROC curves of screening for anxiety through SMILE-C and socio-demographic features (logistic regression)

The parsimonious form of the third model (A_LR_SD) uses the seven domains of the SMILE survey and is given by

$$\begin{aligned}\widehat{\text{logit}\,(p_A)} &= 7.3327 - 0.0982\,\text{SMILE}_\text{D}\text{N} - 0.1712\,\text{SMILE}_\text{S}\text{M} - 0.1237\,\text{SMILE}_\text{R}\text{S} \\ &\quad -1.1627\,G_\text{Male} + 0.3988\,G_\text{Non-binary} \\ &\quad +0.2247\,S_\text{ESECS} - 0.5455\,S_\text{ESSLei} + 0.0522\,S_\text{ESTG} + 0.6634\,S_\text{ESTM},\end{aligned}$$

where $\text{SMILE}_\text{D}\text{N}$, $\text{SMILE}_\text{S}\text{M}$, and $\text{SMILE}_\text{R}\text{S}$ are, respectively, the scores of the domains DN, SM, and RS. Also, in this model, the Hosmer and Lemeshow goodness of fit test reveals no problems with the adjustment (p-value =0.4044). The classification measures are available in Table 8 and Figure 37.

The same three models were estimated with a balanced training sample, where the training sample has been balanced using oversampling techniques. To summarize, the first parsimonious model (A_LR_S_O) is given by

$$\begin{aligned}\widehat{\text{logit}\,(p_A)} &= 8.9888 - 0.0728\,\text{SMILE} - 1.2129\,G_\text{Male} + 0.1350\,G_\text{Non-binary} \\ &\quad +0.1800\,S_\text{ESECS} - 0.7337\,S_\text{ESSLei} - 0.0814\,S_\text{ESTG} + 0.5880\,S_\text{ESTM},\end{aligned}$$

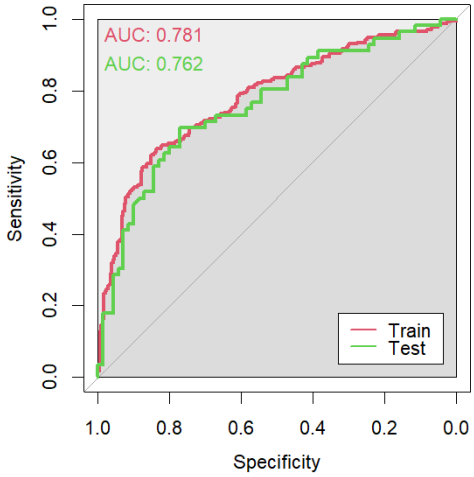

Figure 37: ROC curves of screening for anxiety through SMILE domains and socio-demographic features (logistic regression)

with a p-value of 0.0881 in the Hosmer and Lemeshow goodness of fit test (null hypotheses not rejected). The second parsimonious model (A_LR_SC_O) is given by

$$\begin{aligned}\widehat{\text{logit}\,(p_A)} \;=\; & 9.8152 - 0.1157\,\text{SMILE} - \text{C} - 0.0250\,\text{Age} \\ & -1.1301\,G_{\text{Male}} + 0.1593\,G_{\text{Non-binary}} + \\ & 0.3937\,S_{\text{ESECS}} - 0.5845\,S_{\text{ESSLei}} + 0.0728\,S_{\text{ESTG}} + 0.4783\,S_{\text{ESTM}},\end{aligned}$$

with a p-value of 0.0679 in the Hosmer and Lemeshow goodness of fit test. Finally, the last model (A_LR_SD_O), in its parsimonious form, is given by

$$\begin{aligned}\widehat{\text{logit}\,(p_A)} \;=\; & 7.6394 - 0.0984\,\text{SMILE}_{\text{D}}\text{N} - 0.1756\,\text{SMILE}_{\text{S}}\text{M} - 0.1151\,\text{SMILE}_{\text{R}}\text{S} \\ & -1.0829\,G_{\text{Male}} + 0.3505\,G_{\text{Non-binary}} \\ & +0.0693\,S_{\text{ESECS}} - 0.7128\,S_{\text{ESSLei}} - 0.1470\,S_{\text{ESTG}} + 0.4764\,S_{\text{ESTM}},\end{aligned}$$

with a p-value of 0.0553 in the Hosmer and Lemeshow goodness of fit test (null hypotheses not rejected).

The ROC curve of the training and testing sample in the three models used with a balanced training sample is provided in Figure 38. Those graphs reveal no significant difference between the two curves, and consequently, the null hypothesis is not rejected in DeLong's test for equality of two ROC curves (p-values equals 0.982, 0.9454, and 0.6699, respectively). In fact, the proportion of students with anxiety is not quite different between the two samples, as in the test sample, the number of students with anxiety (44.4%) is just a little lower than those without anxiety (55.6%).

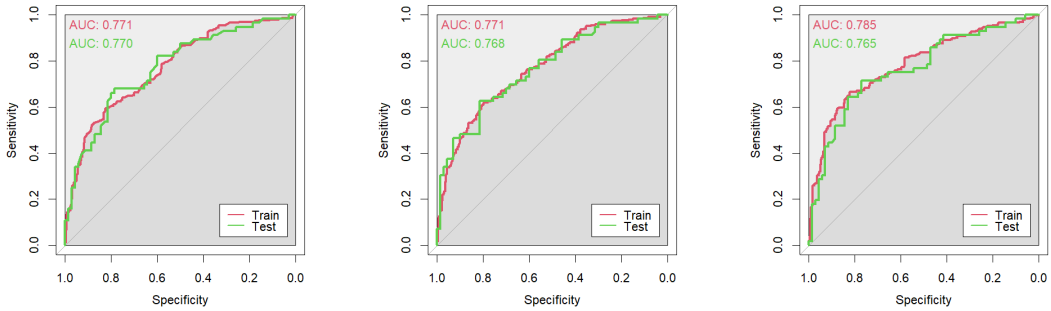

Figure 38: ROC curves from the train and test samples on screening for anxiety of the three balanced multivariate models.

Figure 39 displays the ROC curve of the six estimated models, and Table 8 summarizes all models of screening anxiety. The logistic regression model, A_LR_S (3 variables), achieved the highest AUC of 0.770, the same as the A_LR_S_O (3 variables), but all models displayed satisfactory results.

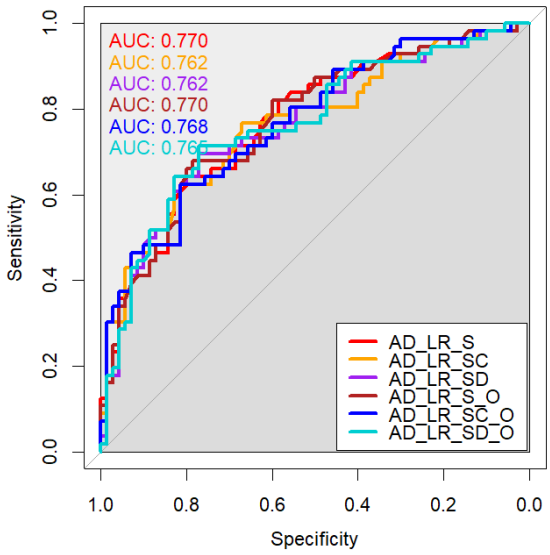

Figure 39: ROC curves of screening for anxiety of the 6 multivariate models analyzed through logistic regression

The main takeaway is that the results for the logistical regression models with or without oversampling are almost the same, with no statistically significant changes in most evaluation metrics. However, there was an improvement in sensitivity, an important benchmark for avoiding missing anxious individuals.

4.6.2 *Depression*

In this section, we delve into the screening of depression by utilizing SMILE and SMILE-C scores, their domains, and other socio-demographic variables gathered during the study. In the same way done with anxiety, we employ the prediction models of logistic regression and decision trees. In cases where only one variable was utilized, the decision tree model was preferred.

4.6.2.1 *Screening for depression through a single variable*

Considering the ROC curve and the area under the curve (AUC) as shown in Figure 40, the best model to predict the screening of depression is the SMILE score (AUC: 0.752), followed by the SMILE-C score (AUC: 0.741). Considering only a domain, the best is still stress management (AUC: 0.727), but there is a change in the worst where instead of the environment exposures domain (AUC: 0.627), it is substance abuse that has the lower AUC (0.520).

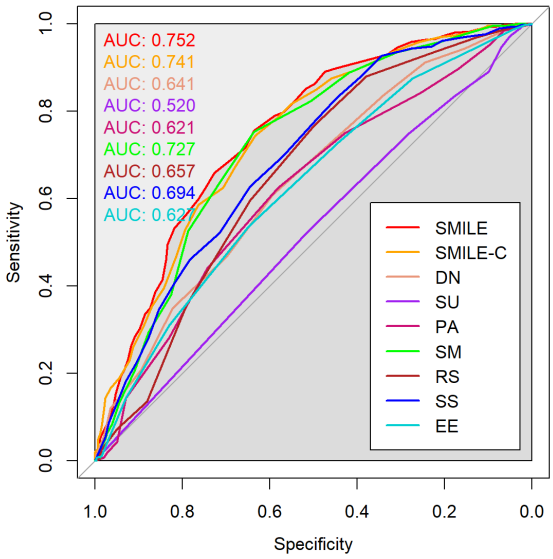

Figure 40: ROC curves of screening for depression (entire sample)

Figure **??** exemplifies the cut-off point for the prediction of depression of the decision tree models with one variable, where when the SMILE score is equal to or greater than 115, there is a 60% chance of a negative screening for anxiety. Otherwise, there is a chance of 40% of having depression issues.

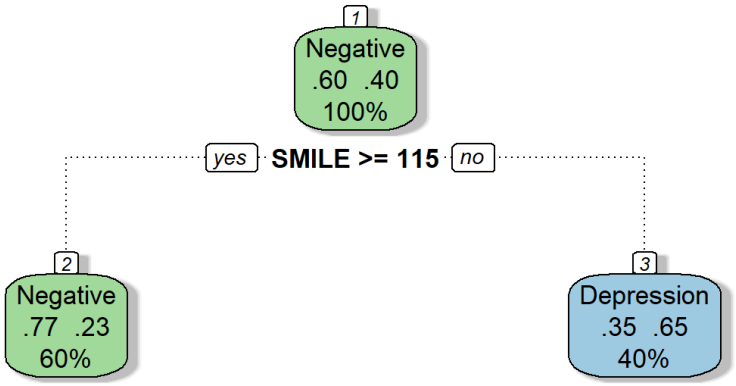

Figure 41: Decision tree for screening for depression through SMILE score

The results for the SMILE and SMILE-C scores and their domains are summarized in Table 9.

- Accuracy (Ac): All models except EE have an accuracy between 0.603 and 0.709, indicating a moderate ability to classify individuals with or without depression correctly.
- Sensitivity (Se): Ranges from 0.020 (SU) to 0.620 (SM), indicating some models miss a significant portion of individuals with depression (low sensitivity), but it has better values than one measured for anxiety models.
- Specificity (Sp): Ranges from 0.632 (PA) to 0.987 (SU), indicating better classification of healthy individuals as depressed than the anxiety models.
- Positive Predictive Value (PPV): Ranges from 0.500 (Su, PA) to 0.783 (SS), indicating a moderate probability that a positive test result reflects true depression.
- Negative Predictive Value (NPV): Ranges from 0.603 (EE, again, all negative) to 0.747 (SM), indicating a moderate probability that a negative test result reflects a truly healthy individual.
- F1 Score: Ranges from 0.038 (SU) to 0.614 (SM), summarizing precision and recall into a single metric.
- AUC: Ranges from 0.500 (EE, random guess) to 0.678 (SM).

Overall, the SMILE score and SS domain show the strongest performance, with accuracy (SMILE: 0.659, SS:0.706), F_1 score (SMILE:0.574, SS:0.493), and AUC (SMILE:0.645, SS:0.647). However, the same as with anxiety, all models have limitations, and more

Table 9: One feature to screen for depression. **Cut-off**: cut-off point – an individual is classified with depression if the feature is lower than the cut-off point; **Ac**: accuracy; **Se**: sensitivity; **Sp**: specificity; **PPV**: positive predicted value; **NPV**: negative predicted value; F_1: F_1 score; **AUC**: area under the ROC curve.

Feature	Cut-off	Ac	Se	Sp	PPV	NPV	F_1	AUC
SMILE	115	0.659	0.580	0.711	0.569	0.720	0.574	0.645
SMILE-C	74	0.659	0.480	0.776	0.585	0.694	0.527	0.628
DN	17	0.611	0.200	0.882	0.526	0.626	0.290	0.541
SU	10	0.603	0.020	0.987	0.500	0.605	0.038	0.603
PA	10	0.603	0.560	0.632	0.500	0.686	0.528	0.596
SM	21	0.690	0.620	0.737	0.608	0.747	0.614	0.678
RS	12	0.675	0.380	0.868	0.655	0.680	0.481	0.624
SS	25	0.706	0.360	0.934	0.783	0.689	0.493	0.647
EE	—	0.603	0	1	—	0.603	—	0.500

variables are needed for a better result. Following the same example as before, considering crucial a high sensitivity, when using only one variable, the SM domain model might be preferred. The worst model was the EE domain model, classified as negative for depression screening in all the population with an AUC of 0.5, indicating it does not provide any useful information for depression prediction.

4.6.2.2 *Decision tree for screening for depression through multiple variables*

The performance of these multi-variable models (Table 10) when compared to the single-variable models (Table 9), the decision tree model D_DT_SD_O (2 variables) achieved an AUC of 0.690, which is marginally better than the best single-variable model (SM, AUC: 0.678). Oversampling achieves slightly better results in the decision tree models, with the only decrease being observed in the A_DT_S (AUC: 0.647) to the A_DT_S_O (AUC: 0.645).

The First model (D_DT_S) uses the SMILE score as a variable. After limiting the depth to avoid overfitting, in its simplified version, this model uses two features to perform the classification: SMILE and school. There is some difference in reliability between the training and test samples, as shown in Figure 42, but not significant. Still, the remaining reliability measures of this classification are presented in Table 10.

The second model (D_DT_SC) uses the SMILE-C score instead of the SMILE score. After limiting the depth to avoid overfitting, in its simplified version, this model uses three

Table 10: Multivariate analysis to screen for depression. **NV**: number of variables used in the screening procedure; **Ac**: accuracy; **Se**: sensitivity; **Sp**: specificity; **PPV**: positive predicted value; **NPV**: negative predicted value; F_1: F_1 score; **AUC**: area under the ROC curve.

Model	NV	Ac	Se	Sp	PPV	NPV	F_1	AUC
D_DT_S	2	0.611	0.400	0.750	0.513	0.655	0.449	0.647
D_DT_SC	3	0.675	0.460	0.816	0.622	0.697	0.529	0.628
D_DT_SD	7	0.706	0.600	0.776	0.638	0.747	0.619	0.681
D_DT_S_O	1	0.659	0.580	0.711	0.569	0.720	0.574	0.645
D_DT_SC_O	1	0.651	0.560	0.711	0.560	0.711	0.560	0.635
D_DT_SD_O	2	0.667	0.700	0.645	0.565	0.766	0.625	0.690
D_LR_S	5	0.690	0.520	0.803	0.634	0.718	0.571	0.725
D_LR_SC	4	0.690	0.480	0.829	0.649	0.708	0.552	0.709
D_LR_SD	8	0.698	0.540	0.803	0.643	0.726	0.587	0.723
D_LR_S_O	6	0.675	0.680	0.671	0.576	0.761	0.624	0.705
D_LR_SC_O	5	0.635	0.620	0.645	0.534	0.721	0.574	0.688
D_LR_SD_O	8	0.683	0.680	0.684	0.586	0.765	0.630	0.717

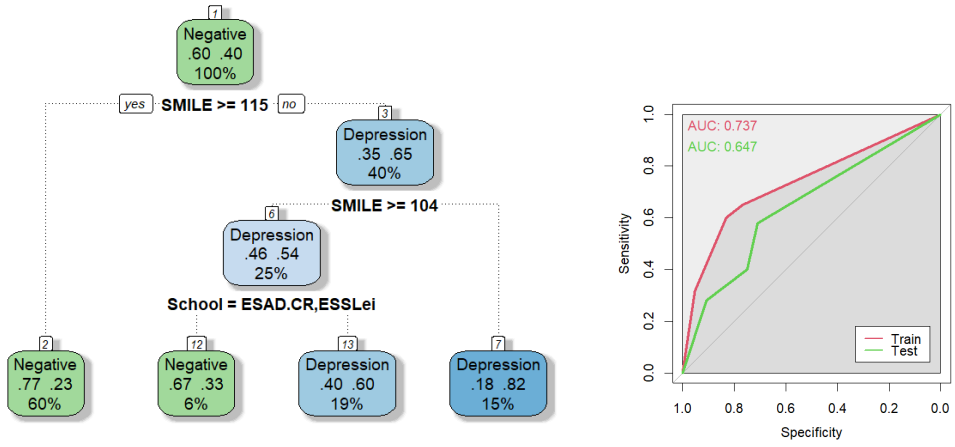

Figure 42: Decision tree and ROC curve for screening for depression through SMILE and socio-demographic features

features to perform the classification: SMILE-C, age, and gender (see Figure 43 and Table 10).

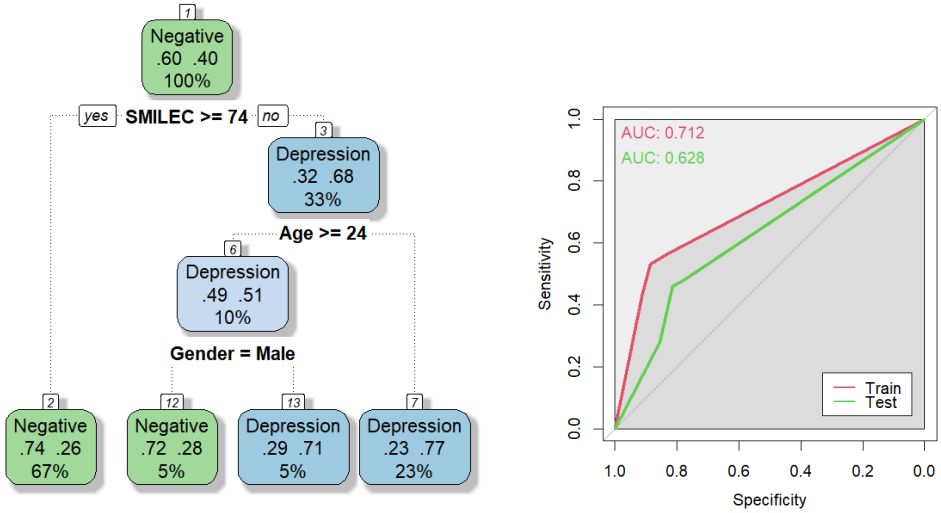

Figure 43: Decision tree and ROC curve for screening for depression through SMILE-C and socio-demographic features

The third model (D_DT_SD) uses the seven domain scores (DN, SU, PA, SM, RS, SS, and EE) instead of the SMILE score. In its simplified version, this model uses seven features to perform the classification: DN, SM, RS, SS, school, age, and gender (see Figures 44 and 44 and Table 10).

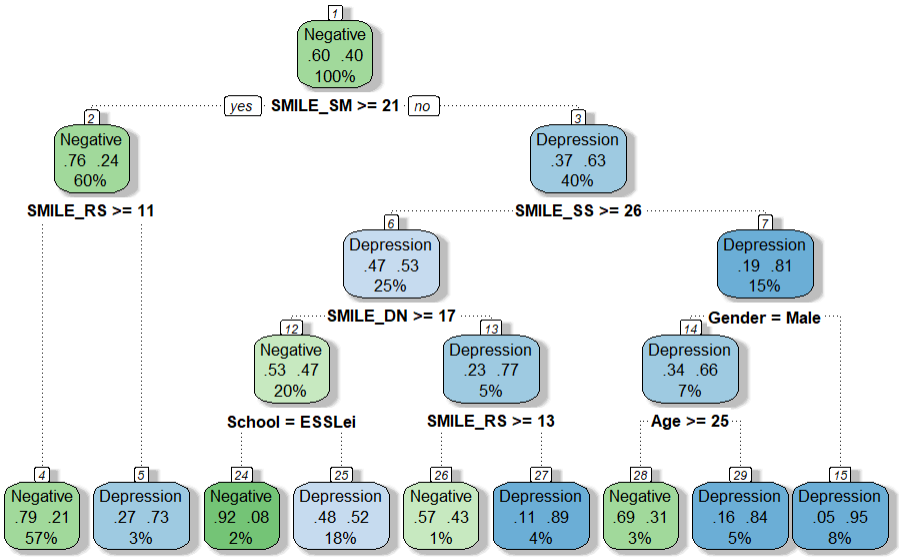

Figure 44: Decision tree for screening for depression through SMILE domains and socio-demographic features

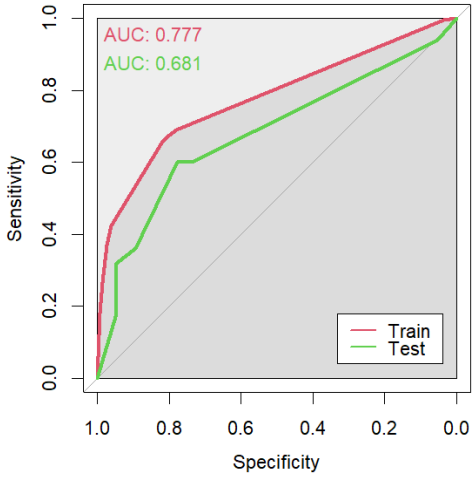

Figure 45: ROC curve for screening for depression through SMILE domains and socio-demographic features

The fourth model (D_DT_S_O) is similar to the first but uses oversampling. This model uses only the SMILE score feature to perform the classification (see Figure 46 and Table 10).

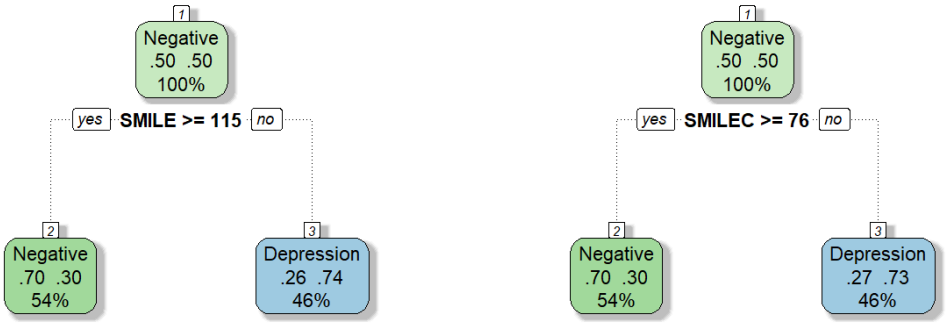

Figure 46: Decision tree for screening for depression through SMILE (left) and SMILE-C (right) domains and socio-demographic features (with oversampling)

The fifth model (D_DT_SC_O) is similar to the second but uses oversampling. This model also uses only one feature, the SMILE-C score, to perform the classification (see Figure 46 and Table 10).

The sixth model (A_DT_SD_O) is similar to the third, using the domain scores instead of the SMILE score but using oversampling. This model uses two features to perform the

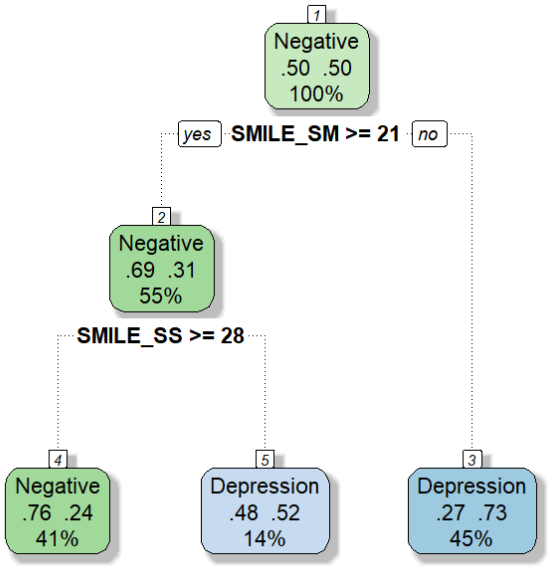

Figure 47: Decision tree for screening for depression through SMILE domains and socio-demographic features (with oversampling)

classification: SS, and SM (see Figure 47). As shown in Table 10 and Figure 48, it has the best AUC (0.690) and F_1 score (0.625) among the decision tree model for depression.

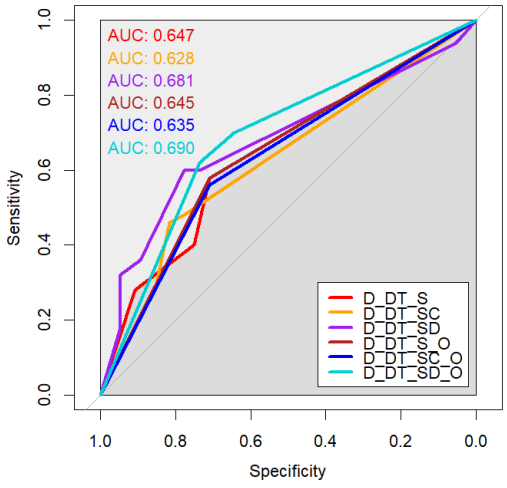

Figure 48: ROC curves of screening for depression of the 6 multivariate models analyzed through decision trees

In what concerns features importance in each model, Figure 49 shows that differently from the anxiety decision tree models, those about depression do not consider the socio-demographic variables as important, especially when oversampling is done in the training sample, none of those variables were used.

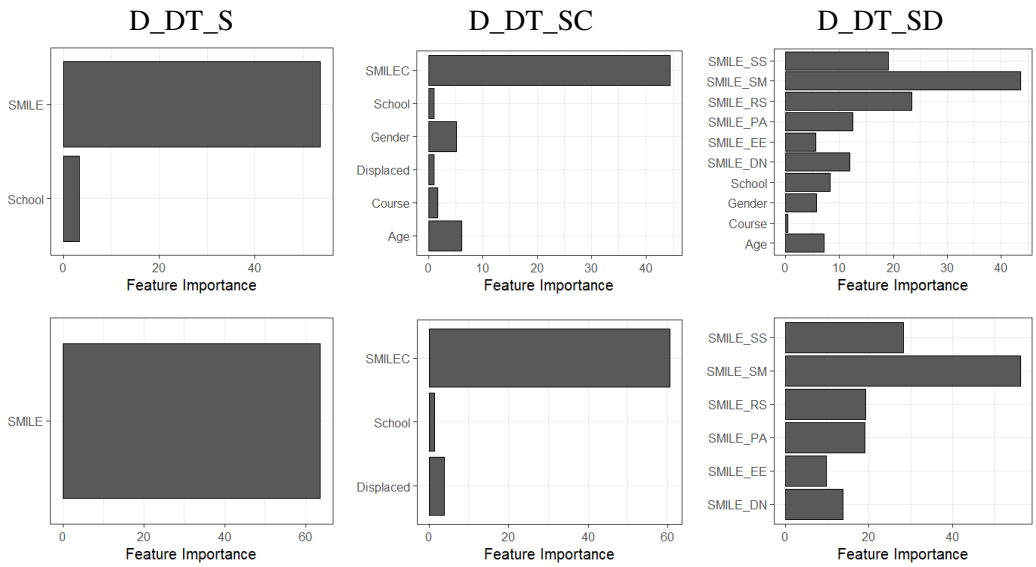

Figure 49: Features importance in each decision trees model for screening depression, without (top) and with (bottom) oversampling

4.6.2.3 *Logistic regression for screening for depression through multiple variables*

In this section, logistic regression (LR) was applied to screening for depression through multiple variables following the same three models as in Section 4.6.1.3 for screen for anxiety. Moreover, the principle of parsimony continues to be applied to each model through the application of the Wald test with a 5% significance level. The performance of the screening procedure associated with the LR models is presented in Table 8.

The first model (D_LR_S) screens for depression through the SMILE score and all the socio-demographic features. The estimated model is given by

$$\begin{aligned}\widehat{\text{logit}\,(p_A)} \;=\; & 11.7610 - 0.0916\,\text{SMILE} - 0.6498\,G_{\text{Male}} + 0.0121\,G_{\text{Non-binary}} \\ & -0.0600\,\text{Age} - 0.0711\,C_{\text{Graduation}} + 0.6226\,C_{\text{Master}} \\ & +0.5748\,S_{\text{ESECS}} - 0.1158\,S_{\text{ESSLei}} + 0.3953\,S_{\text{ESTG}} + 1.1910\,S_{\text{ESTM}},\end{aligned}$$

where p_D is the probability of the student suffering from depression and

- $C_{\text{Graduation}}$ equals to 1 if the student is in a graduation course and is equal to 0 otherwise (Gender feature);
- C_{Master} equals to 1 if the student is in a master course and is equal to 0 otherwise (Gender feature).

The estimated probabilities of depression are close to the true probabilities as the null hypothesis is not rejected in the Hosmer and Lemeshow goodness of fit test (p-value

= 0.9611). The ROC curve associated with the depression classification carried out with model D_LR_S is shown in Figure 50.

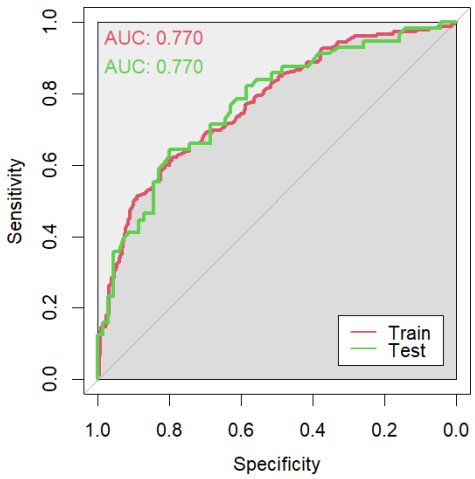

Figure 50: ROC curves of screening for depression through SMILE and socio-demographic features (logistic regression)

Based on the SMILE-C score, the second model (D_LR_SC) is given by

$$\begin{aligned}\widehat{\text{logit}\,(p_D)} \;=\;& 11.0506 - 0.1325\,\text{SMILE} - \text{C} - 0.0653\,\text{Age}\\ &+0.0088\,C_{\text{Graduation}} + 0.5811\,C_{\text{Master}}\\ &+0.8029\,S_{\text{ESECS}} + 0.1006\,S_{\text{ESSLei}} + 0.3984\,S_{\text{ESTG}} + 1.0901\,S_{\text{ESTM}},\end{aligned}$$

without the null hypotheses of the Hosmer and Lemeshow test being rejected (p-value =0.8079). Figure 51 provides the ROC curve of the classification performed with model D_LR_SC.

Using the scores obtained in each domain of the SMILE survey, the estimates of the third model (D_LR_SD) are given by

$$\begin{aligned}\widehat{\text{logit}\,(p_D)} \;=\;& 10.2454 - 0.0577\,\text{Age} - 0.1183\,\text{SMILE}_{\text{S}}\text{M} - 0.1522\,\text{SMILE}_{\text{R}}\text{S}\\ &-0.1111\,\text{SMILE}_{\text{S}}\text{S} - 0.1421\,\text{SMILE}_{\text{E}}\text{E}\\ &-0.7218\,G_{\text{Male}} + 0.0597\,G_{\text{Non-binary}}\\ &0.4157\,S_{\text{ESECS}} - 0.2706\,S_{\text{ESSLei}} + 0.2484\,S_{\text{ESTG}} + 1.1656\,S_{\text{ESTM}},\end{aligned}$$

with a p-value equal to 0.1549 in the Hosmer and Lemeshow goodness of fit test, revealing no problems with the adjustment. Its ROC curve is shown in Figure 52.

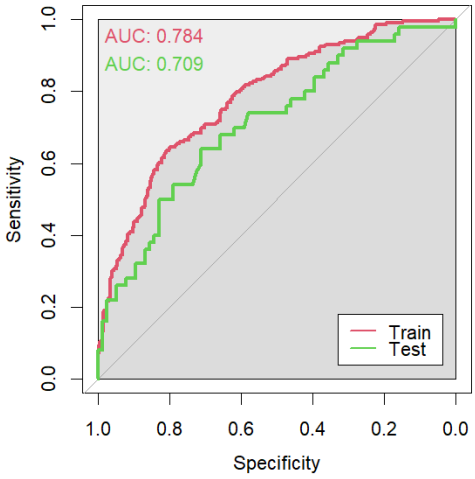

Figure 51: ROC curves of screening for depression through SMILE-C and socio-demographic features (logistic regression)

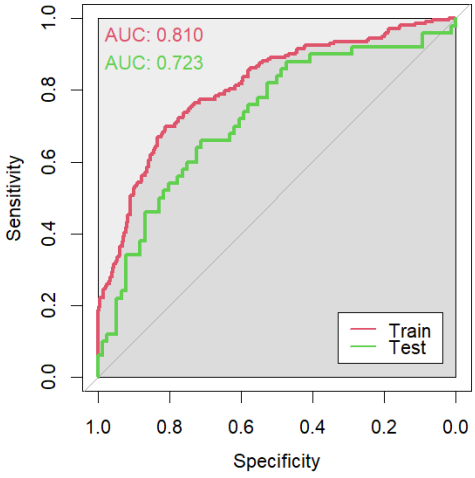

Figure 52: ROC curves of screening for depression through SMILE domains and socio-demographic features (logistic regression)

Estimating the same three models with the balanced training sample using oversampling, the estimates of the first model (D_LR_S_O) are given by

$$\begin{aligned}\widehat{\text{logit}\,(p_D)} &= 13.0921 - 0.0997\,\text{SMILE} - 0.0703\,\text{Age} - \\ &\quad -0.5383\,G_{\text{Male}} - 0.3385\,G_{\text{Non-binary}} + \\ &\quad -0.0226\,C_{\text{Graduation}} + 0.6940\,C_{\text{Master}} + 0.5099\,S_{\text{ESECS}} - \\ &\quad -0.0444\,S_{\text{ESSLei}} + 0.4457\,S_{\text{ESTG}} + 1.2731\,S_{\text{ESTM}} + 0.4216\,\text{Displaced},\end{aligned}$$

where Displaced is a binary dummy variable which is equal to 1 if the student is displaced and is equal to 0 otherwise (Displaced feature). This model attained a p-value of 0.2656 in the Hosmer and Lemeshow goodness of fit test, and, therefore, the null hypothesis is not rejected. The second model (D_LR_SC_O) is given by

$$\begin{aligned}\widehat{\text{logit}\,(p_D)} &= 12.5380 - 0.1476\,\text{SMILE} - \text{C} - 0.0734\,\text{Age} - \\ &\quad +0.0236\,C_{\text{Graduation}} + 0.6400\,C_{\text{Master}} + 0.8105\,S_{\text{ESECS}} + \\ &\quad +0.1433\,S_{\text{ESSLei}} + 0.4926\,S_{\text{ESTG}} + 1.1814\,S_{\text{ESTM}} + 0.4407\,\text{Displaced},\end{aligned}$$

with a p-value of 0.9981 in the Hosmer and Lemeshow goodness of fit test. Thus, it seems that the model's estimates fit the data. Finally, the last model to screen for depression (D_LR_SD_O) is given by

$$\begin{aligned}\widehat{\text{logit}\,(p_D)} &= 10.8357 - 0.0604\,\text{Age} - 0.1155\,\text{SMILE}_{\text{S}}\text{M} - 0.1649\,\text{SMILE}_{\text{R}}\text{S} \\ &\quad -0.1145\,\text{SMILE}_{\text{S}}\text{S} - 0.1449\,\text{SMILE}_{\text{E}}\text{E} \\ &\quad -0.6165\,G_{\text{Male}} - 0.2757\,G_{\text{Non-binary}} \\ &\quad -0.0664\,C_{\text{Graduation}} + 0.6897\,C_{\text{Master}} \\ &\quad +0.4066\,S_{\text{ESECS}} - 0.2124\,S_{\text{ESSLei}} + 0.3647\,S_{\text{ESTG}} + 1.1840\,S_{\text{ESTM}},\end{aligned}$$

with a p-value of 0.0062 in the Hosmer and Lemeshow goodness of fit test. Thus, the null hypothesis is rejected, and therefore, it seems there is a significant difference between the observed and the model-predicted probabilities. Figure 53 shows the ROC curve of the training and testing sample in the three models used with a balanced training sample. This graph reveals a significant difference between the two curves, and consequently, the null hypothesis is rejected in DeLong's test for equality of two ROC curves (p-values equals 0.0323, 0.0299, and 0.0431, respectively). This difference could be derived from the difference in composition between the training sample, which was balanced (same number of students with depression and without depression), and the test sample, where the number of students with depression (39.6%) is lower than those without depression (60.3%).

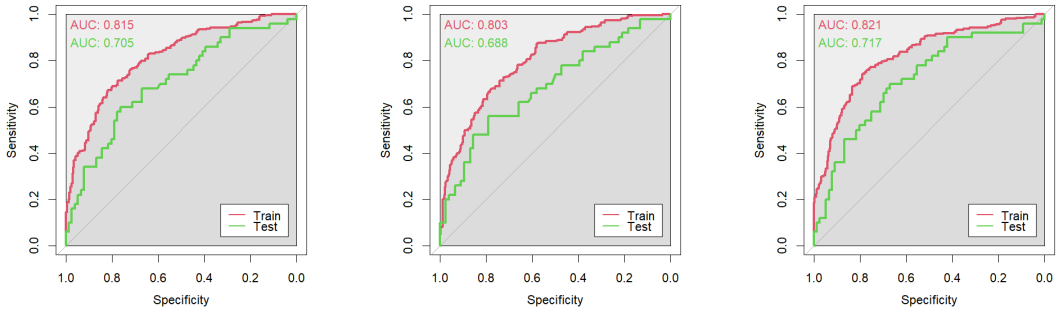

Figure 53: ROC curves from the train and test samples on screening for depression of the three balanced multivariate models.

Figure 54 displays the ROC curve of the six estimated models, and Table 8 summarizes all models of screening anxiety.

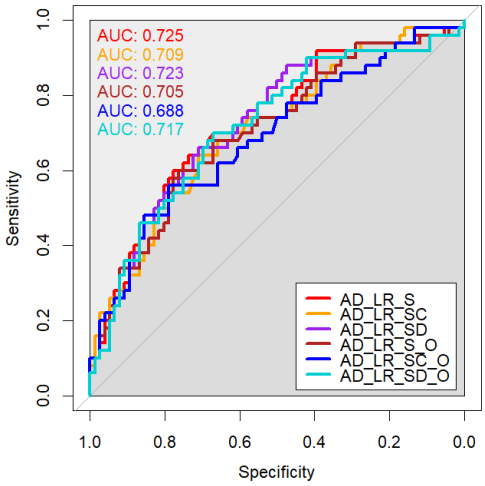

Figure 54: ROC curves of screening for depression of the 6 multivariate models analyzed through logistic regression.

Overall, the results for the logistical regression models mixed results were yielded with oversampling, some evaluation metrics were worsened, with lower accuracy, AUC and specificity. However, there was an improvement in sensitivity, an important benchmark for avoiding missing depressed individuals, and an improvement in F_1 scores.

4.6.3 *Anxiety and Depression*

In this section, we delve into the screening for both depression and anxiety by utilizing SMILE and SMILE-C scores, their domains, and other socio-demographic variables gathered

during the study. Following the same process done with depression and anxiety, we employ the prediction models of logistic regression and decision trees. In cases where only one variable was utilized, the decision tree model was preferred.

4.6.3.1 *Screening for anxiety and depression through a single variable*

Considering the ROC curve and the area under the curve (AUC) as shown in Figure 55, the best model to predict the screening of depression is the SMILE score (AUC: 0.689), followed by the restorative sleep domain score (AUC: 0.681). The worst is the environmental exposures domain (AUC: 0.500) and the physical activity domain, which also has the same AUC.

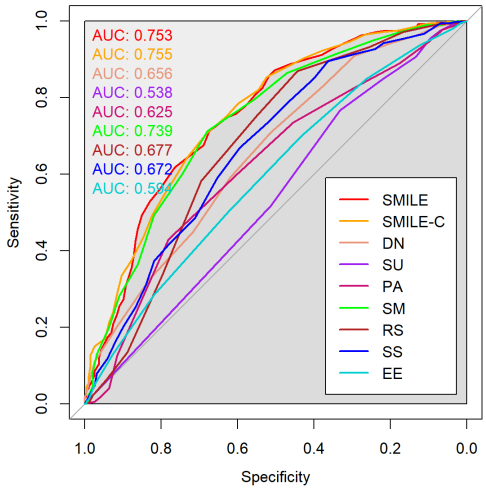

Figure 55: ROC curves of screening for anxiety and depression (entire sample)

Figure 56 exemplifies the cut-off point for the prediction for both anxiety and depression of the decision tree models with one variable, where when the SMILE score is equal to or greater than 115, there is a 76% chance of a negative screening for anxiety. Otherwise, there is a 24% chance of both anxiety and depression issues.

4.6.3.2 *Decision tree for screening for anxiety and depression through multiple variables*

In this section, the same multivariate models were applied for screening for anxiety and depression. For the models without overfitting issues, the model AD_DT_S (SMILE) uses two explanatory variables: SMILE and gender (see Figure 57); the model AD_DT_S (AMILE-C) uses five explanatory variables: SMILE-C, age, school, course, and gender (see Figure 58), and the model AD_DT_SD (SMILE domains) uses six explanatory variables: DN,

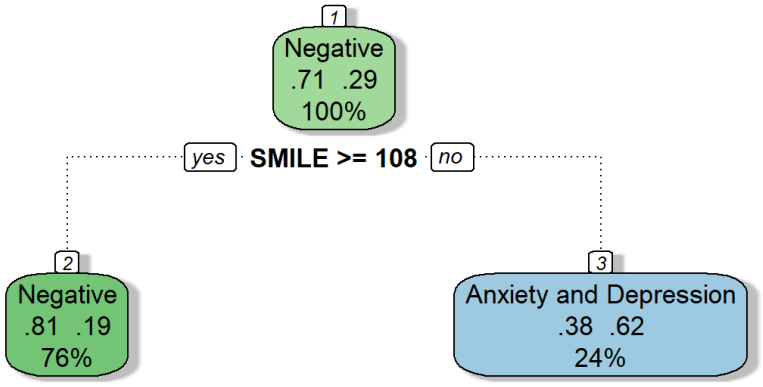

Figure 56: Decision tree for screening for anxiety and depression through SMILE score

Table 11: One feature to screen for anxiety and depression. **Cut-off**: cut-off point – an individual is classified with anxiety and depression if the feature is lower than the cut-off point; **Ac**: accuracy; **Se**: sensitivity; **Sp**: specificity; **PPV**: positive predicted value; **NPV**: negative predicted value; $\boldsymbol{F_1}$: F_1 score; **AUC**: area under the ROC curve.

Feature	**Cut-off**	**Ac**	**Se**	**Sp**	**PPV**	**NPV**	$\boldsymbol{F_1}$	**AUC**
SMILE	108	0.762	0.514	0.865	0.613	0.811	0.559	0.689
SMILE-C	72	0.746	0.486	0.854	0.581	0.800	0.529	0.670
DN	17	0.683	0.216	0.876	0.421	0.729	0.286	0.546
SU	10	0.706	0.027	0.989	0.500	0.710	0.051	0.508
PA	—	0.706	0	1	—	0.706	—	0.500
SM	19	0.690	0.351	0.831	0.464	0.755	0.400	0.591
RS	12	0.762	0.486	0.876	0.621	0.804	0.545	0.681
SS	25	0.730	0.351	0.888	0.565	0.767	0.433	0.619
EE	—	0.706	0	1	—	0.706	—	0.500

RS, SM, SS domains, school and gender (see Figure 59). The main results are summarized in Table 12

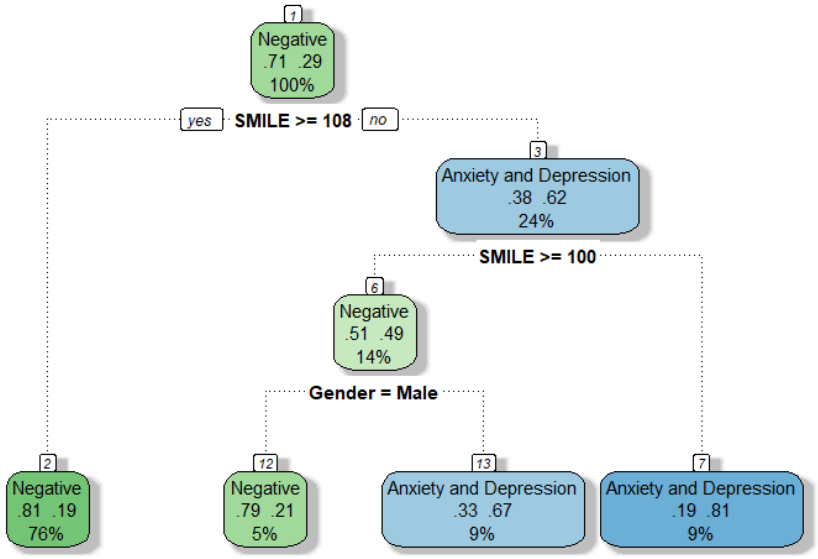

Figure 57: Decision tree for screening for anxiety and depression through SMILE and socio-demographic features

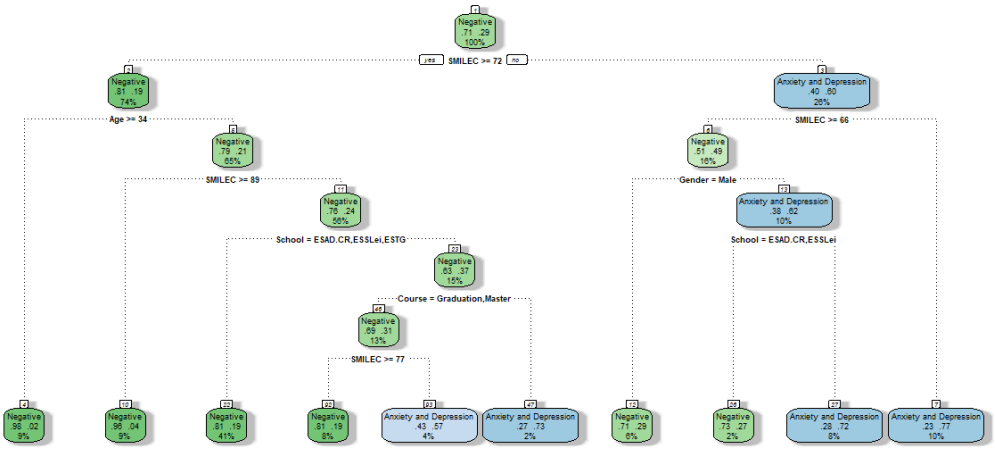

Figure 58: Decision tree for screening for anxiety and depression through SMILE-C and socio-demographic features

For the models with overfitting issues, the model AD_DT_S_O (SMILE) uses three explanatory variables: SMILE, school and gender (see Figure 60); the model AD_DT_S_O (AMILE-C) uses two explanatory variables: SMILE-C and gender (see Figure 61), and the model AD_DT_SD_O (SMILE domains) uses two explanatory variables: RS and SM domains (see Figure 62). The main results are summarized in Table 12 and Figure 64.

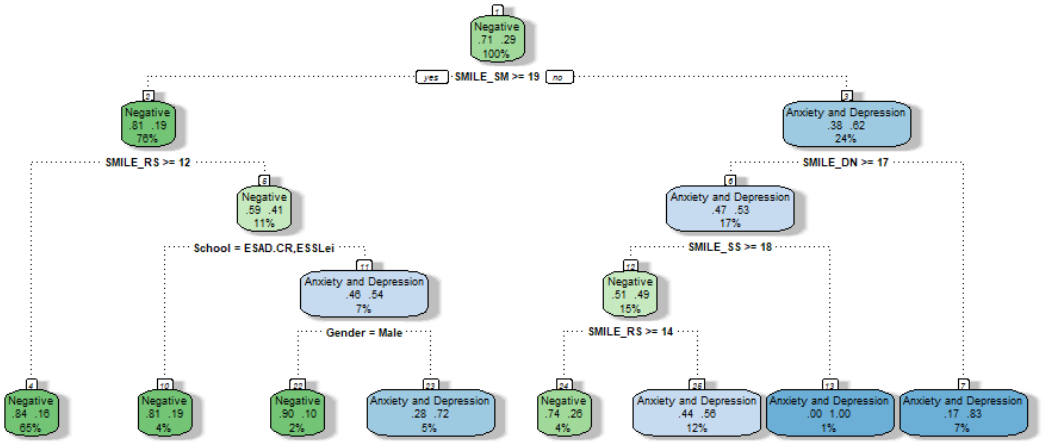

Figure 59: Decision tree for screening for anxiety and depression through SMILE domains and socio-demographic features

Table 12: Multivariate analysis to screen for anxiety and depression. **NV**: number of variables used in the screening procedure; **Ac**: accuracy; **Se**: sensitivity; **Sp**: specificity; **PPV**: positive predicted value; **NPV**: negative predicted value; $\boldsymbol{F_1}$: F_1 score; **AUC**: area under the ROC curve.

Model	**NV**	**Ac**	**Se**	**Sp**	**PPV**	**NPV**	$\boldsymbol{F_1}$	**AUC**
AD_DT_S	2	0.762	0.432	0.899	0.640	0.792	0.516	0.690
AD_DT_SC	5	0.722	0.459	0.831	0.531	0.787	0.493	0.698
AD_DT_SD	6	0.714	0.378	0.854	0.519	0.768	0.438	0.655
AD_DT_S_O	3	0.706	0.622	0.742	0.500	0.825	0.554	0.717
AD_DT_SC_O	2	0.730	0.514	0.820	0.543	0.802	0.528	0.677
AD_DT_SD_O	2	0.698	0.595	0.742	0.489	0.815	0.537	0.663
AD_LR_S	5	0.746	0.432	0.876	0.593	0.788	0.500	0.783
AD_LR_SC	5	0.754	0.459	0.876	0.607	0.796	0.523	0.787
AD_LR_SD	8	0.738	0.405	0.876	0.577	0.780	0.476	0.798
AD_LR_S_O	5	0.722	0.703	0.730	0.520	0.855	0.598	0.782
AD_LR_SC_O	5	0.714	0.730	0.708	0.509	0.863	0.600	0.792
AD_LR_SD_O	9	0.730	0.703	0.742	0.531	0.857	0.605	0.794

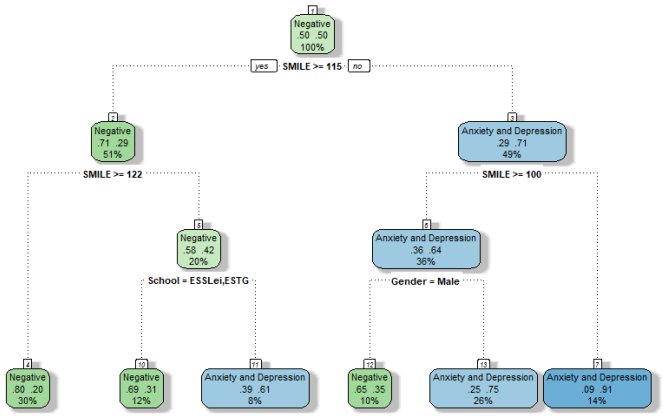

Figure 60: Decision tree for screening for anxiety and depression through SMILE domains and socio-demographic features (with oversampling)

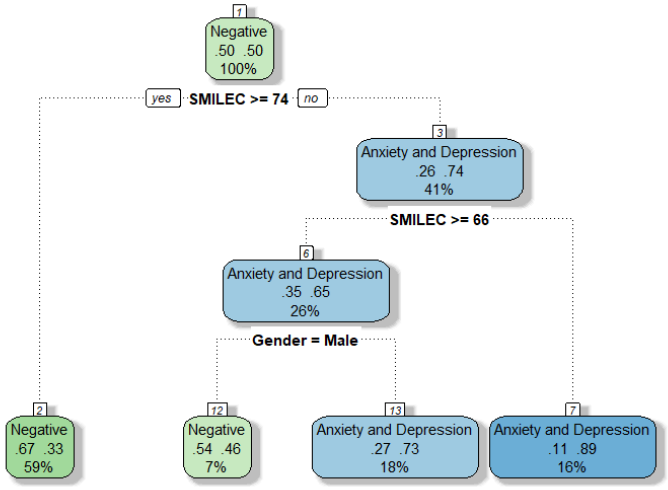

Figure 61: Decision tree for screening for anxiety and depression through SMILE-C (right) domains and socio-demographic features (with oversampling)

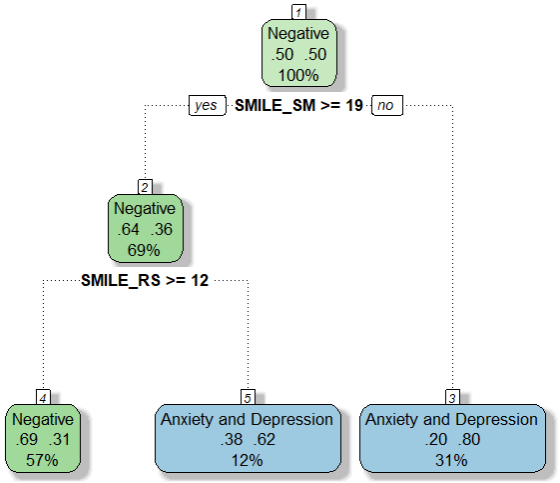

Figure 62: Decision tree for screening for anxiety and depression through SMILE domains and socio-demographic features (with oversampling)

In general, it is not clear that the oversampling has improved the quality of the models, as some improved and others worsened. Furthermore, the model with SMILE seems to have better performance than the others. The main results are summarized in Table 12 and Figure 64.

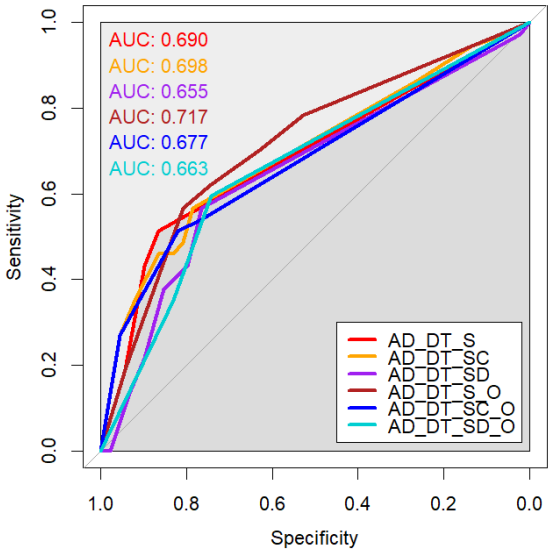

Figure 63: ROC curves of screening for anxiety and depression of the six multivariate models analyzed through decision trees

Regarding feature importance in each model, Figure 64 illustrates that the SMILE, SMILE-C, and domain scores play crucial roles in each model, while socio-demographic variables exhibit low importance, particularly when oversampling is implemented in the training sample.

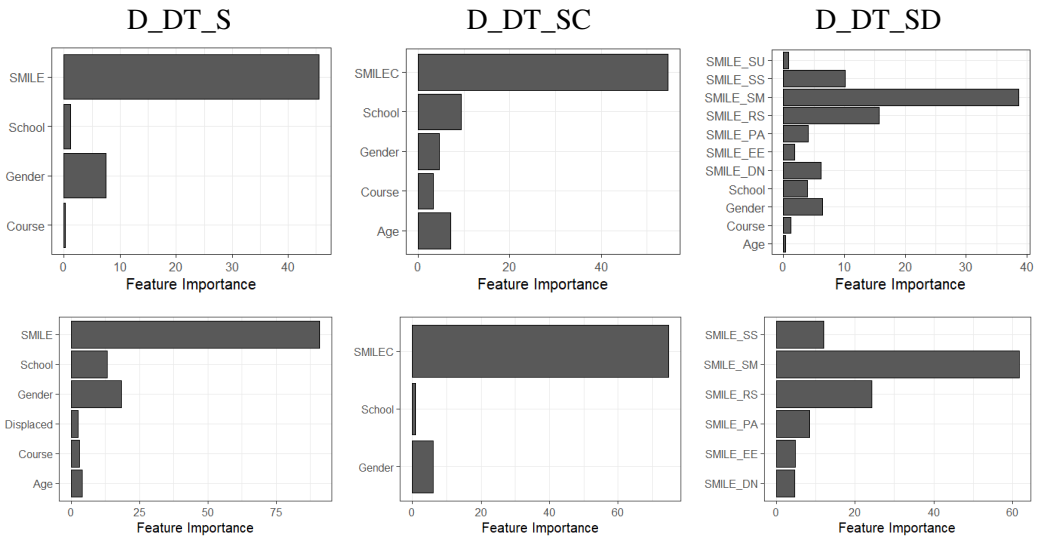

Figure 64: Features importance in each decision trees model for screening anxiety and depression, without (top) and with (bottom) oversampling

4.6.3.3 *Logistic regression for screening for anxiety and depression through multiple variables*

Finally, logistic regression (LR) was applied to screening for both anxiety and depression through multiple variables following the same three models as in Sections 4.6.1.3 and 4.6.2.3 for screen for anxiety. As before, the models are sequentially simplified by the application of the Wald test with a 5% significance level to comply with the principle of parsimony. The performance of the screening procedure associated with the LR models is presented in Table 12.

$$\begin{aligned}\widehat{\text{logit}\,(p_B)} \;=\;& 11.6391 - 0.0967\,\text{SMILE} - 1.1714\,G_{\text{Male}} + 0.6004\,G_{\text{Non-binary}} \\ & -0.0571\,\text{Age} + 0.3601\,C_{\text{Graduation}} + 0.6045\,C_{\text{Master}} \\ & +1.0288\,S_{\text{ESECS}} - 0.0816\,S_{\text{ESSLei}} + 0.4905\,S_{\text{ESTG}} + 1.2614\,S_{\text{ESTM}},\end{aligned}$$

where p_B is the probability of the student suffering from both anxiety and depression. The Hosmer and Lemeshow goodness of fit test (p-value = 0.7148) reveals no difference

between the observed and model-predicted values. The associated ROC curve is shown in Figure 65.

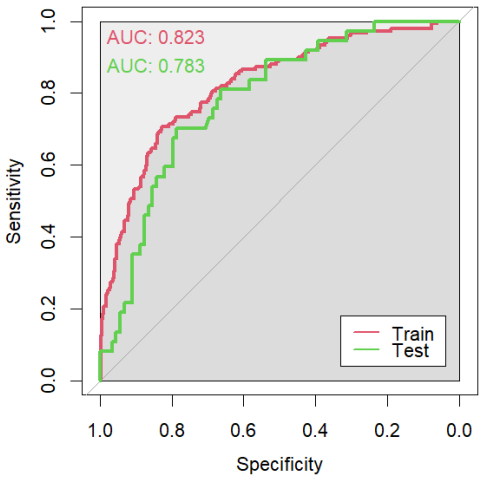

Figure 65: ROC curves of screening for anxiety and depression through SMILE and socio-demographic features (logistic regression)

The second model (AD_LR_SC) is based on the SMILE-C score, and its estimates are given by

$$\begin{aligned}\widehat{\text{logit}\,(p_B)} \;=\;& 11.7681 - 0.1507\,\text{SMILE} - \text{C} - 0.0622\,\text{Age}\\ & -1.0626\,G_{\text{Male}} + 0.6998\,G_{\text{Non-binary}}\\ & +0.4021\,C_{\text{Graduation}} + 0.5982\,C_{\text{Master}}\\ & +1.3140\,S_{\text{ESECS}} + 0.1778\,S_{\text{ESSLei}} + 0.7303\,S_{\text{ESTG}} + 1.2647\,S_{\text{ESTM}}.\end{aligned}$$

The Hosmer and Lemeshow test indicates a good fit with the training sample (p-value =0.4874). Figure 66 provides the ROC curve of the classification performed with model AD_LR_SC.

The estimates of the third model (AD_LR_SD), with the scores obtained in each domain of the SMILE survey, are given by

$$\begin{aligned}\widehat{\text{logit}\,(p_B)} \;=\;& 11.4320 - 0.0592\,\text{Age} - 0.1469\,\text{SMILE}_{\text{S}}\text{U} - 0.1818\,\text{SMILE}_{\text{S}}\text{M}\\ & -0.1520\,\text{SMILE}_{\text{R}}\text{S} - 0.1025\,\text{SMILE}_{\text{S}}\text{S}\\ & -1.2461\,G_{\text{Male}} + 0.5643\,G_{\text{Non-binary}}\\ & +0.279\,C_{\text{Graduation}} + 0.6615\,C_{\text{Master}}\\ & +1.0206\,S_{\text{ESECS}} + 0.0462\,S_{\text{ESSLei}} + 0.5289\,S_{\text{ESTG}} + 1.3050\,S_{\text{ESTM}},\end{aligned}$$

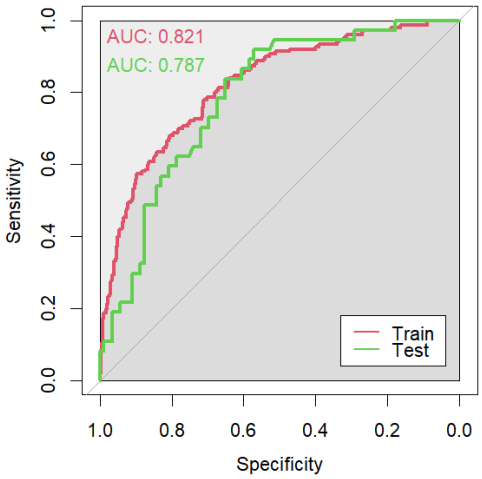

Figure 66: ROC curves of screening for anxiety and depression through SMILE-C and socio-demographic features (logistic regression)

with a p-value equal to 0.0616 in the Hosmer and Lemeshow goodness of fit test. Thus, it seems that the models model's estimates fit the data at an acceptable level. The ROC curve obtained through this model is shown in Figure 67.

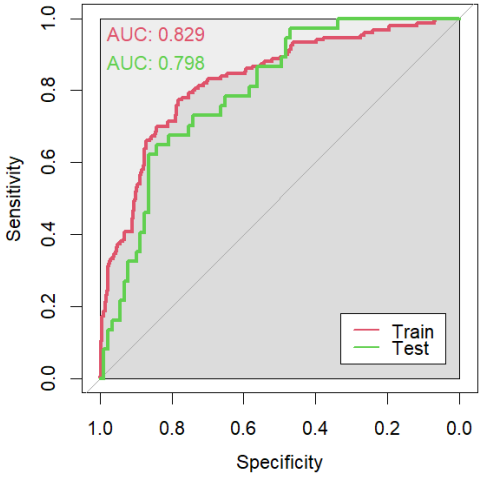

Figure 67: ROC curves of screening for anxiety and depression through SMILE domains and socio-demographic features (logistic regression)

As in previous cases, the three models are also estimated using the balanced training sample by oversampling. Hence, the estimates of the first model (AD_LR_S_O) are given by

$$\begin{aligned}\widehat{\text{logit}\,(p_B)} &= 13.2272 - 0.0988\,\text{SMILE} - 0.0736\,\text{Age}\\ &\quad -1.1488\,G_{\text{Male}} + 1.0489\,G_{\text{Non-binary}}\\ &\quad +0.4887\,C_{\text{Graduation}} + 0.7817\,C_{\text{Master}} + 0.9807\,S_{\text{ESECS}}\\ &\quad +0.0558\,S_{\text{ESSLei}} + 0.3500\,S_{\text{ESTG}} + 1.1919\,S_{\text{ESTM}},\end{aligned}$$

which attained a p-value of 0.1900 in the Hosmer and Lemeshow goodness of fit test. Therefore, it seems to indicate a good-fitting model. The estimated second model (AD_LR_SC_O) is given by

$$\begin{aligned}\widehat{\text{logit}\,(p_B)} &= 13.3920 - 0.1546\,\text{SMILE} - \text{C} - 0.0760\,\text{Age}\\ &\quad -1.0377\,G_{\text{Male}} + 1.4650\,G_{\text{Non-binary}}\\ &\quad +0.5220\,C_{\text{Graduation}} + 0.7343\,C_{\text{Master}} + 1.2266\,S_{\text{ESECS}}\\ &\quad +0.2368\,S_{\text{ESSLei}} + 0.5242\,S_{\text{ESTG}} + 1.1590\,S_{\text{ESTM}} + 0.4407\,\text{Displaced},\end{aligned}$$

with a p-value of 0.7090 in the Hosmer and Lemeshow goodness of fit test and, consequently, revealing that the model fits the data. At last, the third model for screening for anxiety and depression (AD_LR_SD_O) is given by

$$\begin{aligned}\widehat{\text{logit}\,(p_B)} &= 13.8162 - 0.0761\,\text{Age} - 0.0706\,\text{SMILE}_{\text{D}}\text{N} - 0.1542\,\text{SMILE}_{\text{S}}\text{U}\\ &\quad -0.1584\,\text{SMILE}_{\text{S}}\text{M} - 0.1087\,\text{SMILE}_{\text{R}}\text{S} - 0.1205\,\text{SMILE}_{\text{S}}\text{S}\\ &\quad -1.2952\,G_{\text{Male}} + 0.8994\,G_{\text{Non-binary}}\\ &\quad +0.4035\,C_{\text{Graduation}} + 0.6897\,C_{\text{Master}}\\ &\quad +0.9522\,S_{\text{ESECS}} + 0.1430\,S_{\text{ESSLei}} + 0.3471\,S_{\text{ESTG}} + 1.1771\,S_{\text{ESTM}},\end{aligned}$$

with a p-value of 0.7898 in the Hosmer and Lemeshow goodness of fit test. Therefore, the model seems to fit the data. Figure 68 shows the ROC curve of the training and testing sample in the three models using the balanced training sample. This graph reveals no significant difference between the two curves, and consequently, the null hypothesis is not rejected in DeLong's test for equality of two ROC curves (p-values equals 0.2907, 0.4714, and 0.4881, respectively). Nevertheless, there is a significant difference in the composition of the balanced training sample and the test sample, where the number of students with both anxiety and depression (29.4%) is quite lower than those without (70.6%).

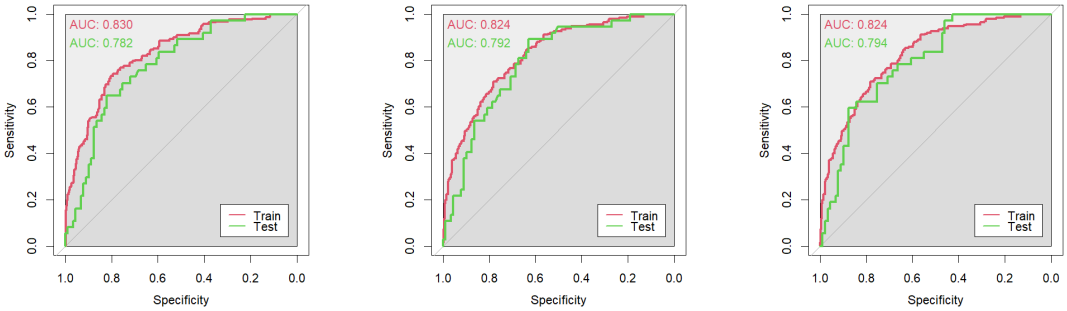

Figure 68: ROC curves from the train and test samples on screening for anxiety and depression of the three balanced multivariate models.

Figure 69 displays the ROC curve of the six estimated models, and Table 12 summarizes all models of screening both anxiety and depression.

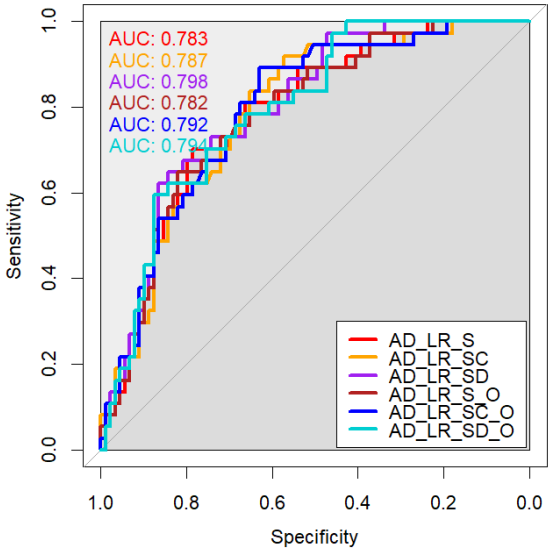

Figure 69: ROC curves of screening for anxiety and depression of the 6 multivariate models analyzed through logistic regression.

In general, logistic regression outperformed the decision tree methodology. In logistic regression models, as with other instances of its application, the use of oversampling techniques appears to have no effect on the model's quality (in fact, the results seem even worse). Ultimately, the performance of the three models is quite similar, with no clear standout model outperforming the others.

5

DISCUSSION

In this study, we present a linguistic and cultural adaptation of the SMILE and SMILE-C questionnaires for Portugal Portuguese from the Brazilian Portuguese version since significant differences between the two are present even if the language is the same. The target audience for the study was students of higher education at the Polytechnic of Leiria. Exploring the implications of the findings concerning the effectiveness of SMILE and SMILE-C scores in detecting anxiety and depression among students is crucial, given that their absence is deemed indicative of good well-being. Thus, logistic regression and decision tree models examined these scores along with their respective domains and sociodemographic variables.

Hence, apart from providing an overview of the overall well-being of IPLeiria students, the primary findings of this study revolve around assessing the internal consistency of the SMILE and SMILE-C questionnaires, along with their effectiveness in identifying anxiety and depression.

While the majority of questions exhibit strong internal consistency, certain domains and specific items have been pinpointed as detractors of this consistency. It is advisable to either eliminate or modify these questions to bolster internal consistency, particularly if it is intended to carry out a validation study of the survey. However, in terms of a complete survey, the values obtained in this study (between 0.8 and 0.86 in both α and KMO measures) are higher than those achieved in initial studies, which present values close to 0.75.

Moreover, the analysis revealed that the SMILE score stands out as the single most influential variable for screening purposes. Its consistently high AUC value in decision tree models emphasizes its strength. Additionally, the high AUC value of the SMILE-C score (even if it is not as high as the SMILE score) suggests its potential value in use for the same application since this is a smaller questionnaire that makes it ideal for maximizing the number of responses. While decision tree models incorporating multiple variables performed better than single-variable models, the improvement was relatively minor. Interestingly, oversampling within these models yielded mixed results. Still, there was a substantial improvement in sensitivity, an important metric to avoid missing in-risk individuals, suggesting the need for further exploration of oversampling techniques. Sociodemographic variables, on the other hand, had minimal impact on the overall performance of decision tree

models, especially when oversampling was applied. Logistic regression models emerged as the superior approach, achieving consistently higher AUC values compared to decision tree models. Oversampling in logistic regression models also had minimal impact, indicating model stability. Interestingly, SMILE-based models performed comparably to SMILE-C models in logistic regression, suggesting that both scoring methods have merit. Despite these promising findings, all models exhibited limitations in accurately classifying individuals. This highlights the need to incorporate additional variables for improved screening efficacy. Future research efforts should explore the inclusion of factors known to influence well-being, such as academic stress, employment, social support networks, family history, and other health-related factors, such as obesity, smoking, and alcoholism.

A recent study introduced the U-SMILE, a shorter version of the SMILE survey specifically designed to evaluate lifestyle among university students (De Boni et al., 2023). This study reveals acceptable internal consistency and evidence of convergent and concurrent validity, supporting its potential as a tool for assessing lifestyle in the intended population. Interestingly, the U-SMILE scores exhibited moderate correlations with mental health measures of anxiety and depression. The development of the U-SMILE underscores the complexity of measuring lifestyle. While the initial conceptual framework proposed seven distinct lifestyle domains, the final U-SMILE structure revealed some overlap between these domains, and the author recommends the use of the overall score. Those findings overlap our own. Furthermore, the development process of the questionnaire highlighted the importance of a better definition and metric for well-being and lifestyle. This need was also evident throughout the course of our research.

It is important to acknowledge that this study has certain limitations. Firstly, the sample population was restricted to students from a single higher education institution, which may restrict the generalizability of the findings to broader student populations. Moreover, the reliance on self-reported data collection methods could introduce biases or inaccuracies.

Given the considerations outlined, future research endeavors should strive to mitigate these limitations by:

- expanding the study population: alternative recruitment strategies, including students from various higher education institutions and backgrounds, can potentially enhance the generalizability of the results.
- incorporating objective measures: utilizing clinical interviews or biological markers alongside self-reported measures can increase the accuracy of diagnoses.
- exploring alternative modeling techniques: investigating the efficacy of machine learning algorithms with more variables can yield further improvements.

- conducting longitudinal studies: tracking participants over time can provide valuable insights into the predictive power of the identified variables for long-term mental health outcomes.

6

CONCLUSION

In this study, a web survey was administered to all students of IPLeiria, comprising socio-demographic inquiries (such as gender, age, school, course, and commuting habits), the SMILE questionnaire for assessing lifestyle and well-being, and the PHQ-2 and GAD-7 questions for screening depression and anxiety. The SMILE survey was previously adapted for the Portuguese population using the existing versions in English and Portuguese (from Brazil).

The main findings of this study can be summarized as follows. A significant prevalence of students with depression (39.7%) and anxiety (44.9%) issues is observed. While the SMILE questionnaire demonstrates high internal consistency, certain domains and questions warrant reconsideration. Consequently, in a revised iteration of the SMILE survey, it is recommended to eliminate or revise certain questions. Furthermore, for SMILE to effectively identify students with anxiety or depression problems, additional variables must be incorporated into the survey. Despite a strong association between SMILE responses and the presence of anxiety or depression problems, the reliability of this classification is insufficient for practical application. As it stands, a low SMILE score merely suggests the possibility of anxiety and depression problems, indicating a need for enhanced assessment through supplementary information. Nonetheless, lower scores in SMILE questionnaire clearly suggest unhealthier lifestyles. Students in these conditions can be advised to seek support from counseling services or other resources to improve their well-being.

Future research efforts should explore factors known to influence well-being, such as academic stress, social support networks, family history, and health-related aspects, like obesity and chronic illnesses. Furthermore, integrating new questions and objective measures may enhance anxiety and depression diagnostic accuracy. To improve the generalization of the survey, it should be applied to other higher education institutions and, if possible, in longitudinal studies, in order to allow an assessment of evolution.

In conclusion, the SMILE survey holds promise as a tool for assessing the quality of life among higher education students and detecting anxiety and depression issues. However, to realize its full potential, enhancements are necessary. This includes refining the survey through the removal or modification of certain questions, as well as the inclusion of new ones to gather supplementary information.

BIBLIOGRAPHY

Balanzá-Martínez, Vicent et al. (2021). "The assessment of lifestyle changes during the COVID-19 pandemic using a multidimensional scale". In: *Revista de Psiquiatría y Salud Mental* 14.1, pp. 16–26. ISSN: 1888-9891. DOI: 10.1016/j.rpsm.2020.07.003.

Bártolo, Ana, Sara Monteiro, and Anabela Pereira (2017). "Factor structure and construct validity of the Generalized Anxiety Disorder 7-item (GAD-7) among Portuguese college students". In: *Cadernos de saude publica* 33, e00212716.

Breiman, Leo (2017). *Classification and regression trees*. Routledge. DOI: 10.1201/9781315139470.

Cervera-Martínez, J. et al. (2021). "Lifestyle changes and mental health during the COVID-19 pandemic: A repeated, cross-sectional web survey". In: *Journal of Affective disorders* 295, pp. 173–182. DOI: 10.1016/j.jad.2021.08.020.

Chang, Jun-Jie et al. (2021). "Prevalence of anxiety symptom and depressive symptom among college students during COVID-19 pandemic: A meta-analysis". In: *Journal of Affective Disorders* 292, pp. 242–254. ISSN: 0165-0327. DOI: 10.1016/j.jad.2021.05.109.

Cordeiro Prata, Helena Cristina (2022). "Validação do Patient Health Questionnaire-9 (PHQ-9) para a população portuguesa". In: *PCLI - Dissertações de Mestrado*. URL: http://hdl.handle.net/10400.12/9067.

De Boni, R. B. et al. (2023). "U-SMILE: a brief version of the Short Multidimensional Inventory on Lifestyle Evaluation". In: *Trends in psychiatry and psychotherapy*. URL: https://doi.org/10.47626/2237-6089-2023-0722.

Hastie, Trevor, Robert Tibshirani, and Jerome Friedman (2009). *The elements of statistical learning: data mining, inference, and prediction*. Springer. DOI: 10.1007/978-0-387-84858-7.

Havrylyuk, Bohdan (2020). "Em Portugal, de quarentena:Impactos sobre o bam-estar e a saúde nos estudantes universitários". Master dissertion. Instituto Superior de Psicologia Aplicada.

Hernández-Torrano, Daniel et al. (2020). "Mental Health and Well-Being of University Students: A Bibliometric Mapping of the Literature". In: *Frontiers in Psychology* 11. ISSN: 1664-1078. DOI: 10.3389/fpsyg.2020.01226.

James, Gareth et al. (2021). *An Introduction to Statistical Learning with Applications in R*. Springer. DOI: 10.1007/978-1-0716-1418-1.

Kroenke, Kurt, Robert L. Spitzer, and Janet B. W. Williams (Nov. 2003). "The Patient Health Questionnaire-2". In: *Medical Care* 41.11, pp. 1284–1292. DOI: 10.1097/01.mlr.0000093487.78664.3c.

Lopes, João (Jan. 2015). "Bem-estar psicológico em estudantes do ensino superior: relação com as variáveis sociodemográficas, pessoais e académicas". Available at http://hdl.handle.net/10174/14604. PhD thesis. Évora: University of Évora.

R Core Team (2023). *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing. URL: https://www.R-project.org/.

Ryff, C. D. and C.L. Keyes (1995). "The structure of psychological well-being revisited". In: *Journal of Personality and Social Psychology* 69 (4), pp. 719–727. DOI: https://doi.org/10.1037/0022-3514.69.4.719.

Santos, R. et al. (2019). "Accuracy Measures for Binary Classification Based on a Quantitative Variable". In: *REVSTAT – Statistical Journal* 17.2, pp. 223–244. DOI: 10.57805/revstat.v17i2.266.

Silva, Armando Manuel Marques, Irma da Silva Brito, and João Manuel da Costa Amado (June 2014). "Tradução, adaptação e validação do questionário *Fantastic Lifestyle Assessment* em estudantes do ensino superior". In: *Ciência & Saúde Coletiva* 19.6, pp. 1901–1909. ISSN: 1413-8123. DOI: 10.1590/1413-81232014196.04822013.

Spitzer, Robert L. et al. (May 2006). "A Brief Measure for Assessing Generalized Anxiety Disorder". In: *Archives of Internal Medicine* 166.10, p. 1092. DOI: 10.1001/archinte.166.10.1092.

Tabachnick, Barbara G. and Linda S. Fidell (2021). *Using Multivariate Statistics*. Pearson. ISBN: 9780137526543.

Wilson, Douglas MC, Eleanor Nielsen, and Donna Ciliska (1984). "Lifestyle assessment". In: *Canadian Family Physician* 30, pp. 1527–1523.

Zelterman, Daniel (2015). *Applied Multivariate Statistics with R*. Springer. DOI: 10.1007/978-3-319-14093-3.

APPENDIX

A

SURVEYS

In this appendix, the complete survey conducted with students of the Polytechnic of Leiria is presented. Since the survey was originally available in Portuguese, it is presented first in English and then in Portuguese.

This presentation includes the initial introduction of the survey as well as the consent for the completion of the questionnaire. Subsequently, the first questions aims to characterize the students, including age, gender, school, type of course, and displacement. Following this, the 43 questions of the SMILE survey are asked (Q1 to Q43), divided into its seven domains. As the SMILE-C survey does not include all SMILE questions, the questions included are identified with [C]. Finally, the two questions from the Patient Health Questionnaire survey (identified with [PHQ-2]) and the seven questions from the Generalized Anxiety Disorder survey (identified with [GAD-7]) are presented.

A.1 ENGLISH VERSION

Lifestyles and Well-being

A.1.1 *Presentation*

We would like to invite you to participate in a research study on lifestyles and well-being of students at the Polytechnic of Leiria by anonymously responding to the following questionnaire.

This research is part of a project by student Daniel Santos, within the scope of the Master's in Data Science, supervised by Professors Rui Santos and Susana Ferreira from the School of Technology and Management of the Polytechnic of Leiria.

The data collected will be analyzed, ensuring anonymity and confidentiality, and will be used exclusively for research purposes within the mentioned project. The study's findings will be presented to the Polytechnic of Leiria.

Your contribution is of utmost importance, and responding to the questionnaire will not take more than 10 minutes.

If you have any doubts or questions, please do not hesitate to contact us via email at: rui.santos@ipleiria.pt.

We appreciate your contribution!

A.1.2 *Consent for the completion of the questionnaire*

1 The questions address aspects related to the perceptions that students have regarding their lifestyles, quality of life, and well-being. There are no right or wrong answers for each question. What matters is that your answers reflect what you think, feel, or do regarding each item. Please answer all items.

   - I agree to participate in this study and I am a student at the Polytechnic of Leiria.
   - I do not agree to participate in this study.
   - I do not agree to participate in this study.

A.1.3 *Student Characterization*

2 Gender
- Female
- Male
- Non-binary

3 Age

4 Program at the Polytechnic of Leiria
- Technical Professional Higher Course (TeSP)
- Bachelor's Degree
- Master's Degree
- Other:

5 School
- ESECS - School of Education and Social Sciences
- ESTG - School of Technology and Management
- ESAD.CR - School of Arts and Design
- ESTM - School of Tourism and Maritime Technology
- ESSLei - School of Health Sciences

6 Are you currently living away from your home address (family address)?
- Living away
- Not living away

A.1.4 *SMILE – Diet and Nutrition [DN]*

7 In the last month, how often in your daily routine···

Q1 Do you eat meals you or someone else in your family prepares?
* Never
* Seldom
* Often
* Always

Q2 When shopping for food, do you check labels for ingredients such as quantity of salt?
* Never
* Seldom
* Often
* Always

Q3 Do you eat processed food (frozen food such as pizza, French fries, puff pastries, deep-fried foods and canned foods)? [C]

* Never
* Seldom
* Often
* Always

Q4 Do you eat fast-food, high-calorie sweet or fatty foods when you are stressed or sad? [C]

* Never
* Seldom
* Often
* Always

Q5 Do you eat healthy foods such as fresh fruits, fresh vegetables, wholegrain, legumes or nuts? [C]

* Never
* Seldom
* Often
* Always

Q6 Do you keep a regular meal schedule? [C]

* Never
* Seldom
* Often
* Always

Q7 Do you share your main meals with friends or family? [C]

* Never
* Seldom
* Often
* Always

A.1.5 *SMILE – Substance Use [SU]*

- 8] In the last month, how often in your daily routine···

Q8 Do you drink 5 or more doses (men) or 4 or more doses (women) of alcoholic beverages on a single occasion, which means within 2 hours?[1] [C]

* Never

1 1 dose of alcohol=1 glass of beer OR 1 glass of wine OR 1 shot of spirit (such as rum, vodka, whisky, tequila or gin).

- Seldom
- Often
- Always

Q9 Do you smoke tobacco (cigarette, electronic cigarette, cigar, pipe, smokeless tobacco)? [C]

- Never
- Seldom
- Often
- Always

Q10 Do you use marijuana or hashish? [C]

- Never
- Seldom
- Often
- Always

Q11 Do you use other drugs (cocaine, crack, amphetamines, ecstasy, opioids without medical prescription, and others)? [C]

- Never
- Seldom
- Often
- Always

A.1.6 *SMILE – Physical activity [PA]*

9 In the last month, how often in your daily routine· · ·

Q12 Do you exercise for at least 30 minutes daily (or 150 minutes a week)? [C]

- Never
- Seldom
- Often
- Always

Q13 Do you play at least 2 hours of team sports (like soccer, volleyball, basketball, rugby, etc.) a week?

- Never
- Seldom
- Often
- Always

Q14 Do you choose to climb stairs instead of using an elevator and/or walking to perform your daily routines instead of using a car/public transportation?

* Never
* Seldom
* Often
* Always

Q15 Do you feel good after performing physical activity?

* Never
* Seldom
* Often
* Always

A.1.7 *SMILE – Stress management [SM]*

10 In the last month, how often in your daily routine···

Q16 Do you make time to relax? [C]

* Never
* Seldom
* Often
* Always

Q17 Do you use any strategy or psychological support to deal with stress (for instance meditation, mindfulness or psychotherapy)? [C]

* Never
* Seldom
* Often
* Always

Q18 Do you use physical strategies to deal with stress (for instance yoga, tai-chi, exercise)? [C]

* Never
* Seldom
* Often
* Always

Q19 Do you practice a faith or religion? [C]

* Never
* Seldom
* Often

- Always

Q20 Do you feel that you have a good work-life balance?

- Never
- Seldom
- Often
- Always

Q21 Do you feel that your work / chores are never done?

- Never
- Seldom
- Often
- Always

Q22 Are you satisfied with the time it takes you to commute to work?

- Never
- Seldom
- Often
- Always

Q23 Do you feel that your life has a meaning? [C]

- Never
- Seldom
- Often
- Always

Q24 Do you feel grateful for the life you have? [C]

- Never
- Seldom
- Often
- Always

A.1.8 *SMILE – Restorative sleep [RS]*

11 In the last month, how often in your daily routine· · ·

Q25 Do you manage to sleep between 7 and 9 hours per night? [C]

- Never
- Seldom
- Often
- Always

Q26 Do you feel rested with the number of hours you sleep? [C]

- Never
- Seldom
- Often
- Always

Q27 Do you usually rest (sleep or take a nap) after lunch?

- Never
- Seldom
- Often
- Always

Q28 Do you maintain a regular sleep schedule? [C]

- Never
- Seldom
- Often
- Always

Q29 Do you use sleeping pills? [C]

- Never
- Seldom
- Often
- Always

A.1.9 *SMILE – Social support [SS]*

12 In the last month, how often in your daily routine· · ·

Q30 Do you interact with your friends and/or relatives? [C]

- Never
- Seldom
- Often
- Always

Q31 Do you feel that you are part of a group of friends, the community or the society? [C]

- Never
- Seldom
- Often
- Always

Q32 Do you have someone you trust who listens to your problems or concerns? [C]

- Never
- Seldom
- Often
- Always

Q33 Do you have someone to help with everyday chores (for instance cooking, housekeeping, shopping)? [C]

- Never
- Seldom
- Often
- Always

Q34 Do you have someone in your life to go out or have fun with when you fell like it?

- Never
- Seldom
- Often
- Always

Q35 Do you take part in celebrations/ reunions with family/ friends/colleagues?

- Never
- Seldom
- Often
- Always

Q36 Do you enjoy your leisure time? [C]

- Never
- Seldom
- Often
- Always

Q37 Do you make yourself available to support your significant ones? [C]

- Never
- Seldom
- Often
- Always

Q38 Are you satisfied with your sexual life?

- Never
- Seldom
- Often

- Always

Q39 Do you feel loved?

- Never
- Seldom
- Often
- Always

A.1.10 *SMILE – Environment exposures (screen time/ outdoor time) [EE]*

13 In the last month, how often in your daily routine· · ·

Q40 Do you spend more than 2 hours a day watching TV, playing computer games, video games or in the internet?

- Never
- Seldom
- Often
- Always

Q41 Do you spend time on a computer / smartphone within one hour of going to sleep?

- Never
- Seldom
- Often
- Always

Q42 Are you in touch with nature (for instance parks, beach, countryside, mountains)? [C]

- Never
- Seldom
- Often
- Always

Q43 Do you feel your relationship to nature, that is all living things, is an important part of who you are?

- Never
- Seldom
- Often
- Always

A.1.11 *Health Questionnaire*

14 Over the last 2 weeks, how often have you been bothered by any of the following problems?

- Little interest or pleasure in doing things [PHQ-2]
  - Not at all
  - Several days
  - More than half the days
  - Nearly every day
- Feeling down, depressed, or hopeless [PHQ-2]
  - Not at all
  - Several days
  - More than half the days
  - Nearly every day
- Feeling nervous, anxious or on edge [GAD-7]
  - Not at all
  - Several days
  - More than half the days
  - Nearly every day
- Not being able to stop or control worrying [GAD-7]
  - Not at all
  - Several days
  - More than half the days
  - Nearly every day
- Worrying too much about different things [GAD-7]
  - Not at all
  - Several days
  - More than half the days
  - Nearly every day
- Trouble relaxing [GAD-7]
  - Not at all
  - Several days
  - More than half the days
  - Nearly every day
- Being so restless that it is hard to sit still [GAD-7]

    - Not at all
    - Several days
    - More than half the days
    - Nearly every day
- Becoming easily annoyed or irritable [GAD-7]
    - Not at all
    - Several days
    - More than half the days
    - Nearly every day
- Feeling afraid as if something awful might happen [GAD-7]
    - Not at all
    - Several days
    - More than half the days
    - Nearly every day

A.2 PORTUGUESE VERSION

Estilos de Vida e Bem-Estar

A.2.1 *Apresentação*

Gostaríamos de o/a convidar a participar numa investigação sobre estilos de vida e bem-estar dos estudantes do Politécnico de Leiria, respondendo, de forma anónima, ao questionário que se segue.

Esta investigação insere-se num projeto do estudante Daniel Santos, no âmbito do Mestrado em Ciência de Dados, a decorrer sob a orientação dos Professores Rui Santos e Susana Ferreira, da Escola Superior de Tecnologia e Gestão do Politécnico de Leiria.

Os dados recolhidos serão analisados, garantindo o anonimato e confidencialidade, sendo utilizados exclusivamente para fins de investigação no âmbito do referido projeto. As conclusões do estudo serão apresentadas ao Politécnico de Leiria.

O seu contributo é da maior importância, sendo que a resposta ao questionário não tomará mais que 10 minutos.

Em caso de dúvidas ou questões, não hesite em contactar-nos por email através do endereço: rui.santos@ipleiria.pt.

Agradecemos o seu contributo!

A.2.2 *Consentimento para a realização do questionário.*

1 As questões colocadas abordam aspetos relativos às perceções que os/as estudantes têm em relação aos seus estilos de vida, qualidade de vida e bem-estar. Para cada questão não há respostas certas ou erradas. O importante é que as suas respostas expressem o que pensa, sente ou faz, em relação a cada um dos itens. Por favor, responda todos os itens.

   - Aceito participar neste estudo e sou estudante do Politécnico de Leiria.
   - Não aceito participar neste estudo.
   - Não aceito participar neste estudo.

A.2.3 *Caracterização do estudante*

2 Género
   - Feminino
   - Masculino
   - Não binário

3 Idade

4 Frequenta no Politécnico de Leiria
   - Curso Técnico Superior Profissional (TeSP)
   - Licenciatura
   - Mestrado
   - Outro:

5 Escola
   - ESECS - Escola Superior de Educação e Ciências Sociais
   - ESTG - Escola Superior de Tecnologia e Gestão
   - ESAD. CR - Escola Superior de Artes e Design
   - ESTM - Escola Superior de Turismo e Tecnologia do Mar
   - ESSLei - Escola Superior de Saúde

6 Encontra-se deslocado do seu local de residência (morada de família)?
   - Deslocado
   - Não deslocado

A.2.4 *SMILE – Dieta e Nutrição*

7 No último mês, com que frequência na sua rotina diária ···
   - Comeu refeições caseiras, confecionadas por si ou por um familiar?
     * Nunca
     * Raramente
     * Frequentemente
     * Sempre
   - Ao comprar alimentos, consultou o rótulo para verificar os ingredientes, como a quantidade de sal?
     * Nunca
     * Raramente
     * Frequentemente
     * Sempre

  - Comeu alimentos processados (ultracongelados como pizza, fritos ou conservas)? [C]
    * Nunca
    * Raramente
    * Frequentemente
    * Sempre
  - Comeu fast food, alimentos ricos em gordura ou açúcar quando se sentiu stressado/a ou triste? [C]
    * Nunca
    * Raramente
    * Frequentemente
    * Sempre
  - Comeu alimentos saudáveis tais como fruta e legumes, cereais integrais, leguminosas ou frutos secos? [C]
    * Nunca
    * Raramente
    * Frequentemente
    * Sempre
  - Manteve uma regularidade em relação ao horário das refeições? [C]
    * Nunca
    * Raramente
    * Frequentemente
    * Sempre
  - Realizou as suas refeições principais com amigos ou familiares? [C]
    * Nunca
    * Raramente
    * Frequentemente
    * Sempre

A.2.5 *SMILE – Consumo de substâncias*

- 8] No último mês, com que frequência na sua rotina diária···
  - Bebeu 5 ou mais doses (homens) ou 4 ou mais doses (mulheres) de bebidas alcoólicas numa única ocasião (ou seja, dentro de um intervalo de duas horas)?[2] [C]

2 *1 dose de álcool = 1 copo de cerveja OU 1 copo de vinho OU 1 dose de uma bebida branca (como rum, vodka, whisky, tequila ou gin).

    - Nunca
    - Raramente
    - Frequentemente
    - Sempre
  - Fumou tabaco (cigarro, cigarro electrónico, charuto, cachimbo)? [C]
    - Nunca
    - Raramente
    - Frequentemente
    - Sempre
  - Consumiu erva ou haxixe? [C]
    - Nunca
    - Raramente
    - Frequentemente
    - Sempre
  - Consumiu outras drogas ilícitas (cocaína, crack, anfetaminas, ecstasy, opióides, entre outras)? [C]
    - Nunca
    - Raramente
    - Frequentemente
    - Sempre

A.2.6 *SMILE – Atividade física*

9 No último mês, com que frequência na sua rotina diária· · ·

  - Faz exercício físico pelo menos 30 minutos por dia (ou 150 minutos por semana)? [C]
    - Nunca
    - Raramente
    - Frequentemente
    - Sempre
  - Praticou desportos de equipa (por exemplo futebol, voleibol, basquetebol, rugby, etc.) pelo menos 2 horas por semana?
    - Nunca
    - Raramente
    - Frequentemente
    - Sempre

- Optou por subir escadas em vez de utilizar o elevador e/ou caminhou para realizar as suas rotinas diárias em vez de utilizar uma viatura/transporte público?
  - Nunca
  - Raramente
  - Frequentemente
  - Sempre
- Sentiu-se bem depois de praticar exercício físico?
  - Nunca
  - Raramente
  - Frequentemente
  - Sempre

A.2.7 *SMILE – Gestão de stress*

10 No último mês, com que frequência na sua rotina diária···

- Conseguiu dedicar algum tempo para relaxar? [C]
  - Nunca
  - Raramente
  - Frequentemente
  - Sempre
- Utilizou alguma estratégia ou apoio psicológico para lidar com o stress (por exemplo meditação, mindfulness ou psicoterapia)? [C]
  - Nunca
  - Raramente
  - Frequentemente
  - Sempre
- Recorreu a estratégias físicas para lidar com o stress (por exemplo yoga, tai-chi, exercício físico)? [C]
  - Nunca
  - Raramente
  - Frequentemente
  - Sempre
- Praticou alguma religião ou crença? [C]
  - Nunca
  - Raramente
  - Frequentemente

    - Sempre
- Sente que tem um bom equilibro entre a sua vida profissional e a sua vida pessoal?
    - Nunca
    - Raramente
    - Frequentemente
    - Sempre
- Sente que o seu trabalho ou as suas tarefas nunca estão concluídas?
    - Nunca
    - Raramente
    - Frequentemente
    - Sempre
- Está satisfeito/a com o tempo que demorou a deslocar-se para o trabalho?
    - Nunca
    - Raramente
    - Frequentemente
    - Sempre
- Sente que a sua vida tem significado? [C]
    - Nunca
    - Raramente
    - Frequentemente
    - Sempre
- Sente-se grato pela vida que tem? [C]
    - Nunca
    - Raramente
    - Frequentemente
    - Sempre

A.2.8 *SMILE – Sono reparador*

11 No último mês, com que frequência na sua rotina diária···

- Dormiu entre 7 a 9 horas por dia? [C]
    - Nunca
    - Raramente
    - Frequentemente
    - Sempre

- Sentiu-se descansado/a com o número de horas que dormiu? [C]
  - Nunca
  - Raramente
  - Frequentemente
  - Sempre
- Descansou (por exemplo fazer uma sesta) depois do almoço?
  - Nunca
  - Raramente
  - Frequentemente
  - Sempre
- Manteve um horário de sono regular? [C]
  - Nunca
  - Raramente
  - Frequentemente
  - Sempre
- Tomou medicamentos para dormir? [C]
  - Nunca
  - Raramente
  - Frequentemente
  - Sempre

A.2.9 *SMILE – Apoio social*

12 No último mês, com que frequência na sua rotina diária···

- Interagiu com os seus amigos e/ou familiares? [C]
  - Nunca
  - Raramente
  - Frequentemente
  - Sempre
- Sentiu que pertence a um grupo de amigos, comunidade ou sociedade? [C]
  - Nunca
  - Raramente
  - Frequentemente
  - Sempre
- Sentiu que tem alguém de confiança que ouve os seus problemas ou preocupações? [C]

    - Nunca
    - Raramente
    - Frequentemente
    - Sempre
- Teve alguém que o/a ajudou nas tarefas diárias (por exemplo cozinhar, cuidar da casa, fazer compras)? [C]
    - Nunca
    - Raramente
    - Frequentemente
    - Sempre
- Teve alguém que o/a acompanhou para sair ou para se divertir quando necessitou?
    - Nunca
    - Raramente
    - Frequentemente
    - Sempre
- Participou em festas/reuniões com familiares/amigos/colegas?
    - Nunca
    - Raramente
    - Frequentemente
    - Sempre
- Gostou do seu tempo de lazer? [C]
    - Nunca
    - Raramente
    - Frequentemente
    - Sempre
- Esteve disponível para apoiar as pessoas que lhe são próximas? [C]
    - Nunca
    - Raramente
    - Frequentemente
    - Sempre
- Sentiu-se satisfeito/a com a sua vida sexual?
    - Nunca
    - Raramente
    - Frequentemente
    - Sempre

- Sentiu-se amado/a?
  - Nunca
  - Raramente
  - Frequentemente
  - Sempre

A.2.10 *SMILE – Exposições ambientais (tempo de ecrã/tempo ao ar livre)*

13 No último mês, com que frequência na sua rotina diária···

- Permaneceu mais de 2 horas por dia a ver televisão, a jogar em consolas ou computadores ou na internet?
  - Nunca
  - Raramente
  - Frequentemente
  - Sempre
- Utilizou o computador ou telemóvel na hora imediatamente anterior a ir dormir?
  - Nunca
  - Raramente
  - Frequentemente
  - Sempre
- Esteve em contacto com a natureza (parques, praia, campo, montanhas)? [C]
  - Nunca
  - Raramente
  - Frequentemente
  - Sempre
- Sentiu que a sua relação com a natureza é uma parte importante de quem é?
  - Nunca
  - Raramente
  - Frequentemente
  - Sempre

A.2.11 *Questionário sobre Saúde*

14 Durante os últimos 14 dias, em quantos foi afectado/a por algum dos seguintes problemas?

- Tive pouco interesse ou prazer em fazer coisas [PHQ-2]

    - Nunca
    - Em vários dias
    - Em mais de metade do número de dias
    - Em quase todos os dias
- Senti desânimo, desalento ou falta de esperança [PHQ-2]
    - Nunca
    - Em vários dias
    - Em mais de metade do número de dias
    - Em quase todos os dias
- Senti-me nervoso/a, ansioso/a ou irritado/a [GAD-7]
    - Nunca
    - Em vários dias
    - Em mais de metade do número de dias
    - Em quase todos os dias
- Fui incapaz de parar de me preocupar ou de controlar as preocupações [GAD-7]
    - Nunca
    - Em vários dias
    - Em mais de metade do número de dias
    - Em quase todos os dias
- Preocupei-me demais com diferentes assuntos [GAD-7]
    - Nunca
    - Em vários dias
    - Em mais de metade do número de dias
    - Em quase todos os dias
- Tive dificuldade em relaxar [GAD-7]
    - Nunca
    - Em vários dias
    - Em mais de metade do número de dias
    - Em quase todos os dias
- Estive tão inquieto/a que era difícil ficar sossegado/a [GAD-7]
    - Nunca
    - Em vários dias
    - Em mais de metade do número de dias
    - Em quase todos os dias
- Estive facilmente incomodado/a ou irritável [GAD-7]

    - Nunca
    - Em vários dias
    - Em mais de metade do número de dias
    - Em quase todos os dias
- Senti receio, como se algo terrível pudesse acontecer [GAD-7]
    - Nunca
    - Em vários dias
    - Em mais de metade do número de dias
    - Em quase todos os dias

B

R SCRIPT FOR THE STATISTICAL ANALYSIS

```
############################
# Load packages
library(car)
library(caret)
library(cvms)
library(devtools)
library(dplyr)
library(flextable)
library(ggcorrplot)
library(ggplot2)
library(here)
library(knitr)
library(lattice)
library(likert)
library(MASS)
library(mlr)
library(nnet)
library(nortest)
library(partykit)
library(pROC)
library(psych)
library(rattle)
library(readxl)
library(ResourceSelection)
library(rpart)
library(rpart.plot)
library(stats)
library(tibble)
library(tidyverse)
library(UBL)
library(viridis)
###############
# Read dataset
df <- read_excel("df.xlsx", sheet = "Sheet1")
############################
# Function to display the confusion matrix
Plot_confusion_matrix <- function(CM) {
  layout(matrix(c(1,1,2)))
  par(mar=c(2,2,2,2))
  plot(c(100, 345), c(300, 450), type = "n", xlab="",
```

```
    ylab="", xaxt='n', yaxt='n')
  title('Confusion matrix', cex.main=2)
  # create the matrix
  rect(150, 430, 240, 370, col='green4')
  text(195, 435, 'Negative', cex=1.2)
  rect(250, 430, 340, 370, col='red3')
  text(295, 435, CM$positive, cex=1.2)
  text(125, 370, 'Predicted', cex=1.3, srt=90, font=2)
  text(245, 450, 'Actual', cex=1.3, font=2)
  rect(150, 305, 240, 365, col='red3')
  rect(250, 305, 340, 365, col='green4')
  text(140, 400, 'Negative', cex=1.2, srt=90)
  text(140, 335, CM$positive, cex=1.2, srt=90)
  res <- as.numeric(CM$table)
  text(195, 400, res[1], cex=1.6, font=2, col='white')
  text(195, 335, res[2], cex=1.6, font=2, col='white')
  text(295, 400, res[3], cex=1.6, font=2, col='white')
  text(295, 335, res[4], cex=1.6, font=2, col='white')
  plot(c(100, 0), c(100, 0), type = "n", xlab="",
    ylab="", main = "Measures", xaxt='n', yaxt='n')
  text(10, 85, names(CM$byClass[1]), cex=1.2, font=2)
  text(10, 70, round(as.numeric(CM$byClass[1]), 3), cex=1.2)
  text(30, 85, names(CM$byClass[2]), cex=1.2, font=2)
  text(30, 70, round(as.numeric(CM$byClass[2]), 3), cex=1.2)
  text(50, 85, names(CM$byClass[5]), cex=1.2, font=2)
  text(50, 70, round(as.numeric(CM$byClass[5]), 3), cex=1.2)
  text(70, 85, names(CM$byClass[6]), cex=1.2, font=2)
  text(70, 70, round(as.numeric(CM$byClass[6]), 3), cex=1.2)
  text(90, 85, names(CM$byClass[7]), cex=1.2, font=2)
  text(90, 70, round(as.numeric(CM$byClass[7]), 3), cex=1.2)
  text(30, 35, names(CM$overall[1]), cex=1.5, font=2)
  text(30, 20, round(as.numeric(CM$overall[1]), 3), cex=1.4)
  text(70, 35, names(CM$overall[2]), cex=1.5, font=2)
  text(70, 20, round(as.numeric(CM$overall[2]), 3), cex=1.4)
}

############################
# Data preprocessing
############################

table(df[,6])
table(df[,9])

df <- data.frame(df[df[,6] == names(table(df[,6]))[1] &
    as.matrix(df[,9]) %in% names(table(df[,9]))[1:3],-(1:6)])
dim(df)

# clean column names
colnames(df)[1] <- "Gender"
```

```
colnames(df)[2] <- "Age"
colnames(df)[3] <- "Course"
colnames(df)[4] <- "School"
colnames(df)[5] <- "Displaced"

TipoQ <- c(rep("DN",7), rep("SU", 4), rep("PA", 4), rep("SM", 9),
    rep("RS", 5), rep("SS", 10), rep("EE", 4),
    rep("PHQ", 2), rep("GAD", 7))
colnames(df)[6:ncol(df)] <- paste0("Q", str_pad(1:52, 2, "left", "0"),
    ": ", TipoQ)

df$Gender <- factor(df$Gender, labels = c("Female", "Male",
    "Non-binary"))
df$Age <- as.numeric(df$Age)
df$Course <- ordered(df$Course, labels = c("TeSP", "Graduation",
    "Master"))
df$School <- factor(df$School, labels = c("ESAD.CR", "ESECS",
    "ESSLei", "ESTG", "ESTM"))
df$Displaced <- factor(df$Displaced, labels = c("Displaced",
    "Not displaced"))

PQ <- c(1:2, 5:7, 12:20, 22:28, 30:39, 42:43)
for (j in PQ+5){df[,j] <- ordered(df[,j], levels=c("Nunca",
    "Raramente", "Frequentemente", "Sempre"),
    labels=c("Never", "Seldom",
        "Often", "Always"))}
NQ <- c(3:4, 8:11, 21, 29, 40:41)
for (j in NQ+5){df[,j] <- ordered(df[,j],
    levels=c("Sempre", "Frequentemente", "Raramente", "Nunca"),
    labels=c("Always", "Often", "Seldom", "Never"))}

for (j in 49:57){df[,j] <- ordered(df[,j],
    levels=c("Nunca", "Em vários dias",
        "Em mais de metade do número de dias",
        "Em quase todos os dias"),
    labels=c("Not at all", "Several days",
        "More than half the days", "Nearly every day"))}

# Indexs
IDN <- 6:12
ISU <- 13:16
IPA <- 17:20
ISM <- 21:29
IRS <- 30:34
ISS <- 35:44
IEE <- 45:48
ISMILE <- 6:48
ISMILEC <- c(8:17,21:24, 28:31, 33:38,41:42,46)
IPHQ <- 49:50
```

```
IGAD <- 51:57

# Scores: SMILE, PHQ and GAD
for (j in 1:dim(df)[1]){
  df$SMILE[j] <- sum(as.numeric(df[j,ISMILE]))
  df$SMILEC[j] <- sum(as.numeric(df[j,ISMILEC]))
  df$PHQ[j] <- sum(as.numeric(df[j,IPHQ]))-2
  df$GAD[j] <- sum(as.numeric(df[j,IGAD]))-7
  df$SMILE_DN[j]  <- sum(as.numeric(df[j,IDN]))
  df$SMILE_SU[j]  <- sum(as.numeric(df[j,ISU]))
  df$SMILE_PA[j]  <- sum(as.numeric(df[j,IPA]))
  df$SMILE_SM[j]  <- sum(as.numeric(df[j,ISM]))
  df$SMILE_RS[j]  <- sum(as.numeric(df[j,IRS]))
  df$SMILE_SS[j]  <- sum(as.numeric(df[j,ISS]))
  df$SMILE_EE[j]  <- sum(as.numeric(df[j,IEE]))
  }

# Smile divided in 4 bins at about 25%
df$SMILE_Classes <- cut(df$SMILE,
                        breaks=c(0,quantile(df$SMILE, 0.25)[[1]],
                        quantile(df$SMILE, 0.50)[[1]],
                        quantile(df$SMILE, 0.75)[[1]], 172),
                        right=FALSE,
                        labels= c("SMILE_1", "SMILE_2",
                        "SMILE_3", "SMILE_4"),
                        include.lowest = TRUE)

# Diagnosis of Depression
df$Depression <- factor(df$PHQ>=3, labels=c("Negative","Depression"))

# Anxiety Diagnosis
df$Anxiety <- factor(df$GAD>=10, labels=c("Negative", "Anxiety"))

# Simultaneous diagnosis of Anxiety and Depression
df$Anxiety_Depression <- factor(df$GAD>=10 & df$PHQ>=3,
    labels=c("Negative","Anxiety and Depression"))

# Diagnosis of Anxiety and Depression: Negative, only anxiety,
# only depression, and both
df$Anxiety_Depression_comp <- interaction(df$Depression,
    df$Anxiety, sep = "_")
df$Anxiety_Depression_comp <- ordered(df$Anxiety_Depression_comp,
    levels=c("Negative_Negative", "Negative_Anxiety",
        "Depression_Negative", "Depression_Anxiety"),
    labels=c("Negative", "Anxiety", "Depression",
        "Anxiety and Depression"))
unique(df$Anxiety_Depression_comp)

# Remove individual that indicates 2 years old
```

```
df <- df[df$Age > 17,]

# Create age groups
df$age_group <- cut(df$Age,
                        breaks=c(18,20,22,25,seq(30,60,10)),
                        right=FALSE,
                        include.lowest = TRUE)

# Numerical dataset
dfN <- df
for (i in 6:57) {
  dfN[, i] <- as.numeric(df[, i])
}
for (i in 70:72) {
  dfN[, i] <- ifelse(df[, i] == "Negative", 0, 1)
}

str(df)
str(dfN)
colnames(df)
save.image(file="dfp.Rdata")

############################
load("dfp.Rdata")
############################

############################
### Characterization of each variable
############################

#### Gender Distribution

gender_counts <- table(df$Gender)
gender_perc <- round(100 * gender_counts / sum(gender_counts), 1)
gender_counts
gender_perc
pie(gender_counts,
    labels = paste0(names(gender_counts), " (", gender_perc, "%)"),
    main = "Gender distribution",
    col = c("pink", "blue", "green"),
    main=NULL,xlab=NULL, ylab=NULL)

#### School Distribution

Escola_counts <- table(df$School)
Escola_perc <- round(100 * Escola_counts / sum(Escola_counts), 1)
Escola_counts
Escola_perc
pie(Escola_counts,
```

```
    labels = paste0(names(Escola_counts), " (", Escola_perc, "%)"),
    main = "School Distribution",
    col = rainbow(length(Escola_counts)))

#### Education Level Distribution

Level_counts <- table(df$Course)
Level_perc <- round(100 * Level_counts / sum(Level_counts), 1)
Level_counts
Level_perc
pie(Level_counts,
    labels = paste0(names(Level_counts), " (", Level_perc, "%)"),
    main = "Education Level Distribution",
    col = rainbow(length(Level_counts)))

#### Course versus School

ggplot(data=df, aes(School,fill=Course)) +
  geom_bar(position = "dodge") +
  labs(x = "Escola",
       y = "Número de estudantes",
       title ="Tipo de curso por Escola") +
  scale_fill_manual(values=c("lightgreen", "lightblue", "orange"),
                    name = "Curso")


ggplot(data=df, aes(School,fill=Course)) +
  geom_bar(position = "fill") +
  labs(x = "School",
       y = "Proportion of students",
       #title ="Education level by school"
       )

#### Family Displacement

Family_counts <- table(df$Displaced)
Family_perc <- round(100 * Family_counts / sum(Family_counts), 1)
Family_counts
Family_perc
pie(Family_counts,
    labels = paste0(names(Family_counts), " (", Family_perc, "%)"),
    main = "Family Displacement",
    col = rainbow(length(Family_counts)))

ggplot(data=df, aes(Displaced,fill=Gender)) +
  geom_bar(position = "dodge") +
  labs(x = "School",
       y = "Number of students",
       title ="Gender by deslocado") +
```

```
  scale_fill_manual(values=c("pink", "blue", "green"))

ggplot(data=df, aes(Displaced,fill=Gender)) +
  geom_bar(position = "fill") +
  labs(x = "School",
       y = "Proportion of students",
       title ="Gender by deslocado"
       ) +
  scale_fill_manual(values=c("pink", "blue", "green"))


#### Age distribution

summary(df$Age)

hist(df$Age, breaks = seq(18,62,4), col = "lightblue")

ggplot(data=df, aes(Age)) +
  geom_bar(fill = "lightblue") +
  labs(x = "Age",
       y = "Number of students",
       title ="Age")

ggplot(data=df, aes(Age,fill=Gender)) +
  geom_bar() +
  labs(x = "Age",
       y = "Number of students") +
  scale_fill_manual(values=c("pink", "blue", "green"),
                    name = "Gender")

ggplot(data=df, aes(Age,fill=Gender)) +
  geom_histogram(alpha = 0.75, position = "identity",
                 breaks=seq(18,60,1)) +
  labs(x = "Age",
       #y = "Número de estudantes"
       ) +
  scale_fill_manual(values=c("pink", "blue", "green"),
                    name = "Gender")

table(df$age_group)
round(prop.table(table(df$age_group))*100,2)

ggplot(data=df, aes(age_group,fill=Gender)) +
  geom_bar() +
  labs(x = "School",
       y = "Gender distribution",
       title ="Gender by age") +
  scale_fill_manual(values=c("pink", "blue", "green"))
```

```
ggplot(data=df, aes(age_group,fill=Gender)) +
  geom_bar(position = "fill") +
  labs(x = "School",
       y = "Gender distribution",
       title ="Gender by age") +
  scale_fill_manual(values=c("pink", "blue", "green"))

table(df$Gender, df$age_group)
table(df$age_group)
table(df$Gender)

names(df)
School <- unique(df$School)
Course <- unique(df$Course)

for(i in 1:5){for(j in 1:3){
  print(School[i])
  print(Course[j])
  print(table(df[df$School==School[i] &
           df$Course==Course[j],]$Gender));
  print(table(df[df$School==School[i] &
             df$Course==Course[j],]$Displaced))
}}

for(i in 1:5){
  print(School[i])
  print(table(df[df$School==School[i],]$Gender));
  print(table(df[df$School==School[i],]$Displaced));
  print(table(df[df$School==School[i],]$Course))
}

for(j in 1:3){
  print(Course[j])
  print(table(df[df$Course==Course[j],]$Gender));
  print(table(df[df$Course==Course[j],]$Displaced))
}

table(df$School)
table(df$Gender)
table(df$Displaced)
table(df$Course)

######## PHQ Score and Depression

ggplot(data=df, aes(PHQ)) +
  geom_bar(fill = "lightblue") +
  labs(x = "Pontuação PHQ-2",
       y = "Number of students")
```

```
ggplot(data=df, aes(PHQ)) +
  geom_bar(fill = c(rep("lightblue",3),rep("firebrick",4))) +
  labs(title ="PHQ-2 e Depression",
       x = "PHQ-2 score",
       y = "Number of students")

summary(df$PHQ)

Depression_counts <- table(df$Depression)
Depression_perc <- round(prop.table(Depression_counts)*100, 1)
Depression_counts
Depression_perc
pie(table(df$Depression),
    labels = paste0(names(Depression_counts),
        " (", Depression_perc, "%)"),
    main = "Diagnóstico de Depression pelo PHQ",
    col = c("lightblue","firebrick"))

######## GAD Score and Anxiety

ggplot(data=df, aes(GAD)) +
  geom_bar(fill = "lightgreen") +
  labs(x = "GAD score",
       y = "Number of students")

ggplot(data=df, aes(GAD)) +
  geom_bar(fill = c(rep("lightgreen",10),rep("firebrick",12))) +
  labs(title ="GAD e Anxiety",
       x = "GAD score",
       y = "Number of students")

summary(df$GAD)

Anxiety_counts <- table(df$Anxiety)
Anxiety_perc <- round(prop.table(Anxiety_counts)*100, 1)
Anxiety_counts
Anxiety_perc

pie(table(df$Anxiety),
    labels = paste0(names(Anxiety_counts), " (", Anxiety_perc, "%)"),
    main = "Diagnóstico de Anxiety pelo GAD",
    col = c("lightgreen","firebrick"))

######## Simultaneous depression and anxiety

AD_counts <- table(df$Anxiety_Depression)
AD_perc <- round(prop.table(AD_counts)*100, 1)
AD_counts
AD_perc
```

```
pie(table(df$Anxiety_Depression),
    labels = paste0(names(AD_counts), " (", AD_perc, "%)"),
    main = "Simultaneous depression and anxiety diagnosis",
    col = rainbow(length(AD_counts)))

######## Depression, anxiety, and both

ADC_counts <- table(df$Anxiety_Depression_comp)
ADC_perc <- round(prop.table(ADC_counts)*100, 1)
ADC_counts
ADC_perc
pie(table(df$Anxiety_Depression_comp),
    labels = paste0(names(ADC_counts), " (", ADC_perc, "%)"),
    main = "Diagnóstico de Anxiety and Depression pelos PHQ e GAD",
    col = c("lightgreen", "yellow", "orange", "firebrick"))

ggplot(data=df, aes(School,fill=Anxiety_Depression_comp)) +
  geom_bar(position = "fill") +
  labs(x = "School",
       y = "Proportion of students",
       title ="Education level by school"
  ) +
  scale_fill_manual(values=c("lightgreen", "yellow",
    "orange", "firebrick"),
    name = "")


ggplot(data=df, aes(Course,fill=Anxiety_Depression_comp)) +
  geom_bar(position = "fill") +
  labs(x = "Course",
       y = "Proportion of students",
       title ="Education level by school"
  ) +
  scale_fill_manual(values=c("lightgreen", "yellow",
    "orange", "firebrick"),
    name = "")


ggplot(data=df, aes(Gender,fill=Anxiety_Depression_comp)) +
  geom_bar(position = "fill") +
  labs(x = "Gender",
       y = "Proportion of students",
       title ="Education level by school"
  ) +
  scale_fill_manual(values=c("lightgreen", "yellow",
  "orange", "firebrick"),
  name = "")
```

```
ggplot(data=df, aes(Displaced,fill=Anxiety_Depression_comp)) +
  geom_bar(position = "fill") +
  labs(x = "Displaced",
       y = "Proportion of students",
       title ="Education level by school"
  ) +
  scale_fill_manual(values=c("lightgreen", "yellow",
  "orange", "firebrick"),
  name = "")

##########PHQ
#### School
summary_School <- aggregate(PHQ ~ School, data = df,
    FUN = function(x) c(mean = mean(x), sd = sd(x)))
summary_School
aggregate(PHQ ~ School, data = df, FUN = summary)
# Test of normality in each School
tapply(df$PHQ, df$School, shapiro.test)
# test of homogeneity of variances
bartlett.test(PHQ ~ School, data = dfN)
leveneTest(dfN$PHQ ~ dfN$School,center="median")
# test for equality of location
oneway.test(PHQ ~ School, data = dfN, var.equal = TRUE)
kruskal.test(PHQ ~ School, data = dfN)
# Fisher's Test of Independence
set.seed(1234)
fisher.test(table(df$School, df$PHQ),
            simulate.p.value = TRUE)

#### Course
summary_Course <- aggregate(PHQ ~ Course, data = df,
    FUN = function(x) c(mean = mean(x), sd = sd(x)))
summary_Course
aggregate(PHQ ~ Course, data = df, FUN = summary)
# Test of normality in each Course
tapply(df$PHQ, df$Course, shapiro.test)
# test of homogeneity of variances
bartlett.test(PHQ ~ Course, data = dfN)
leveneTest(dfN$PHQ ~ dfN$Course,center="median")
# test for equality of location
oneway.test(PHQ ~ Course, data = dfN, var.equal = TRUE)
kruskal.test(PHQ ~ Course, data = dfN)
# Fisher's Test of Independence
set.seed(1234)
fisher.test(table(df$Course, df$PHQ),
            simulate.p.value = TRUE)

#### Gender
summary_gender <- aggregate(PHQ ~ Gender, data = df,
```

```
    FUN = function(x) c(mean = mean(x), sd = sd(x)))
summary_gender
aggregate(PHQ ~ Gender, data = df, FUN = summary)
# Test of normality in each gender
tapply(df$PHQ, df$Gender, shapiro.test)
# test of homogeneity of variances
bartlett.test(PHQ ~ Gender, data = dfN)
leveneTest(dfN$PHQ ~ dfN$Gender,center="median")
# test for equality of location
oneway.test(PHQ ~ Gender, data = dfN, var.equal = TRUE)
kruskal.test(PHQ ~ Gender, data = dfN)
# Fisher's Test of Independence
set.seed(1234)
fisher.test(table(df$Gender, df$PHQ),
            simulate.p.value = TRUE)

#### Displaced
summary_Displaced <- aggregate(PHQ ~ Displaced, data = df,
    FUN = function(x) c(mean = mean(x), sd = sd(x)))
summary_Displaced
aggregate(PHQ ~ Displaced, data = df, FUN = summary)
# Test of normality in each Displaced
tapply(df$PHQ, df$Displaced, shapiro.test)
# test of homogeneity of variances
bartlett.test(PHQ ~ Displaced, data = dfN)
leveneTest(dfN$PHQ ~ dfN$Displaced,center="median")
# test for equality of location
t.test(PHQ ~ Displaced, data = dfN, var.equal = TRUE)
wilcox.test(PHQ ~ Displaced, data = dfN)
# Fisher's Test of Independence
set.seed(1234)
fisher.test(table(df$Displaced, df$PHQ),
            simulate.p.value = TRUE)

# All
mean(df$PHQ)
sd(df$PHQ)
shapiro.test(df$PHQ)

##########GAD
#### School
summary_School <- aggregate(GAD ~ School, data = df,
    FUN = function(x) c(mean = mean(x), sd = sd(x)))
summary_School
aggregate(GAD ~ School, data = df, FUN = summary)
# Test of normality in each School
tapply(df$GAD, df$School, shapiro.test)
# test of homogeneity of variances
bartlett.test(GAD ~ School, data = dfN)
```

```
leveneTest(dfN$GAD ~ dfN$School,center="median")
# test for equality of location
oneway.test(GAD ~ School, data = dfN, var.equal = TRUE)
kruskal.test(GAD ~ School, data = dfN)
# Fisher's Test of Independence
set.seed(1234)
fisher.test(table(df$School, df$GAD),
            simulate.p.value = TRUE)

#### Course
summary_Course <- aggregate(GAD ~ Course, data = df,
    FUN = function(x) c(mean = mean(x), sd = sd(x)))
summary_Course
aggregate(GAD ~ Course, data = df, FUN = summary)
# Test of normality in each Course
tapply(df$GAD, df$Course, shapiro.test)
# test of homogeneity of variances
bartlett.test(GAD ~ Course, data = dfN)
leveneTest(dfN$GAD ~ dfN$Course,center="median")
# test for equality of location
oneway.test(GAD ~ Course, data = dfN, var.equal = TRUE)
kruskal.test(GAD ~ Course, data = dfN)
# Fisher's Test of Independence
set.seed(1234)
fisher.test(table(df$Course, df$GAD),
            simulate.p.value = TRUE)


#### Gender
summary_gender <- aggregate(GAD ~ Gender, data = df,
    FUN = function(x) c(mean = mean(x), sd = sd(x)))
summary_gender
aggregate(GAD ~ Gender, data = df, FUN = summary)
# Test of normality in each gender
tapply(df$GAD, df$Gender, shapiro.test)
# test of homogeneity of variances
bartlett.test(GAD ~ Gender, data = dfN)
leveneTest(dfN$GAD ~ dfN$Gender,center="median")
# test for equality of location
oneway.test(GAD ~ Gender, data = dfN, var.equal = TRUE)
kruskal.test(GAD ~ Gender, data = dfN)
# Fisher's Test of Independence
set.seed(1234)
fisher.test(table(df$Gender, df$GAD),
            simulate.p.value = TRUE)

#### Displaced
summary_Displaced <- aggregate(GAD ~ Displaced, data = df,
    FUN = function(x) c(mean = mean(x), sd = sd(x)))
```

```
summary_Displaced
aggregate(GAD ~ Displaced, data = df, FUN = summary)
# Test of normality in each Displaced
tapply(df$GAD, df$Displaced, shapiro.test)
# test of homogeneity of variances
bartlett.test(GAD ~ Displaced, data = dfN)
leveneTest(dfN$GAD ~ dfN$Displaced,center="median")
# test for equality of location
t.test(GAD ~ Displaced, data = dfN, var.equal = TRUE)
wilcox.test(GAD ~ Displaced, data = dfN)
# Fisher's Test of Independence
set.seed(1234)
fisher.test(table(df$Displaced, df$GAD),
            simulate.p.value = TRUE)

# All
mean(df$GAD)
sd(df$GAD)
shapiro.test(df$GAD)


######## SMILE score
#### Histogram of the Total Score

barplot(table(df$SMILE), xlab="Total Score",
    main="Histogram of the Total Score")

ggplot(data=df, aes(SMILE)) +
  geom_bar(fill = "lightblue") +
  labs(x = "SMILE",
       y = "Number of students",
       title ="SMILE")

ggplot(df, aes(x=SMILE)) +
  geom_histogram(aes(y = ..density..),
                 fill="lightblue",breaks=seq(77,152,5)) +
  labs(x = "SMILE score", y = "Frequency",
       title ="SMILE Histogram",
       subtitle = "Students at the Polytechnic of Leiria"
       ) +
  geom_density(lwd = 1,
               linetype = 1,
               colour = "blue",
               fill = 4, alpha = 0.25) +
  scale_x_continuous(limits = c(75, 155))


ggplot(df, aes(x=SMILE)) +
  geom_boxplot(color="blue") +
```

```
  labs(x = "SMILE score", y = "",
       title ="SMILE score boxplot",
       subtitle = "Students at the Polytechnic of Leiria"
       ) +
  scale_y_discrete(label="")

ggplot(df, aes(sample=SMILE)) +
  stat_qq(distribution = qnorm,color="blue") +
  stat_qq_line() +
  labs(x = "Normal quantiles", y = "SMILE quantiles",
       title ="SMILE score boxplot",
       subtitle = "Students at the Polytechnic of Leiria"
  )

summary(df$SMILE)
sd(df$SMILE)
mean(df$SMILE)

# Normality
shapiro.test(df$SMILE)

### SMILE divided by classes
table(df$SMILE_Classes)
summary_SMILE_Classes <- aggregate(SMILE ~ df$SMILE_Classes, data = df,
    FUN = function(x) c(mean = mean(x), sd = sd(x)))
summary_SMILE_Classes
aggregate(SMILE ~ SMILE_Classes, data = df, FUN = summary)

############################
### SMILE by Groups
############################

#### SMILE by gender

ggplot(df, aes(SMILE,fill=Gender)) +
  geom_histogram(binwidth=5) +
  labs(x = "SMILE score", y = "Frequency",
       title ="SMILE Histogram by gender",
       subtitle = "Students at the Polytechnic of Leiria") +
  scale_fill_manual(values=c("pink", "blue", "green"))

ggplot(df, aes(x = SMILE, fill = Gender)) +
  geom_density(alpha = 0.5) +
  scale_fill_manual(values = c("pink", "blue", "green")) +
  labs(title = "Distribution of Total Scores by School",
       x = "Total Score", y = "Density") +
  theme_classic() +
  scale_x_continuous(limits = c(75, 155))
```

```
ggplot(df, aes(SMILE, Gender)) +
  geom_boxplot(color=c("pink","blue","green")) +
  labs(x = "SMILE score", y = "",
       title ="SMILE score boxplot by gender",
       subtitle = "Students at the Polytechnic of Leiria")

summary_gender <- aggregate(SMILE ~ Gender, data = df,
    FUN = function(x) c(mean = mean(x), sd = sd(x)))
summary_gender
aggregate(SMILE ~ Gender, data = df, FUN = summary)

# Test of normality in each gender
tapply(df$SMILE, df$Gender, shapiro.test)

# Equality of variances test
bartlett.test(SMILE ~ Gender, data = dfN)

library(car);
leveneTest(dfN$SMILE ~ dfN$Gender,center="median")

oneway.test(SMILE ~ Gender, data = dfN, var.equal = FALSE)
kruskal.test(SMILE ~ Gender, data = dfN)

# Fisher's Test of Independence
set.seed(1234)
fisher.test(table(df$Gender, df$SMILE),
            simulate.p.value = TRUE)

# Removing category Non-binary, which only has 10 observations.
dfN_aux <- dfN[df$Gender != "Non-binary",]
table(dfN_aux$Gender)

# Equality of variances test
var.test(SMILE ~ Gender, data = dfN[df$Gender != "Não binário",])

# Equality of location test
t.test(SMILE ~ Gender, data = dfN[df$Gender != "Não binário",],
    var.equal = TRUE)
wilcox.test(SMILE ~ Gender, data = dfN[df$Gender != "Não binário",])

table(dfN$Gender)

############################
#### SMILE by course - Education Level

ggplot(df, aes(x = SMILE, fill = Course)) +
  geom_density(alpha = 0.5) +
  scale_fill_manual(values = rainbow(3)) +
  labs(title = "Distribution of Total Scores by Education level",
```

```
       x = "Total Score", y = "Density") +
  theme_classic() +
  scale_x_continuous(limits = c(75, 155))

ggplot(df, aes(SMILE, Course)) +
  geom_boxplot(color=rainbow(3)) +
  labs(x = "SMILE score", y = "",
       title ="SMILE score boxplot by course type",
       subtitle = "Students at the Polytechnic of Leiria")

summary_data_Course <- aggregate(SMILE ~ Course, data = df,
 FUN = function(x) c(mean = mean(x), sd = sd(x)))
summary_data_Course
aggregate(SMILE ~ Course, data = df, FUN = summary)

# Test of normality in each course
tapply(df$SMILE, df$Course, shapiro.test)

# Equality of variances test
bartlett.test(SMILE ~ Course, data = dfN)

# Equality of location test
oneway.test(SMILE ~ Course, data = dfN, var.equal = TRUE)
kruskal.test(SMILE ~ Course, data = dfN)

# Fisher's Test of Independence
set.seed(1234)
fisher.test(table(df$Course, df$SMILE),
            simulate.p.value = TRUE)

############################
#### SMILE by School

ggplot(df, aes(SMILE, School)) +
  geom_boxplot(color=rainbow(5)) +
  labs(x = "SMILE score", y = "",
       title ="SMILE score boxplot by age group",
       subtitle = "Students at the Polytechnic of Leiria")

ggplot(df, aes(x = SMILE, fill = School)) +
  geom_density(alpha = 0.5) +
  scale_fill_manual(values = c("red", "blue", "green",
    "yellow", "cyan")) +
  labs(title = "Distribution of Total Scores by School",
       x = "Total Score", y = "Density") +
  theme_classic()+
  scale_x_continuous(limits = c(75, 155))

summary_data_School <- aggregate(SMILE ~ School, data = df,
```

```
    FUN = function(x) c(mean = mean(x), sd = sd(x)))
summary_data_School
aggregate(SMILE ~ School, data = df, FUN = summary)

# Test of normality in each School
tapply(df$SMILE, df$School, shapiro.test)

# Equality of variances test
leveneTest(SMILE ~ School, data = dfN,  center= "median")

# Equality of location test
oneway.test(SMILE ~ School, data = dfN, var.equal = TRUE)
kruskal.test(SMILE ~ School, data = dfN)

tapply(dfN$SMILE, dfN$School, mean)
tapply(dfN$SMILE, dfN$School, sd)

# Fisher's Test of Independence
set.seed(1234)
fisher.test(table(df$School, df$SMILE),
            simulate.p.value = TRUE)

############################
#### SMILE by displaced or non-displaced students - by Family

ggplot(df, aes(x = SMILE, fill = Displaced)) +
  geom_density(alpha = 0.5) +
  scale_fill_manual(values = rainbow(2)) +
  labs(title = "Distribution of Total Scores by Family Status",
       x = "Total Score", y = "Density") +
  theme_classic()+
  scale_x_continuous(limits = c(75, 155))

ggplot(df, aes(SMILE, Displaced)) +
  geom_boxplot(color=c("red","green")) +
  labs(x = "SMILE score", y = "",
       title ="SMILE score boxplot by dislocated",
       subtitle = "Students at the Polytechnic of Leiria")

summary_data_Displaced <- aggregate(SMILE ~ Displaced, data = df,
    FUN = function(x) c(mean = mean(x), sd = sd(x)))
summary_data_Displaced
aggregate(SMILE ~ Displaced, data = df, FUN = summary)

# Test of normality
tapply(df$SMILE, df$Displaced, shapiro.test)

# Equality of variances test
var.test(SMILE ~ Displaced, data = dfN)
```

```
# Equality of location test
t.test(SMILE ~ Displaced, data = dfN, var.equal = TRUE)
wilcox.test(SMILE ~ Displaced, data = dfN)

tapply(dfN$SMILE, dfN$Displaced, mean)
tapply(dfN$SMILE, dfN$Displaced, sd)

# Fisher's Test of Independence
set.seed(1234)
fisher.test(table(df$Gender, df$SMILE),
            simulate.p.value = TRUE)

############################
#### SMILE by Age

ggplot(df, aes(x = Age, y = SMILE)) +
  geom_point(color="blue") +
  theme_classic()

ggplot(df, aes(SMILE, age_group)) +
  geom_boxplot(color=rainbow(7)) +
  labs(x = "SMILE score", y = "",
       title ="SMILE score boxplot by age group",
       subtitle = "Students at the Polytechnic of Leiria")

ggplot(df, aes(x = SMILE, fill = age_group)) +
  geom_density(alpha = 0.5) +
  scale_fill_manual(values = rainbow(7)) +
  labs(title = "Distribution of Total Scores by School",
    x = "Total Score", y = "Density") +
  theme_classic()

summary_data_age <- aggregate(SMILE ~ Age, data = df,
    FUN = function(x) c(mean = mean(x), sd = sd(x)))
summary_data_age
aggregate(SMILE ~ Age, data = df, FUN = summary)

# Test of normality
shapiro.test(df$Age)
# Correlation test
cor.test(df$Age, df$SMILE, method = "spearman", exact=FALSE)

summary_age_group <- aggregate(SMILE ~ age_group, data = df,
    FUN = function(x) c(mean = mean(x), sd = sd(x)))
summary_age_group
aggregate(SMILE ~ age_group, data = df, FUN = summary)

table(df$age_group)
```

```
# Test of normality in each School
tapply(df$SMILE, df$age_group, shapiro.test)

# Equality of variances test
leveneTest(SMILE ~ age_group, data = dfN,  center= "median")

# Equality of location test
oneway.test(SMILE ~ age_group, data = dfN, var.equal = TRUE)
kruskal.test(SMILE ~ age_group, data = dfN)

############################
### Correlations and survey validation
############################

# DN Diet and nutrition
corr_matrix <- cor(dfN[,IDN],method = c("spearman"))
p.mat <- cor_pmat(dfN[,IDN])
round(corr_matrix,3)
ggcorrplot(corr_matrix, hc.order = TRUE, type = "lower",
    p.mat=p.mat, lab = TRUE)

alpha(dfN[,IDN]) # SMILE
KMO(dfN[,IDN])
alpha(dfN[,intersect(IDN, ISMILEC)]) # SMILE-C
KMO(dfN[,intersect(IDN, ISMILEC)])

names(dfN[,IDN])
names(dfN[,intersect(IDN, ISMILEC)])

table(df[,6])
plot(likert(df[,IDN]),
     ordered = FALSE,
     wrap = 30) +
  ggtitle("") +
  labs(x = "",
       y = "Percentage of responses") +
  guides(fill=guide_legend(title="responses"))

# SU Substance Use
corr_matrix <- cor(dfN[,ISU],method = c("spearman"))
p.mat <- cor_pmat(dfN[,ISU])
round(corr_matrix,3)
ggcorrplot(corr_matrix, hc.order = TRUE, type = "lower",
    p.mat=p.mat, lab = TRUE)

alpha(dfN[,ISU]) # SMILE
KMO(dfN[,ISU])
alpha(dfN[,intersect(ISU, ISMILEC)]) # SMILE-C
```

```
KMO(dfN[,intersect(ISU, ISMILEC)])

names(dfN[,ISU])
names(dfN[,intersect(ISU, ISMILEC)])

plot(likert(df[,ISU]),
     ordered = FALSE,
     wrap = 30) +
  ggtitle("") +
  labs(x = "",
       y = "Percentage of responses") +
  guides(fill=guide_legend(title="responses"))

# PA Physical activity
corr_matrix <- cor(dfN[,IPA],method = c("spearman"))
p.mat <- cor_pmat(dfN[,IPA])
round(corr_matrix,3)
ggcorrplot(corr_matrix, hc.order = TRUE, type = "lower",
    p.mat=p.mat, lab = TRUE)

alpha(dfN[,IPA]) # SMILE
KMO(dfN[,IPA])
#alpha(dfN[,intersect(, ISMILEC)]) # SMILE-C
#KMO(dfN[,intersect(IPA, ISMILEC)])

names(dfN[,IPA])
names(dfN[,intersect(IPA, ISMILEC)])

plot(likert(df[,IPA]),
     ordered = FALSE,
     wrap = 30) +
  ggtitle("") +
  labs(x = "",
       y = "Percentage of responses") +
  guides(fill=guide_legend(title="responses"))

# SM Stress management
corr_matrix <- cor(dfN[,ISM],method = c("spearman"))
p.mat <- cor_pmat(dfN[,ISM])
round(corr_matrix,3)
ggcorrplot(corr_matrix, hc.order = TRUE, type = "lower",
    p.mat=p.mat, lab = TRUE)

alpha(dfN[,ISM]) # SMILE
KMO(dfN[,ISM])
alpha(dfN[,intersect(ISM, ISMILEC)]) # SMILE-C
KMO(dfN[,intersect(ISM, ISMILEC)])

names(dfN[,ISM])
```

```
names(dfN[,intersect(ISM, ISMILEC)])

plot(likert(df[,ISM]),
     ordered = FALSE,
     wrap = 30) +
  ggtitle("") +
  labs(x = "",
       y = "Percentage of responses") +
  guides(fill=guide_legend(title="responses"))

# RS Restorative sleep
corr_matrix <- cor(dfN[,IRS],method = c("spearman"))
p.mat <- cor_pmat(dfN[,IRS])
round(corr_matrix,3)
ggcorrplot(corr_matrix, hc.order = TRUE, type = "lower",
 p.mat=p.mat, lab = TRUE)

alpha(dfN[,IRS]) # SMILE
KMO(dfN[,IRS])
alpha(dfN[,intersect(IRS, ISMILEC)]) # SMILE-C
KMO(dfN[,intersect(IRS, ISMILEC)])

names(dfN[,IRS])
names(dfN[,intersect(IRS, ISMILEC)])

plot(likert(df[,IRS]),
     ordered = FALSE,
     wrap = 30) +
  ggtitle("") +
  labs(x = "",
       y = "Percentage of responses") +
  guides(fill=guide_legend(title="responses"))

# SS Social support
corr_matrix <- cor(dfN[,ISS],method = c("spearman"))
p.mat <- cor_pmat(dfN[,ISS])
round(corr_matrix,3)
ggcorrplot(corr_matrix, hc.order = TRUE, type = "lower",
    p.mat=p.mat, lab = TRUE)

alpha(dfN[,ISS]) # SMILE
KMO(dfN[,ISS])
alpha(dfN[,intersect(ISS, ISMILEC)]) # SMILE-C
KMO(dfN[,intersect(ISS, ISMILEC)])

names(dfN[,ISS])
names(dfN[,intersect(ISS, ISMILEC)])

plot(likert(df[,ISS]),
```

```
      ordered = FALSE,
      wrap = 30) +
  ggtitle("") +
  labs(x = "",
       y = "Percentage of responses") +
  guides(fill=guide_legend(title="responses"))

# EE Environment exposures
corr_matrix <- cor(dfN[,IEE],method = c("spearman"))
p.mat <- cor_pmat(dfN[,IEE])
round(corr_matrix,3)
ggcorrplot(corr_matrix, hc.order = TRUE, type = "lower",
    p.mat=p.mat, lab = TRUE)

alpha(dfN[,IEE], check.keys=TRUE) # SMILE
KMO(dfN[,IEE])
#alpha(dfN[,intersect(IEE, ISMILEC)]) # SMILE-C
KMO(dfN[,intersect(IEE, ISMILEC)])

names(dfN[,IEE])
names(dfN[,intersect(IEE, ISMILEC)])

plot(likert(df[,IEE]),
      ordered = FALSE,
      wrap = 30) +
  ggtitle("") +
  labs(x = "",
       y = "Percentage of responses") +
  guides(fill=guide_legend(title="responses"))

# between domains
corr_matrix <- cor(dfN[,62:68],method = c("spearman"))
p.mat <- cor_pmat(dfN[,62:68])
round(corr_matrix,3)
ggcorrplot(corr_matrix, hc.order = TRUE, type = "lower",
    p.mat=p.mat, lab = TRUE)


# SMILE
corr_matrix <- cor(dfN[,ISMILE],method = c("spearman"))
p.mat <- cor_pmat(dfN[,ISMILE])
round(corr_matrix,3)
ggcorrplot(corr_matrix, hc.order = TRUE, type = "lower",
    p.mat=p.mat, lab = TRUE)

alpha(dfN[,ISMILE], check.keys=TRUE)
KMO(dfN[,ISMILE])

# SMILE-C
```

```
corr_matrix <- cor(dfN[,ISMILEC],method = c("spearman"))
p.mat <- cor_pmat(dfN[,ISMILEC])
round(corr_matrix,3)
ggcorrplot(corr_matrix, hc.order = TRUE, type = "lower",
    p.mat=p.mat, lab = TRUE)

alpha(dfN[,ISMILEC], check.keys=TRUE)
KMO(dfN[,ISMILEC])

# PHQ
corr_matrix <- cor(dfN[,IPHQ],method = c("spearman"))
p.mat <- cor_pmat(dfN[,IPHQ])
round(corr_matrix,3)
ggcorrplot(corr_matrix, hc.order = TRUE, type = "lower",
 p.mat=p.mat, lab = TRUE)

alpha(dfN[,IPHQ])
KMO(dfN[,IPHQ])

plot(likert(df[,IPHQ]),
     ordered = FALSE,
     wrap = 30) +
  ggtitle("") +
  labs(x = "",
       y = "Percentage of responses") +
  guides(fill=guide_legend(title=""))

# GAD
corr_matrix <- cor(dfN[,IGAD],method = c("spearman"))
p.mat <- cor_pmat(dfN[,IGAD])
round(corr_matrix,3)
ggcorrplot(corr_matrix, hc.order = TRUE, type = "lower",
 p.mat=p.mat, lab = TRUE)

alpha(dfN[,IGAD])
KMO(dfN[,IGAD])

plot(likert(df[,IGAD]),
     ordered = FALSE,
     wrap = 30) +
  ggtitle("") +
  labs(x = "",
       y = "Percentage of responses") +
  guides(fill=guide_legend(title=""))

# PHQ and GAD

dfN$SMILEC
corr_matrix <- cor(dfN[,c("SMILE","SMILEC","PHQ","GAD")],
```

```
    method = c("spearman"))
p.mat <- cor_pmat(dfN[,IGAD])
round(corr_matrix,3)
ggcorrplot(corr_matrix, hc.order = TRUE, type = "lower",
    p.mat=p.mat, lab = TRUE)

############################
#### SMILE versus SMILE-C

# Test of normality
shapiro.test(df$SMILEC)

cor.test(df$SMILE, df$SMILEC, method = "spearman", exact=FALSE)

ggplot(df, aes(SMILE, SMILEC)) +
  geom_point(color="blue") +
  labs(y = "SMILE-C score", x = "SMILE score",
       title ="SMILE versus SMILE-C",
       subtitle = "Students at the Polytechnic of Leiria")

ggplot(df, aes(SMILE, SMILEC)) +
  geom_point(aes(color=Depression, shape=Anxiety)) +
  labs(y = "SMILE-C score", x = "SMILE score",
       title ="SMILE versus SMILE-C",
       subtitle = "Students at the Polytechnic of Leiria")

############################
#### SMILE by Depression

ggplot(df, aes(x = SMILE, fill = Depression)) +
  geom_density(alpha = 0.9) +
  scale_fill_manual(values = c("lightblue", "firebrick")) +
  labs(title = "Distribution of SMILE depending on Depression",
       x = "SMILE", y = "Densidade") +
  theme_classic() +
  guides(fill=guide_legend(title="PHQ"))


ggplot(df, aes(SMILE, Depression)) +
  geom_boxplot(color=c("green", "red")) +
  labs(x = "SMILE score", y = "",
       title ="SMILE score boxplot by depression",
       subtitle = "Students at the Polytechnic of Leiria")

summary_data_Depression <- aggregate(SMILE ~ Depression, data = df,
    FUN = function(x) c(mean = mean(x), sd = sd(x)))
summary_data_Depression
aggregate(SMILE ~ Depression, data = df, FUN = summary)
```

```
# Test of normality
tapply(df$SMILE, df$Depression, shapiro.test)

table(df$Depression)

# Equality of variances test
var.test(SMILE ~ Depression, data = df)

# Equality of location test
t.test(SMILE ~ Depression, data = df, var.equal = FALSE)
wilcox.test(SMILE ~ Depression, data = df)

############################
#### SMILE by Anxiety

ggplot(df, aes(x = SMILE, fill = Anxiety)) +
  geom_density() +
  scale_fill_manual(values = c("lightgreen", "firebrick")) +
  labs(title = "Distribution of SMILE depending on Anxiety",
       x = "SMILE", y = "Densidade") +
  theme_classic() +
  guides(fill=guide_legend(title="GAD"))

ggplot(df, aes(SMILE, Anxiety)) +
  geom_boxplot(color=c("green", "red")) +
  labs(x = "SMILE score", y = "",
       title ="SMILE score boxplot by anxiety",
       subtitle = "Students at the Polytechnic of Leiria")

summary_data_Anxiety <- aggregate(SMILE ~ Anxiety, data = df,
    FUN = function(x) c(mean = mean(x), sd = sd(x)))
summary_data_Anxiety
aggregate(SMILE ~ Anxiety, data = df, FUN = summary)

# Test of normality
tapply(df$SMILE, df$Anxiety, shapiro.test)

# Equality of variances test
var.test(SMILE ~ Anxiety, data = df)

# Equality of location test
t.test(SMILE ~ Anxiety, data = df, var.equal = FALSE)
wilcox.test(SMILE ~ Anxiety, data = df)

############################
#### SMILE by simultaneous anxiety and depression

ggplot(df, aes(x = SMILE, fill = Anxiety_Depression)) +
  geom_density(alpha = 0.5) +
```

```
  scale_fill_manual(values = c("green", "red")) +
  labs(title = "Distribution anxiety",
    x = "Total Score", y = "Density") +
  theme_classic()

ggplot(df, aes(SMILE, Anxiety_Depression)) +
  geom_boxplot(color=c("green", "red")) +
  labs(x = "SMILE score", y = "",
       title ="SMILE score boxplot by anxiety",
       subtitle = "Students at the Polytechnic of Leiria")

summary_data_Anx_Dep <- aggregate(SMILE ~ Anxiety_Depression, data = df,
    FUN = function(x) c(mean = mean(x), sd = sd(x)))
summary_data_Anx_Dep
aggregate(SMILE ~ Anxiety_Depression, data = df, FUN = summary)

# Test of normality
tapply(df$SMILE, df$Anxiety_Depression, shapiro.test)

# Equality of variances test
var.test(SMILE ~ Anxiety_Depression, data = df)

# Equality of location test
t.test(SMILE ~ Anxiety_Depression, data = df, var.equal = FALSE)
wilcox.test(SMILE ~ Anxiety_Depression, data = df)

############################
#### SMILE by anxiety, depression, both or negative

ggplot(df, aes(x = SMILE, fill = Anxiety_Depression_comp)) +
  geom_density(alpha = 1) +
  scale_fill_manual(values = c("lightgreen", "yellow",
    "orange", "firebrick")) +
  labs(title = "Distribution of SMILE depending
        on Anxiety and Depression",
       x = "SMILE", y = "Densidade") +
  theme_classic() +
  guides(fill=guide_legend(title="PHQ + GAD"))


ggplot(df, aes(SMILE, Anxiety_Depression_comp)) +
  geom_boxplot(color=c("green","yellow","orange","red")) +
  labs(x = "SMILE score", y = "",
       title ="SMILE score boxplot by anxiety",
       subtitle = "Students at the Polytechnic of Leiria")

summary_data_Anx_Comp <- aggregate(SMILE ~ Anxiety_Depression_comp,
    data = df, FUN = function(x) c(mean = mean(x), sd = sd(x)))
summary_data_Anx_Comp
```

```
aggregate(SMILE ~ Anxiety_Depression_comp, data = df, FUN = summary)

# Test of normality
tapply(df$SMILE, df$Anxiety_Depression_comp, shapiro.test)

# Equality of variances test
bartlett.test(SMILE ~ Anxiety_Depression_comp, data = dfN)

# Equality of location test
oneway.test(SMILE ~ Anxiety_Depression_comp, data = dfN,
    var.equal = FALSE)
kruskal.test(SMILE ~ Anxiety_Depression_comp, data = dfN)


############################
### Anxiety and Depression classification
### based on SMILE scores
############################

############################
#### Univariate ROC Curves
par(mar=c(0,0,0,0), omi=c(0,0,0,0))
#########################
Cores <- c("red", "orange", "darksalmon", "purple", "deeppink3",
    "green", "firebrick", "blue", "darkturquoise")

#Anxiety
roc_A_SMILE = roc(df$Anxiety == "Negative" ~ df$SMILE)
roc_A_SMILEC = roc(df$Anxiety == "Negative" ~ df$SMILEC)
roc_A_SMILE_DN = roc(df$Anxiety == "Negative" ~ df$SMILE_DN)
roc_A_SMILE_SU = roc(df$Anxiety == "Negative" ~ df$SMILE_SU)
roc_A_SMILE_PA = roc(df$Anxiety == "Negative" ~ df$SMILE_PA)
roc_A_SMILE_SM = roc(df$Anxiety == "Negative" ~ df$SMILE_SM)
roc_A_SMILE_RS = roc(df$Anxiety == "Negative" ~ df$SMILE_RS)
roc_A_SMILE_SS = roc(df$Anxiety == "Negative" ~ df$SMILE_SS)
roc_A_SMILE_EE = roc(df$Anxiety == "Negative" ~ df$SMILE_EE)
plot(roc_A_SMILE, print.auc = TRUE, col = Cores[1], lwd =2,
     print.auc.y=0.98, print.auc.x=0.98,
     auc.polygon = TRUE, max.auc.polygon = TRUE)
plot(roc_A_SMILEC, add=TRUE , print.auc = TRUE, col = Cores[2],
    lwd =2, print.auc.y = 0.93,  print.auc.x=0.98)
plot(roc_A_SMILE_DN, add=TRUE , print.auc = TRUE, col = Cores[3],
    lwd =2, print.auc.y = 0.88,  print.auc.x=0.98)
plot(roc_A_SMILE_SU, add=TRUE , print.auc = TRUE, col = Cores[4],
    lwd =2, print.auc.y = 0.83,  print.auc.x=0.98)
plot(roc_A_SMILE_PA, add=TRUE , print.auc = TRUE, col = Cores[5],
    lwd =2, print.auc.y = 0.78,  print.auc.x=0.98)
plot(roc_A_SMILE_SM, add=TRUE , print.auc = TRUE, col = Cores[6],
    lwd =2, print.auc.y = 0.73,  print.auc.x=0.98)
```

```
plot(roc_A_SMILE_RS, add=TRUE , print.auc = TRUE, col = Cores[7],
    lwd =2, print.auc.y = 0.68,  print.auc.x=0.98)
plot(roc_A_SMILE_SS, add=TRUE , print.auc = TRUE, col = Cores[8],
    lwd =2, print.auc.y = 0.63,  print.auc.x=0.98)
plot(roc_A_SMILE_EE, add=TRUE , print.auc = TRUE, col = Cores[9],
    lwd =2, print.auc.y = 0.58,  print.auc.x=0.98)
legend(0.37,0.59, c("SMILE","SMILE-C ","DN", "SU", "PA", "SM",
    "RS", "SS", "EE"),
    col = Cores, lwd =2)

#Depression
roc_A_SMILE = roc(df$Depression == "Negative" ~ df$SMILE)
roc_A_SMILEC = roc(df$Depression == "Negative" ~ df$SMILEC)
roc_A_SMILE_DN = roc(df$Depression == "Negative" ~ df$SMILE_DN)
roc_A_SMILE_SU = roc(df$Depression == "Negative" ~ df$SMILE_SU)
roc_A_SMILE_PA = roc(df$Depression == "Negative" ~ df$SMILE_PA)
roc_A_SMILE_SM = roc(df$Depression == "Negative" ~ df$SMILE_SM)
roc_A_SMILE_RS = roc(df$Depression == "Negative" ~ df$SMILE_RS)
roc_A_SMILE_SS = roc(df$Depression == "Negative" ~ df$SMILE_SS)
roc_A_SMILE_EE = roc(df$Depression == "Negative" ~ df$SMILE_EE)
plot(roc_A_SMILE, print.auc = TRUE, col = Cores[1],
    lwd =2, print.auc.y=0.98, print.auc.x=0.98,
     auc.polygon = TRUE, max.auc.polygon = TRUE)
plot(roc_A_SMILEC, add=TRUE , print.auc = TRUE, col = Cores[2],
    lwd =2, print.auc.y = 0.93,  print.auc.x=0.98)
plot(roc_A_SMILE_DN, add=TRUE , print.auc = TRUE, col = Cores[3],
    lwd =2, print.auc.y = 0.88,  print.auc.x=0.98)
plot(roc_A_SMILE_SU, add=TRUE , print.auc = TRUE, col = Cores[4],
    lwd =2, print.auc.y = 0.83,  print.auc.x=0.98)
plot(roc_A_SMILE_PA, add=TRUE , print.auc = TRUE, col = Cores[5],
    lwd =2, print.auc.y = 0.78,  print.auc.x=0.98)
plot(roc_A_SMILE_SM, add=TRUE , print.auc = TRUE, col = Cores[6],
    lwd =2, print.auc.y = 0.73,  print.auc.x=0.98)
plot(roc_A_SMILE_RS, add=TRUE , print.auc = TRUE, col = Cores[7],
    lwd =2, print.auc.y = 0.68,  print.auc.x=0.98)
plot(roc_A_SMILE_SS, add=TRUE , print.auc = TRUE, col = Cores[8],
    lwd =2, print.auc.y = 0.63,  print.auc.x=0.98)
plot(roc_A_SMILE_EE, add=TRUE , print.auc = TRUE, col = Cores[9],
    lwd =2, print.auc.y = 0.58,  print.auc.x=0.98)
legend(0.37,0.59, c("SMILE","SMILE-C ","DN", "SU", "PA", "SM",
    "RS", "SS", "EE"),
    col = Cores, lwd =2)

#Depression and Anxiety
roc_A_SMILE = roc(df$Anxiety_Depression == "Negative" ~ df$SMILE)
roc_A_SMILEC = roc(df$Anxiety_Depression == "Negative" ~ df$SMILEC)
roc_A_SMILE_DN = roc(df$Anxiety_Depression == "Negative" ~ df$SMILE_DN)
roc_A_SMILE_SU = roc(df$Anxiety_Depression == "Negative" ~ df$SMILE_SU)
roc_A_SMILE_PA = roc(df$Anxiety_Depression == "Negative" ~ df$SMILE_PA)
```

```
roc_A_SMILE_SM = roc(df$Anxiety_Depression == "Negative" ~ df$SMILE_SM)
roc_A_SMILE_RS = roc(df$Anxiety_Depression == "Negative" ~ df$SMILE_RS)
roc_A_SMILE_SS = roc(df$Anxiety_Depression == "Negative" ~ df$SMILE_SS)
roc_A_SMILE_EE = roc(df$Anxiety_Depression == "Negative" ~ df$SMILE_EE)
plot(roc_A_SMILE, print.auc = TRUE, col = Cores[1], lwd =2,
     print.auc.y=0.98, print.auc.x=0.98,
     auc.polygon = TRUE, max.auc.polygon = TRUE)
plot(roc_A_SMILEC, add=TRUE , print.auc = TRUE, col = Cores[2],
    lwd =2, print.auc.y = 0.93,  print.auc.x=0.98)
plot(roc_A_SMILE_DN, add=TRUE , print.auc = TRUE, col = Cores[3],
    lwd =2, print.auc.y = 0.88,  print.auc.x=0.98)
plot(roc_A_SMILE_SU, add=TRUE , print.auc = TRUE, col = Cores[4],
    lwd =2, print.auc.y = 0.83,  print.auc.x=0.98)
plot(roc_A_SMILE_PA, add=TRUE , print.auc = TRUE, col = Cores[5],
    lwd =2, print.auc.y = 0.78,  print.auc.x=0.98)
plot(roc_A_SMILE_SM, add=TRUE , print.auc = TRUE, col = Cores[6],
    lwd =2, print.auc.y = 0.73,  print.auc.x=0.98)
plot(roc_A_SMILE_RS, add=TRUE , print.auc = TRUE, col = Cores[7],
    lwd =2, print.auc.y = 0.68,  print.auc.x=0.98)
plot(roc_A_SMILE_SS, add=TRUE , print.auc = TRUE, col = Cores[8],
    lwd =2, print.auc.y = 0.63,  print.auc.x=0.98)
plot(roc_A_SMILE_EE, add=TRUE , print.auc = TRUE, col = Cores[9],
    lwd =2, print.auc.y = 0.58,  print.auc.x=0.98)
legend(0.37,0.59, c("SMILE","SMILE-C ","DN", "SU", "PA", "SM",
    "RS", "SS", "EE"),
    col = Cores, lwd =2)

############################
#### Preparing the data
set.seed(123)  # Set a seed for reproducibility
# Split the data into training and testing sets (80/20 split)
train_indices <- createDataPartition(df$Anxiety_Depression_comp,
    p = 0.8, list = FALSE)
df_train <- df[train_indices, ]  # Training data (80%)
df_test <- df[-train_indices, ]  # Testing data (20%)

############################
#### One feature to screen for anxiety

#### SMILE
Tree_D3 <- rpart(Anxiety ~ SMILE,
                 data = df_train, method = "class",
                 control=list(cp=0.0001, maxdepth=1),
                 parms=list(split="information"))

fancyRpartPlot(Tree_D3, caption = NULL)

Tree_D3_Class_test <- predict(Tree_D3, newdata = df_test, type = "c")
```

```
CM=confusionMatrix(as.factor(Tree_D3_Class_test),
                   df_test$Anxiety,
                   positive = "Anxiety")

Tree_D3_prob_test <- predict(Tree_D3, newdata = df_test,
    type = "prob")[,2]

roc_Tree_D3_test <- roc(df_test$Anxiety ~ Tree_D3_prob_test)

paste(round(CM$overall[1],3), "&", round(CM$byClass[1],3), "&",
      round(CM$byClass[2],3), "&", round(CM$byClass[3],3), "&",
      round(CM$byClass[4],3), "&", round(CM$byClass[7],3), "&",
      round(roc_Tree_D3_test$auc,3))

### SMILE-C
Tree_D3 <- rpart(Anxiety ~ SMILEC,
                 data = df_train, method = "class",
                 control=list(cp=0.0001, maxdepth=1),
                 parms=list(split="information"))

fancyRpartPlot(Tree_D3, caption = NULL)

Tree_D3_Class_test <- predict(Tree_D3, newdata = df_test, type = "c")

CM=confusionMatrix(as.factor(Tree_D3_Class_test),
                   df_test$Anxiety,
                   positive = "Anxiety")

Tree_D3_prob_test <- predict(Tree_D3, newdata = df_test,
    type = "prob")[,2]

roc_Tree_D3_test = roc(df_test$Anxiety ~ Tree_D3_prob_test)

paste(round(CM$overall[1],3), "&", round(CM$byClass[1],3), "&",
      round(CM$byClass[2],3), "&", round(CM$byClass[3],3), "&",
      round(CM$byClass[4],3), "&", round(CM$byClass[7],3), "&",
      round(roc_Tree_D3_test$auc,3))

### SMILE_DN
Tree_D3 <- rpart(Anxiety ~ SMILE_DN,
                 data = df_train, method = "class",
                 control=list(cp=0.0001, maxdepth=1),
                 parms=list(split="information"))

fancyRpartPlot(Tree_D3, caption = NULL)

Tree_D3_Class_test <- predict(Tree_D3, newdata = df_test,
    type = "c")
```

```
CM=confusionMatrix(as.factor(Tree_D3_Class_test),
                   df_test$Anxiety,
                   positive = "Anxiety")

CM$overall
CM$byClass

Tree_D3_prob_test <- predict(Tree_D3, newdata = df_test,
    type = "prob")[,2]

roc_Tree_D3_test = roc(df_test$Anxiety ~ Tree_D3_prob_test)

paste(round(CM$overall[1],3), "&", round(CM$byClass[1],3), "&",
      round(CM$byClass[2],3), "&", round(CM$byClass[3],3), "&",
      round(CM$byClass[4],3), "&", round(CM$byClass[7],3), "&",
      round(roc_Tree_D3_test$auc,3))

### SMILE_SU
Tree_D3 <- rpart(Anxiety ~ SMILE_SU,
                 data = df_train, method = "class",
                 control=list(cp=0.0001, maxdepth=1),
                 parms=list(split="information"))

fancyRpartPlot(Tree_D3, caption = NULL)

Tree_D3_Class_test <- predict(Tree_D3, newdata = df_test, type = "c")

CM=confusionMatrix(as.factor(Tree_D3_Class_test),
                   df_test$Anxiety,
                   positive = "Anxiety")

Tree_D3_prob_test <- predict(Tree_D3, newdata = df_test,
    type = "prob")[,2]

roc_Tree_D3_test = roc(df_test$Anxiety ~ Tree_D3_prob_test)

paste(round(CM$overall[1],3), "&", round(CM$byClass[1],3), "&",
      round(CM$byClass[2],3), "&", round(CM$byClass[3],3), "&",
      round(CM$byClass[4],3), "&", round(CM$byClass[7],3), "&",
      round(roc_Tree_D3_test$auc,3))

### SMILE_PA
Tree_D3 <- rpart(Anxiety ~ SMILE_PA,
                 data = df_train, method = "class",
                 control=list(cp=0.0001, maxdepth=1),
                 parms=list(split="information"))

fancyRpartPlot(Tree_D3, caption = NULL)
```

```
Tree_D3_Class_test <- predict(Tree_D3, newdata = df_test, type = "c")

CM=confusionMatrix(as.factor(Tree_D3_Class_test),
                   df_test$Anxiety,
                   positive = "Anxiety")

Tree_D3_prob_test <- predict(Tree_D3, newdata = df_test,
    type = "prob")[,2]

roc_Tree_D3_test = roc(df_test$Anxiety ~ Tree_D3_prob_test)

paste(round(CM$overall[1],3), "&", round(CM$byClass[1],3), "&",
      round(CM$byClass[2],3), "&", round(CM$byClass[3],3), "&",
      round(CM$byClass[4],3), "&", round(CM$byClass[7],3), "&",
      round(roc_Tree_D3_test$auc,3))

############################SMILE_SM
Tree_D3 <- rpart(Anxiety ~ SMILE_SM,
                 data = df_train, method = "class",
                 control=list(cp=0.0001, maxdepth=1),
                 parms=list(split="information"))

fancyRpartPlot(Tree_D3, caption = NULL)

Tree_D3_Class_test <- predict(Tree_D3, newdata = df_test, type = "c")

CM=confusionMatrix(as.factor(Tree_D3_Class_test),
                   df_test$Anxiety,
                   positive = "Anxiety")

Tree_D3_prob_test <- predict(Tree_D3, newdata = df_test,
    type = "prob")[,2]

roc_Tree_D3_test = roc(df_test$Anxiety ~ Tree_D3_prob_test)

paste(round(CM$overall[1],3), "&", round(CM$byClass[1],3), "&",
      round(CM$byClass[2],3), "&", round(CM$byClass[3],3), "&",
      round(CM$byClass[4],3), "&", round(CM$byClass[7],3), "&",
      round(roc_Tree_D3_test$auc,3))

### SMILE_RS
Tree_D3 <- rpart(Anxiety ~ SMILE_RS,
                 data = df_train, method = "class",
                 control=list(cp=0.0001, maxdepth=1),
                 parms=list(split="information"))

fancyRpartPlot(Tree_D3, caption = NULL)

Tree_D3_Class_test <- predict(Tree_D3, newdata = df_test, type = "c")
```

```
CM=confusionMatrix(as.factor(Tree_D3_Class_test),
                   df_test$Anxiety,
                   positive = "Anxiety")

Tree_D3_prob_test <- predict(Tree_D3, newdata = df_test,
    type = "prob")[,2]

roc_Tree_D3_test = roc(df_test$Anxiety ~ Tree_D3_prob_test)

paste(round(CM$overall[1],3), "&", round(CM$byClass[1],3), "&",
      round(CM$byClass[2],3), "&", round(CM$byClass[3],3), "&",
      round(CM$byClass[4],3), "&", round(CM$byClass[7],3), "&",
      round(roc_Tree_D3_test$auc,3))

### SMILE_SS
Tree_D3 <- rpart(Anxiety ~ SMILE_SS,
                 data = df_train, method = "class",
                 control=list(cp=0.0001, maxdepth=1),
                 parms=list(split="information"))

fancyRpartPlot(Tree_D3, caption = NULL)

Tree_D3_Class_test <- predict(Tree_D3, newdata = df_test, type = "c")

CM=confusionMatrix(as.factor(Tree_D3_Class_test),
                   df_test$Anxiety,
                   positive = "Anxiety")

Tree_D3_prob_test <- predict(Tree_D3, newdata = df_test,
    type = "prob")[,2]

roc_Tree_D3_test = roc(df_test$Anxiety ~ Tree_D3_prob_test)

paste(round(CM$overall[1],3), "&", round(CM$byClass[1],3), "&",
      round(CM$byClass[2],3), "&", round(CM$byClass[3],3), "&",
      round(CM$byClass[4],3), "&", round(CM$byClass[7],3), "&",
      round(roc_Tree_D3_test$auc,3))

### SMILE_EE
Tree_D3 <- rpart(Anxiety ~ SMILE_EE,
                 data = df_train, method = "class",
                 control=list(cp=0.0001, maxdepth=1),
                 parms=list(split="information"))

fancyRpartPlot(Tree_D3, caption = NULL)

Tree_D3_Class_test <- predict(Tree_D3, newdata = df_test, type = "c")
```

```
CM=confusionMatrix(as.factor(Tree_D3_Class_test),
                   df_test$Anxiety,
                   positive = "Anxiety")

Tree_D3_prob_test <- predict(Tree_D3, newdata = df_test,
    type = "prob")[,2]

roc_Tree_D3_test = roc(df_test$Anxiety ~ Tree_D3_prob_test)

paste(round(CM$overall[1],3), "&", round(CM$byClass[1],3), "&",
      round(CM$byClass[2],3), "&", round(CM$byClass[3],3), "&",
      round(CM$byClass[4],3), "&", round(CM$byClass[7],3), "&",
      round(roc_Tree_D3_test$auc,3))

############################
#### One feature to screen for depression

### SMILE
Tree_D3 <- rpart(Depression ~ SMILE,
                 data = df_train, method = "class",
                 control=list(cp=0.0001, maxdepth=1),
                 parms=list(split="information"))

fancyRpartPlot(Tree_D3, caption = NULL)

Tree_D3_Class_test <- predict(Tree_D3, newdata = df_test, type = "c")

CM=confusionMatrix(as.factor(Tree_D3_Class_test),
                   df_test$Depression,
                   positive = "Depression")

Tree_D3_prob_test <- predict(Tree_D3, newdata = df_test,
    type = "prob")[,2]

roc_Tree_D3_test = roc(df_test$Depression ~ Tree_D3_prob_test)

paste(round(CM$overall[1],3), "&", round(CM$byClass[1],3), "&",
      round(CM$byClass[2],3), "&", round(CM$byClass[3],3), "&",
      round(CM$byClass[4],3), "&", round(CM$byClass[7],3), "&",
      round(roc_Tree_D3_test$auc,3))

### SMILE-C
Tree_D3 <- rpart(Depression ~ SMILEC,
                 data = df_train, method = "class",
                 control=list(cp=0.0001, maxdepth=1),
                 parms=list(split="information"))

fancyRpartPlot(Tree_D3, caption = NULL)
```

```
Tree_D3_Class_test <- predict(Tree_D3, newdata = df_test, type = "c")

CM=confusionMatrix(as.factor(Tree_D3_Class_test),
                   df_test$Depression,
                   positive = "Depression")

Tree_D3_prob_test <- predict(Tree_D3, newdata = df_test,
    type = "prob")[,2]

roc_Tree_D3_test = roc(df_test$Depression ~ Tree_D3_prob_test)

paste(round(CM$overall[1],3), "&", round(CM$byClass[1],3), "&",
      round(CM$byClass[2],3), "&", round(CM$byClass[3],3), "&",
      round(CM$byClass[4],3), "&", round(CM$byClass[7],3), "&",
      round(roc_Tree_D3_test$auc,3))

### SMILE_DN
Tree_D3 <- rpart(Depression ~ SMILE_DN,
                 data = df_train, method = "class",
                 control=list(cp=0.0001, maxdepth=1),
                 parms=list(split="information"))

fancyRpartPlot(Tree_D3, caption = NULL)

Tree_D3_Class_test <- predict(Tree_D3, newdata = df_test, type = "c")

CM=confusionMatrix(as.factor(Tree_D3_Class_test),
                   df_test$Depression,
                   positive = "Depression")

CM$overall
CM$byClass
Tree_D3_prob_test <- predict(Tree_D3, newdata = df_test,
    type = "prob")[,2]

roc_Tree_D3_test = roc(df_test$Depression ~ Tree_D3_prob_test)

paste(round(CM$overall[1],3), "&", round(CM$byClass[1],3), "&",
      round(CM$byClass[2],3), "&", round(CM$byClass[3],3), "&",
      round(CM$byClass[4],3), "&", round(CM$byClass[7],3), "&",
      round(roc_Tree_D3_test$auc,3))

### SMILE_SU
Tree_D3 <- rpart(Depression ~ SMILE_SU,
                 data = df_train, method = "class",
                 control=list(cp=0.0001, maxdepth=1),
                 parms=list(split="information"))

fancyRpartPlot(Tree_D3, caption = NULL)
```

```
Tree_D3_Class_test <- predict(Tree_D3, newdata = df_test, type = "c")

CM=confusionMatrix(as.factor(Tree_D3_Class_test),
                   df_test$Depression,
                   positive = "Depression")

Tree_D3_prob_test <- predict(Tree_D3, newdata = df_test,
    type = "prob")[,2]

roc_Tree_D3_test = roc(df_test$Depression ~ Tree_D3_prob_test)

paste(round(CM$overall[1],3), "&", round(CM$byClass[1],3), "&",
      round(CM$byClass[2],3), "&", round(CM$byClass[3],3), "&",
      round(CM$byClass[4],3), "&", round(CM$byClass[7],3), "&",
      round(roc_Tree_D3_test$auc,3))

### SMILE_PA
Tree_D3 <- rpart(Depression ~ SMILE_PA,
                 data = df_train, method = "class",
                 control=list(cp=0.0001, maxdepth=1),
                 parms=list(split="information"))

fancyRpartPlot(Tree_D3, caption = NULL)

Tree_D3_Class_test <- predict(Tree_D3, newdata = df_test, type = "c")

CM=confusionMatrix(as.factor(Tree_D3_Class_test),
                   df_test$Depression,
                   positive = "Depression")

Tree_D3_prob_test <- predict(Tree_D3, newdata = df_test,
    type = "prob")[,2]

roc_Tree_D3_test = roc(df_test$Depression ~ Tree_D3_prob_test)

paste(round(CM$overall[1],3), "&", round(CM$byClass[1],3), "&",
      round(CM$byClass[2],3), "&", round(CM$byClass[3],3), "&",
      round(CM$byClass[4],3), "&", round(CM$byClass[7],3), "&",
      round(roc_Tree_D3_test$auc,3))

###SMILE_SM
Tree_D3 <- rpart(Depression ~ SMILE_SM,
                 data = df_train, method = "class",
                 control=list(cp=0.0001, maxdepth=1),
                 parms=list(split="information"))

fancyRpartPlot(Tree_D3, caption = NULL)
```

```
Tree_D3_Class_test <- predict(Tree_D3, newdata = df_test, type = "c")

CM=confusionMatrix(as.factor(Tree_D3_Class_test),
                   df_test$Depression,
                   positive = "Depression")

Tree_D3_prob_test <- predict(Tree_D3, newdata = df_test,
    type = "prob")[,2]

roc_Tree_D3_test = roc(df_test$Depression ~ Tree_D3_prob_test)

paste(round(CM$overall[1],3), "&", round(CM$byClass[1],3), "&",
      round(CM$byClass[2],3), "&", round(CM$byClass[3],3), "&",
      round(CM$byClass[4],3), "&", round(CM$byClass[7],3), "&",
      round(roc_Tree_D3_test$auc,3))

### SMILE_RS
Tree_D3 <- rpart(Depression ~ SMILE_RS,
                 data = df_train, method = "class",
                 control=list(cp=0.0001, maxdepth=1),
                 parms=list(split="information"))

fancyRpartPlot(Tree_D3, caption = NULL)

Tree_D3_Class_test <- predict(Tree_D3, newdata = df_test, type = "c")

CM=confusionMatrix(as.factor(Tree_D3_Class_test),
                   df_test$Depression,
                   positive = "Depression")

Tree_D3_prob_test <- predict(Tree_D3, newdata = df_test,
    type = "prob")[,2]

roc_Tree_D3_test = roc(df_test$Depression ~ Tree_D3_prob_test)

paste(round(CM$overall[1],3), "&", round(CM$byClass[1],3), "&",
      round(CM$byClass[2],3), "&", round(CM$byClass[3],3), "&",
      round(CM$byClass[4],3), "&", round(CM$byClass[7],3), "&",
      round(roc_Tree_D3_test$auc,3))

### SMILE_SS
Tree_D3 <- rpart(Depression ~ SMILE_SS,
                 data = df_train, method = "class",
                 control=list(cp=0.0001, maxdepth=1),
                 parms=list(split="information"))

fancyRpartPlot(Tree_D3, caption = NULL)

Tree_D3_Class_test <- predict(Tree_D3, newdata = df_test, type = "c")
```

```
CM=confusionMatrix(as.factor(Tree_D3_Class_test),
                   df_test$Depression,
                   positive = "Depression")

Tree_D3_prob_test <- predict(Tree_D3, newdata = df_test,
    type = "prob")[,2]

roc_Tree_D3_test = roc(df_test$Depression ~ Tree_D3_prob_test)

paste(round(CM$overall[1],3), "&", round(CM$byClass[1],3), "&",
      round(CM$byClass[2],3), "&", round(CM$byClass[3],3), "&",
      round(CM$byClass[4],3), "&", round(CM$byClass[7],3), "&",
      round(roc_Tree_D3_test$auc,3))

### SMILE_EE
Tree_D3 <- rpart(Depression ~ SMILE_EE,
                 data = df_train, method = "class",
                 control=list(cp=0.0001, maxdepth=1),
                 parms=list(split="information"))

fancyRpartPlot(Tree_D3, caption = NULL)

Tree_D3_Class_test <- predict(Tree_D3, newdata = df_test, type = "c")

CM=confusionMatrix(as.factor(Tree_D3_Class_test),
                   df_test$Depression,
                   positive = "Depression")

Tree_D3_prob_test <- predict(Tree_D3, newdata = df_test,
    type = "prob")[,2]

roc_Tree_D3_test = roc(df_test$Depression ~ Tree_D3_prob_test)
paste(round(CM$overall[1],3), "&", round(CM$byClass[1],3), "&",
      round(CM$byClass[2],3), "&", round(CM$byClass[3],3), "&",
      round(CM$byClass[4],3), "&", round(CM$byClass[7],3), "&",
      round(roc_Tree_D3_test$auc,3))

############################
#### One feature to screen for Anxiety and Depression

### SMILE
Tree_D3 <- rpart(Anxiety_Depression ~ SMILE,
                 data = df_train, method = "class",
                 control=list(cp=0.0001, maxdepth=1),
                 parms=list(split="information"))

fancyRpartPlot(Tree_D3, caption = NULL)
```

```
Tree_D3_Class_test <- predict(Tree_D3, newdata = df_test, type = "c")

CM=confusionMatrix(as.factor(Tree_D3_Class_test),
                   df_test$Anxiety_Depression,
                   positive = "Anxiety and Depression")

Tree_D3_prob_test <- predict(Tree_D3, newdata = df_test,
    type = "prob")[,2]

roc_Tree_D3_test = roc(df_test$Anxiety_Depression ~ Tree_D3_prob_test)

paste(round(CM$overall[1],3), "&", round(CM$byClass[1],3), "&",
      round(CM$byClass[2],3), "&", round(CM$byClass[3],3), "&",
      round(CM$byClass[4],3), "&", round(CM$byClass[7],3), "&",
      round(roc_Tree_D3_test$auc,3))

### SMILE-C
Tree_D3 <- rpart(Anxiety_Depression ~ SMILEC,
                 data = df_train, method = "class",
                 control=list(cp=0.0001, maxdepth=1),
                 parms=list(split="information"))

fancyRpartPlot(Tree_D3, caption = NULL)

Tree_D3_Class_test <- predict(Tree_D3, newdata = df_test, type = "c")

CM=confusionMatrix(as.factor(Tree_D3_Class_test),
                   df_test$Anxiety_Depression,
                   positive = "Anxiety and Depression")

Tree_D3_prob_test <- predict(Tree_D3, newdata = df_test,
    type = "prob")[,2]

roc_Tree_D3_test = roc(df_test$Anxiety_Depression ~ Tree_D3_prob_test)

paste(round(CM$overall[1],3), "&", round(CM$byClass[1],3), "&",
      round(CM$byClass[2],3), "&", round(CM$byClass[3],3), "&",
      round(CM$byClass[4],3), "&", round(CM$byClass[7],3), "&",
      round(roc_Tree_D3_test$auc,3))

### SMILE_DN
Tree_D3 <- rpart(Anxiety_Depression ~ SMILE_DN,
                 data = df_train, method = "class",
                 control=list(cp=0.0001, maxdepth=1),
                 parms=list(split="information"))

fancyRpartPlot(Tree_D3, caption = NULL)

Tree_D3_Class_test <- predict(Tree_D3, newdata = df_test, type = "c")
```

```
CM=confusionMatrix(as.factor(Tree_D3_Class_test),
                   df_test$Anxiety_Depression,
                   positive = "Anxiety and Depression")

CM$overall
CM$byClass

Tree_D3_prob_test <- predict(Tree_D3, newdata = df_test,
    type = "prob")[,2]

roc_Tree_D3_test = roc(df_test$Anxiety_Depression ~ Tree_D3_prob_test)

paste(round(CM$overall[1],3), "&", round(CM$byClass[1],3), "&",
      round(CM$byClass[2],3), "&", round(CM$byClass[3],3), "&",
      round(CM$byClass[4],3), "&", round(CM$byClass[7],3), "&",
      round(roc_Tree_D3_test$auc,3))

### SMILE_SU
Tree_D3 <- rpart(Anxiety_Depression ~ SMILE_SU,
                 data = df_train, method = "class",
                 control=list(cp=0.0001, maxdepth=1),
                 parms=list(split="information"))

fancyRpartPlot(Tree_D3, caption = NULL)

Tree_D3_Class_test <- predict(Tree_D3, newdata = df_test, type = "c")

CM=confusionMatrix(as.factor(Tree_D3_Class_test),
                   df_test$Anxiety_Depression,
                   positive = "Anxiety and Depression")

Tree_D3_prob_test <- predict(Tree_D3, newdata = df_test,
    type = "prob")[,2]

roc_Tree_D3_test = roc(df_test$Anxiety_Depression ~ Tree_D3_prob_test)

paste(round(CM$overall[1],3), "&", round(CM$byClass[1],3), "&",
      round(CM$byClass[2],3), "&", round(CM$byClass[3],3), "&",
      round(CM$byClass[4],3), "&", round(CM$byClass[7],3), "&",
      round(roc_Tree_D3_test$auc,3))

### SMILE_PA
Tree_D3 <- rpart(Anxiety_Depression ~ SMILE_PA,
                 data = df_train, method = "class",
                 control=list(cp=0.0001, maxdepth=1),
                 parms=list(split="information"))

fancyRpartPlot(Tree_D3, caption = NULL)
```

```
Tree_D3_Class_test <- predict(Tree_D3, newdata = df_test, type = "c")

CM=confusionMatrix(as.factor(Tree_D3_Class_test),
                   df_test$Anxiety_Depression,
                   positive = "Anxiety and Depression")

Tree_D3_prob_test <- predict(Tree_D3, newdata = df_test,
    type = "prob")[,2]

roc_Tree_D3_test = roc(df_test$Anxiety_Depression ~ Tree_D3_prob_test)

paste(round(CM$overall[1],3), "&", round(CM$byClass[1],3), "&",
      round(CM$byClass[2],3), "&", round(CM$byClass[3],3), "&",
      round(CM$byClass[4],3), "&", round(CM$byClass[7],3), "&",
      round(roc_Tree_D3_test$auc,3))

### SMILE_SM
Tree_D3 <- rpart(Anxiety_Depression ~ SMILE_SM,
                 data = df_train, method = "class",
                 control=list(cp=0.0001, maxdepth=1),
                 parms=list(split="information"))

fancyRpartPlot(Tree_D3, caption = NULL)

Tree_D3_Class_test <- predict(Tree_D3, newdata = df_test, type = "c")

CM=confusionMatrix(as.factor(Tree_D3_Class_test),
                   df_test$Anxiety_Depression,
                   positive = "Anxiety and Depression")

Tree_D3_prob_test <- predict(Tree_D3, newdata = df_test,
    type = "prob")[,2]

roc_Tree_D3_test = roc(df_test$Anxiety_Depression ~ Tree_D3_prob_test)

paste(round(CM$overall[1],3), "&", round(CM$byClass[1],3), "&",
      round(CM$byClass[2],3), "&", round(CM$byClass[3],3), "&",
      round(CM$byClass[4],3), "&", round(CM$byClass[7],3), "&",
      round(roc_Tree_D3_test$auc,3))

### SMILE_RS
Tree_D3 <- rpart(Anxiety_Depression ~ SMILE_RS,
                 data = df_train, method = "class",
                 control=list(cp=0.0001, maxdepth=1),
                 parms=list(split="information"))

fancyRpartPlot(Tree_D3, caption = NULL)
```

```
Tree_D3_Class_test <- predict(Tree_D3, newdata = df_test, type = "c")

CM=confusionMatrix(as.factor(Tree_D3_Class_test),
                   df_test$Anxiety_Depression,
                   positive = "Anxiety and Depression")

Tree_D3_prob_test <- predict(Tree_D3, newdata = df_test,
    type = "prob")[,2]

roc_Tree_D3_test = roc(df_test$Anxiety_Depression ~ Tree_D3_prob_test)

paste(round(CM$overall[1],3), "&", round(CM$byClass[1],3), "&",
      round(CM$byClass[2],3), "&", round(CM$byClass[3],3), "&",
      round(CM$byClass[4],3), "&", round(CM$byClass[7],3), "&",
      round(roc_Tree_D3_test$auc,3))

### SMILE_SS
Tree_D3 <- rpart(Anxiety_Depression ~ SMILE_SS,
                 data = df_train, method = "class",
                 control=list(cp=0.0001, maxdepth=1),
                 parms=list(split="information"))

fancyRpartPlot(Tree_D3, caption = NULL)

Tree_D3_Class_test <- predict(Tree_D3, newdata = df_test, type = "c")

CM=confusionMatrix(as.factor(Tree_D3_Class_test),
                   df_test$Anxiety_Depression,
                   positive = "Anxiety and Depression")

Tree_D3_prob_test <- predict(Tree_D3, newdata = df_test,
    type = "prob")[,2]

roc_Tree_D3_test = roc(df_test$Anxiety_Depression ~ Tree_D3_prob_test)

paste(round(CM$overall[1],3), "&", round(CM$byClass[1],3), "&",
      round(CM$byClass[2],3), "&", round(CM$byClass[3],3), "&",
      round(CM$byClass[4],3), "&", round(CM$byClass[7],3), "&",
      round(roc_Tree_D3_test$auc,3))

### SMILE_EE
Tree_D3 <- rpart(Anxiety_Depression ~ SMILE_EE,
                 data = df_train, method = "class",
                 control=list(cp=0.0001, maxdepth=1),
                 parms=list(split="information"))

fancyRpartPlot(Tree_D3, caption = NULL)

Tree_D3_Class_test <- predict(Tree_D3, newdata = df_test, type = "c")
```

```
CM=confusionMatrix(as.factor(Tree_D3_Class_test),
                   df_test$Anxiety_Depression,
                   positive = "Anxiety and Depression")

Tree_D3_prob_test <- predict(Tree_D3, newdata = df_test,
type = "prob")[,2]

roc_Tree_D3_test = roc(df_test$Anxiety_Depression ~ Tree_D3_prob_test)

paste(round(CM$overall[1],3), "&", round(CM$byClass[1],3), "&",
      round(CM$byClass[2],3), "&", round(CM$byClass[3],3), "&",
      round(CM$byClass[4],3), "&", round(CM$byClass[7],3), "&",
      round(roc_Tree_D3_test$auc,3))


##########################################################
##### Multivariate analysis to screen for anxiety

##########################################################
# Decision trees - whithout oversampling

# SMILE
Tree_S <- rpart(Anxiety ~ SMILE + Gender + Age + Course +
        School + Displaced,
                 data = df_train, method = "class",
                 control=list(cp=0.0001),
                              parms=list(split="information"))

rpart.plot(Tree_S)
fancyRpartPlot(Tree_S, caption = NULL)

Tree_S_prob_train <- predict(Tree_S, newdata = df_train,
    type = "prob")[,1]
Tree_S_prob_test <- predict(Tree_S, newdata = df_test,
    type = "prob")[,1]
roc_Tree_S_train = roc(df_train$Anxiety ~ Tree_S_prob_train)
roc_Tree_S_test = roc(df_test$Anxiety ~ Tree_S_prob_test)
roc.test(roc_Tree_S_train, roc_Tree_S_test)
################################
Tree_S <- rpart(Anxiety ~ SMILE + Gender + Age + Course +
    School + Displaced,
                data = df_train, method = "class",
                control=list(cp=0.0001,
                             maxdepth=4),
                             parms=list(split="information"))

Tree_S_prob_train <- predict(Tree_S, newdata = df_train,
    type = "prob")[,1]
```

```
Tree_S_prob_test <- predict(Tree_S, newdata = df_test,
    type = "prob")[,1]
roc_Tree_S_train = roc(df_train$Anxiety ~ Tree_S_prob_train)
r1A = roc_Tree_S_test = roc(df_test$Anxiety ~ Tree_S_prob_test)
roc.test(roc_Tree_S_train, roc_Tree_S_test)

fancyRpartPlot(Tree_S, caption = NULL)

plot(roc_Tree_S_train, print.auc = TRUE, col = 2, lwd =3,
     print.auc.y=0.98, print.auc.x=0.98,
     auc.polygon = TRUE, max.auc.polygon = TRUE);
plot(roc_Tree_S_test, add=TRUE , print.auc = TRUE, col = 3, lwd =3,
     print.auc.y=0.90, print.auc.x=0.98)
legend(0.37,0.25, c("Train ","Test "),
       col = c(2,3), lwd =2)

Tree_S_Class_test <- predict(Tree_S, newdata = df_test, type = "c")

CM=confusionMatrix(as.factor(Tree_S_Class_test),
                   df_test$Anxiety,
                   positive = "Anxiety")
paste(round(CM$overall[1],3), "&", round(CM$byClass[1],3), "&",
      round(CM$byClass[2],3), "&", round(CM$byClass[3],3), "&",
      round(CM$byClass[4],3), "&", round(CM$byClass[7],3), "&",
      round(roc_Tree_S_test$auc,3))

DI <- data.frame(imp = Tree_S$variable.importance)
DI$variable <- names(Tree_S$variable.importance)
ggplot(DI) +
  geom_col(aes(x = variable, y = imp),
           col = "black", show.legend = F) +
  labs(x = NULL,
       y = "Feature Importance",
       title =NULL) +
  coord_flip() +
  scale_fill_grey() +
  theme_bw()

#########################################################
# SMILE-C
Tree_S <- rpart(Anxiety ~ SMILEC + Gender + Age + Course +
    School + Displaced,
                data = df_train, method = "class",
                control=list(cp=0.0001),
                parms=list(split="information"))

fancyRpartPlot(Tree_S, caption = NULL)

Tree_S_prob_train <- predict(Tree_S, newdata = df_train,
```

```
    type = "prob")[,1]
Tree_S_prob_test <- predict(Tree_S, newdata = df_test,
    type = "prob")[,1]
roc_Tree_S_train = roc(df_train$Anxiety ~ Tree_S_prob_train)
roc_Tree_S_test = roc(df_test$Anxiety ~ Tree_S_prob_test)
roc.test(roc_Tree_S_train, roc_Tree_S_test)
#####################################
Tree_S <- rpart(Anxiety ~ SMILEC + Gender + Age + Course +
    School + Displaced,
                data = df_train, method = "class",
                control=list(cp=0.0001,
                             maxdepth=3),
                             parms=list(split="information"))

Tree_S_prob_train <- predict(Tree_S, newdata = df_train,
    type = "prob")[,2]
Tree_S_prob_test <- predict(Tree_S, newdata = df_test,
    type = "prob")[,2]
roc_Tree_S_train = roc(df_train$Anxiety ~ Tree_S_prob_train)
r2A = roc_Tree_S_test = roc(df_test$Anxiety ~ Tree_S_prob_test)
roc.test(roc_Tree_S_train, roc_Tree_S_test)

fancyRpartPlot(Tree_S, caption = NULL)

plot(roc_Tree_S_train, print.auc = TRUE, col = 2, lwd =3,
     print.auc.y=0.98, print.auc.x=0.98,
     auc.polygon = TRUE, max.auc.polygon = TRUE);
plot(roc_Tree_S_test, add=TRUE , print.auc = TRUE, col = 3, lwd =3,
     print.auc.y=0.90, print.auc.x=0.98)
legend(0.4,0.28, c("Train ","Test "),
       col = c(2,3), lwd =2)

Tree_S_Class_test <- predict(Tree_S, newdata = df_test, type = "c")

CM=confusionMatrix(as.factor(Tree_S_Class_test),
                   df_test$Anxiety,
                   positive = "Anxiety")
paste(round(CM$overall[1],3), "&", round(CM$byClass[1],3), "&",
      round(CM$byClass[2],3), "&", round(CM$byClass[3],3), "&",
      round(CM$byClass[4],3), "&", round(CM$byClass[7],3), "&",
      round(roc_Tree_S_test$auc,3))

DI <- data.frame(imp = Tree_S$variable.importance)
DI$variable <- names(Tree_S$variable.importance)
ggplot(DI) +
  geom_col(aes(x = variable, y = imp),
           col = "black", show.legend = F) +
  labs(x = NULL,
       y = "Feature Importance",
```

```
       title =NULL) +
  coord_flip() +
  scale_fill_grey() +
  theme_bw()

#########################################################
# Domains
Tree_S <- rpart(Anxiety ~ SMILE_DN + SMILE_SU + SMILE_PA + SMILE_SM +
                  SMILE_RS + SMILE_SS + SMILE_EE + Gender + Age +
                  Course + School + Displaced,
                data = df_train, method = "class",
                control=list(cp=0.0001),
                parms=list(split="information"))

rpart.plot(Tree_S)
fancyRpartPlot(Tree_S, caption = NULL)

Tree_S_prob_train <- predict(Tree_S, newdata = df_train,
    type = "prob")[,1]
Tree_S_prob_test <- predict(Tree_S, newdata = df_test,
    type = "prob")[,1]
roc_Tree_S_train = roc(df_train$Anxiety ~ Tree_S_prob_train)
roc_Tree_S_test = roc(df_test$Anxiety ~ Tree_S_prob_test)
roc.test(roc_Tree_S_train, roc_Tree_S_test)
#####################################
Tree_S <- rpart(Anxiety ~ SMILE_DN + SMILE_SU + SMILE_PA + SMILE_SM +
                  SMILE_RS + SMILE_SS + SMILE_EE + Gender + Age +
                  Course + School + Displaced,
                data = df_train, method = "class",
                control=list(cp=0.0001,
                             maxdepth=5),
                             parms=list(split="information"))

Tree_S_prob_train <- predict(Tree_S, newdata = df_train,
    type = "prob")[,1]
Tree_S_prob_test <- predict(Tree_S, newdata = df_test,
    type = "prob")[,1]
roc_Tree_S_train = roc(df_train$Anxiety ~ Tree_S_prob_train)
r3A = roc_Tree_S_test = roc(df_test$Anxiety ~ Tree_S_prob_test)
roc.test(roc_Tree_S_train, roc_Tree_S_test)

fancyRpartPlot(Tree_S, caption = NULL)

plot(roc_Tree_S_train, print.auc = TRUE, col = 2, lwd =3,
     print.auc.y=0.98, print.auc.x=0.98,
     auc.polygon = TRUE, max.auc.polygon = TRUE);
plot(roc_Tree_S_test, add=TRUE , print.auc = TRUE, col = 3, lwd =3,
     print.auc.y=0.90, print.auc.x=0.98)
legend(0.40,0.28, c("Train ","Test "),
```

```
       col = c(2,3), lwd =2)

Tree_S_Class_test <- predict(Tree_S, newdata = df_test, type = "c")

CM=confusionMatrix(as.factor(Tree_S_Class_test),
                   df_test$Anxiety,
                   positive = "Anxiety")

paste(round(CM$overall[1],3), "&", round(CM$byClass[1],3), "&",
      round(CM$byClass[2],3), "&", round(CM$byClass[3],3), "&",
      round(CM$byClass[4],3), "&", round(CM$byClass[7],3), "&",
      round(roc_Tree_S_test$auc,3))

DI <- data.frame(imp = Tree_S$variable.importance)
DI$variable <- names(Tree_S$variable.importance)
ggplot(DI) +
  geom_col(aes(x = variable, y = imp),
           col = "black", show.legend = F) +
  labs(x = NULL,
       y = "Feature Importance",
       title =NULL) +
  coord_flip() +
  scale_fill_grey() +
  theme_bw()

#########################################################
# Decision trees - whith oversampling

#########################################################
### Oversampling
table(df_train$Anxiety)
set.seed(123)
df_train_O <- RandOverClassif(Anxiety~., df_train, "balance")
table(df_train_O$Anxiety)
#########################################################
# SMILE
Tree_S_O <- rpart(Anxiety ~ SMILE + Gender + Age + Course +
    School + Displaced,
                data = df_train_O, method = "class",
                control=list(cp=0.0001),
                parms=list(split="information"))

rpart.plot(Tree_S_O)
fancyRpartPlot(Tree_S_O, caption = NULL)

Tree_S_O_prob_train <- predict(Tree_S_O, newdata = df_train_O,
    type = "prob")[,1]
Tree_S_O_prob_test <- predict(Tree_S_O, newdata = df_test,
    type = "prob")[,1]
```

```
roc_Tree_S_O_train = roc(df_train_O$Anxiety ~ Tree_S_O_prob_train)
roc_Tree_S_O_test = roc(df_test$Anxiety ~ Tree_S_O_prob_test)
roc.test(roc_Tree_S_O_train, roc_Tree_S_O_test)
#####################################
Tree_S_O <- rpart(Anxiety ~ SMILE + Gender + Age + Course +
    School + Displaced,
                data = df_train_O, method = "class",
                control=list(cp=0.0001,
                             maxdepth=5),
                             parms=list(split="information"))
Tree_S_O_prob_train <- predict(Tree_S_O, newdata = df_train_O,
    type = "prob")[,1]
Tree_S_O_prob_test <- predict(Tree_S_O, newdata = df_test,
    type = "prob")[,1]
roc_Tree_S_O_train = roc(df_train_O$Anxiety ~ Tree_S_O_prob_train)
r4A = roc_Tree_S_O_test = roc(df_test$Anxiety ~ Tree_S_O_prob_test)
roc.test(roc_Tree_S_O_train, roc_Tree_S_O_test)

fancyRpartPlot(Tree_S_O, caption = NULL)

plot(roc_Tree_S_O_train, print.auc = TRUE, col = 2, lwd =3,
     print.auc.y=0.98, print.auc.x=0.98,
     auc.polygon = TRUE, max.auc.polygon = TRUE);
plot(roc_Tree_S_O_test, add=TRUE , print.auc = TRUE, col = 3,
    lwd =3, print.auc.y=0.90, print.auc.x=0.98)
legend(0.40,0.27, c("Train ","Test "),
       col = c(2,3), lwd =2)

Tree_S_O_Class_test <- predict(Tree_S_O, newdata = df_test,
    type = "c")

CM=confusionMatrix(as.factor(Tree_S_O_Class_test),
                   df_test$Anxiety,
                   positive = "Anxiety")

paste(round(CM$overall[1],3), "&", round(CM$byClass[1],3), "&",
      round(CM$byClass[2],3), "&", round(CM$byClass[3],3), "&",
      round(CM$byClass[4],3), "&", round(CM$byClass[7],3), "&",
      round(roc_Tree_S_O_test$auc,3))

DI <- data.frame(imp = Tree_S_O$variable.importance)
DI$variable <- names(Tree_S_O$variable.importance)
ggplot(DI) +
  geom_col(aes(x = variable, y = imp),
           col = "black", show.legend = F) +
  labs(x = NULL,
       y = "Feature Importance",
       title =NULL) +
  coord_flip() +
```

```
  scale_fill_grey() +
  theme_bw()

#########################################################
# SMILE-C
Tree_S_O <- rpart(Anxiety ~ SMILEC + Gender + Age + Course +
    School + Displaced,
                data = df_train_O, method = "class",
                control=list(cp=0.0001),
                parms=list(split="information"))

fancyRpartPlot(Tree_S_O, caption = NULL)

Tree_S_O_prob_train <- predict(Tree_S_O, newdata = df_train_O,
    type = "prob")[,1]
Tree_S_O_prob_test <- predict(Tree_S_O, newdata = df_test,
    type = "prob")[,1]
roc_Tree_S_O_train = roc(df_train_O$Anxiety ~ Tree_S_O_prob_train)
roc_Tree_S_O_test = roc(df_test$Anxiety ~ Tree_S_O_prob_test)
roc.test(roc_Tree_S_O_train, roc_Tree_S_O_test)
#####################################
Tree_S_O <- rpart(Anxiety ~ SMILEC + Gender + Age + Course +
    School + Displaced,
                data = df_train_O, method = "class",
                control=list(cp=0.0001,
                             maxdepth=4),
                             parms=list(split="information"))

Tree_S_O_prob_train <- predict(Tree_S_O, newdata = df_train_O,
    type = "prob")[,2]
Tree_S_O_prob_test <- predict(Tree_S_O, newdata = df_test,
    type = "prob")[,2]
roc_Tree_S_O_train = roc(df_train_O$Anxiety ~ Tree_S_O_prob_train)
r5A = roc_Tree_S_O_test = roc(df_test$Anxiety ~ Tree_S_O_prob_test)
roc.test(roc_Tree_S_O_train, roc_Tree_S_O_test)

fancyRpartPlot(Tree_S_O, caption = NULL)

plot(roc_Tree_S_O_train, print.auc = TRUE, col = 2, lwd =3,
     print.auc.y=0.98, print.auc.x=0.98,
     auc.polygon = TRUE, max.auc.polygon = TRUE);
plot(roc_Tree_S_O_test, add=TRUE , print.auc = TRUE, col = 3,
    lwd =3, print.auc.y=0.90, print.auc.x=0.98)
legend(0.40,0.28, c("Train ","Test "),
       col = c(2,3), lwd =2)

Tree_S_O_Class_test <- predict(Tree_S_O, newdata = df_test,
    type = "c")
```

```
CM=confusionMatrix(as.factor(Tree_S_O_Class_test),
                   df_test$Anxiety,
                   positive = "Anxiety")

paste(round(CM$overall[1],3), "&", round(CM$byClass[1],3), "&",
      round(CM$byClass[2],3), "&", round(CM$byClass[3],3), "&",
      round(CM$byClass[4],3), "&", round(CM$byClass[7],3), "&",
      round(roc_Tree_S_O_test$auc,3))

DI <- data.frame(imp = Tree_S_O$variable.importance)
DI$variable <- names(Tree_S_O$variable.importance)
ggplot(DI) +
  geom_col(aes(x = variable, y = imp),
           col = "black", show.legend = F) +
  labs(x = NULL,
       y = "Feature Importance",
       title =NULL) +
  coord_flip() +
  scale_fill_grey() +
  theme_bw()

#########################################################
# Domains
Tree_S_O <- rpart(Anxiety ~ SMILE_DN + SMILE_SU + SMILE_PA + SMILE_SM +
                  SMILE_RS + SMILE_SS + SMILE_EE + Gender + Age +
                  Course + School + Displaced,
                data = df_train_O, method = "class",
                control=list(cp=0.0001),
                parms=list(split="information"))

rpart.plot(Tree_S_O)
fancyRpartPlot(Tree_S_O, caption = NULL)

Tree_S_O_prob_train <- predict(Tree_S_O, newdata = df_train_O,
    type = "prob")[,1]
Tree_S_O_prob_test <- predict(Tree_S_O, newdata = df_test,
    type = "prob")[,1]
roc_Tree_S_O_train = roc(df_train_O$Anxiety ~ Tree_S_O_prob_train)
roc_Tree_S_O_test = roc(df_test$Anxiety ~ Tree_S_O_prob_test)
roc.test(roc_Tree_S_O_train, roc_Tree_S_O_test)
#######################################
Tree_S_O <- rpart(Anxiety ~ SMILE_DN + SMILE_SU + SMILE_PA + SMILE_SM +
                  SMILE_RS + SMILE_SS + SMILE_EE + Gender + Age +
                  Course + School + Displaced,
                data = df_train_O, method = "class",
                control=list(cp=0.0001,
                             maxdepth=4),
                             parms=list(split="information"))
```

```
Tree_S_O_prob_train <- predict(Tree_S_O, newdata = df_train_O,
    type = "prob")[,1]
Tree_S_O_prob_test <- predict(Tree_S_O, newdata = df_test,
    type = "prob")[,1]
roc_Tree_S_O_train = roc(df_train_O$Anxiety ~ Tree_S_O_prob_train)
r6A = roc_Tree_S_O_test = roc(df_test$Anxiety ~ Tree_S_O_prob_test)
roc.test(roc_Tree_S_O_train, roc_Tree_S_O_test)

fancyRpartPlot(Tree_S_O, caption = NULL)

plot(roc_Tree_S_O_train, print.auc = TRUE, col = 2, lwd =3,
     print.auc.y=0.98, print.auc.x=0.98,
     auc.polygon = TRUE, max.auc.polygon = TRUE);
plot(roc_Tree_S_O_test, add=TRUE , print.auc = TRUE, col = 3,
    lwd =3, print.auc.y=0.90, print.auc.x=0.98)
legend(0.40,0.28, c("Train ","Test "),
       col = c(2,3), lwd =2)

Tree_S_O_Class_test <- predict(Tree_S_O, newdata = df_test,
    type = "c")

CM=confusionMatrix(as.factor(Tree_S_O_Class_test),
                   df_test$Anxiety,
                   positive = "Anxiety")

paste(round(CM$overall[1],3), "&", round(CM$byClass[1],3), "&",
      round(CM$byClass[2],3), "&", round(CM$byClass[3],3), "&",
      round(CM$byClass[4],3), "&", round(CM$byClass[7],3), "&",
      round(roc_Tree_S_O_test$auc,3))

DI <- data.frame(imp = Tree_S_O$variable.importance)
DI$variable <- names(Tree_S_O$variable.importance)
ggplot(DI) +
  geom_col(aes(x = variable, y = imp),
           col = "black", show.legend = F) +
  labs(x = NULL,
       y = "Feature Importance",
       title =NULL) +
  coord_flip() +
  scale_fill_grey() +
  theme_bw()

# ROC summary
Cores <- c("red", "orange",  "purple", "firebrick",
    "blue", "darkturquoise")
plot(r1A, print.auc = TRUE, col = Cores[1], lwd =3,
     print.auc.y=0.98, print.auc.x=0.98,
     auc.polygon = TRUE, max.auc.polygon = TRUE);
plot(r2A, add=TRUE , print.auc = TRUE, col = Cores[2], lwd =3,
```

```
     print.auc.y=0.92, print.auc.x=0.98)
plot(r3A, add=TRUE , print.auc = TRUE, col = Cores[3], lwd =3,
     print.auc.y=0.86, print.auc.x=0.98)
plot(r4A, add=TRUE , print.auc = TRUE, col = Cores[4], lwd =3,
     print.auc.y=0.80, print.auc.x=0.98)
plot(r5A, add=TRUE , print.auc = TRUE, col = Cores[5], lwd =3,
     print.auc.y=0.74, print.auc.x=0.98)
plot(r6A, add=TRUE , print.auc = TRUE, col = Cores[6], lwd =3,
     print.auc.y=0.68, print.auc.x=0.98)
legend(0.49,0.48,
       c("A_DT_S ","A_DT_SC ","A_DT_SD",
         "A_DT_S_O ","A_DT_SC_O ","A_DT_SD_O"),
       col = Cores, lwd =3)


#########################################################
#########################################################
##### Multivariate analysis to screen for Depression

#########################################################
# Decision trees - whithout oversampling


# SMILE
Tree_S <- rpart(Depression ~ SMILE + Gender + Age + Course +
    School + Displaced,
                data = df_train, method = "class",
                control=list(cp=0.0001),
                parms=list(split="information"))

rpart.plot(Tree_S)
fancyRpartPlot(Tree_S, caption = NULL)

Tree_S_prob_train <- predict(Tree_S, newdata = df_train,
    type = "prob")[,1]
Tree_S_prob_test <- predict(Tree_S, newdata = df_test,
    type = "prob")[,1]
roc_Tree_S_train = roc(df_train$Depression ~ Tree_S_prob_train)
roc.test(roc_Tree_S_train, roc_Tree_S_test)
#####################################
Tree_S <- rpart(Depression ~ SMILE + Gender + Age + Course +
    School + Displaced,
                data = df_train, method = "class",
                control=list(cp=0.0001,
                             maxdepth=3),
                             parms=list(split="information"))

Tree_S_prob_train <- predict(Tree_S, newdata = df_train,
    type = "prob")[,1]
```

```
Tree_S_prob_test <- predict(Tree_S, newdata = df_test,
    type = "prob")[,1]
roc_Tree_S_train = roc(df_train$Depression ~ Tree_S_prob_train)
r1A = roc_Tree_S_test = roc(df_test$Depression ~ Tree_S_prob_test)
roc.test(roc_Tree_S_train, roc_Tree_S_test)

fancyRpartPlot(Tree_S, caption = NULL)

plot(roc_Tree_S_train, print.auc = TRUE, col = 2, lwd =3,
     print.auc.y=0.98, print.auc.x=0.98,
     auc.polygon = TRUE, max.auc.polygon = TRUE);
plot(roc_Tree_S_test, add=TRUE , print.auc = TRUE, col = 3,
    lwd =3, print.auc.y=0.90, print.auc.x=0.98)
legend(0.33,0.18, c("Train ","Test "),
       col = c(2,3), lwd =2)

Tree_S_Class_test <- predict(Tree_S, newdata = df_test,
    type = "c")

CM=confusionMatrix(as.factor(Tree_S_Class_test),
                   df_test$Depression,
                   positive = "Depression")

paste(round(CM$overall[1],3), "&", round(CM$byClass[1],3), "&",
      round(CM$byClass[2],3), "&", round(CM$byClass[3],3), "&",
      round(CM$byClass[4],3), "&", round(CM$byClass[7],3), "&",
      round(roc_Tree_S_test$auc,3))

DI <- data.frame(imp = Tree_S$variable.importance)
DI$variable <- names(Tree_S$variable.importance)
ggplot(DI) +
  geom_col(aes(x = variable, y = imp),
           col = "black", show.legend = F) +
  labs(x = NULL,
       y = "Feature Importance",
       title =NULL) +
  coord_flip() +
  scale_fill_grey() +
  theme_bw()

#########################################################
# SMILE-C
Tree_S <- rpart(Depression ~ SMILEC + Gender + Age + Course +
    School + Displaced,
                data = df_train, method = "class",
                control=list(cp=0.0001),
                parms=list(split="information"))

fancyRpartPlot(Tree_S, caption = NULL)
```

```
Tree_S_prob_train <- predict(Tree_S, newdata = df_train,
    type = "prob")[,1]
Tree_S_prob_test <- predict(Tree_S, newdata = df_test,
    type = "prob")[,1]
roc_Tree_S_train = roc(df_train$Depression ~ Tree_S_prob_train)
roc_Tree_S_test = roc(df_test$Depression ~ Tree_S_prob_test)
roc.test(roc_Tree_S_train, roc_Tree_S_test)
#####################################
Tree_S <- rpart(Depression ~ SMILEC + Gender + Age + Course +
    School + Displaced,
                data = df_train, method = "class",
                control=list(cp=0.0001,
                             maxdepth=3),
                             parms=list(split="information"))

Tree_S_prob_train <- predict(Tree_S, newdata = df_train,
type = "prob")[,2]
Tree_S_prob_test <- predict(Tree_S, newdata = df_test,
type = "prob")[,2]
roc_Tree_S_train = roc(df_train$Depression ~ Tree_S_prob_train)
r2A = roc_Tree_S_test = roc(df_test$Depression ~ Tree_S_prob_test)
roc.test(roc_Tree_S_train, roc_Tree_S_test)

fancyRpartPlot(Tree_S, caption = NULL)

plot(roc_Tree_S_train, print.auc = TRUE, col = 2, lwd =3,
     print.auc.y=0.98, print.auc.x=0.98,
     auc.polygon = TRUE, max.auc.polygon = TRUE);
plot(roc_Tree_S_test, add=TRUE , print.auc = TRUE, col = 3,
    lwd =3, print.auc.y=0.90, print.auc.x=0.98)
legend(0.33,0.18, c("Train ","Test "),
       col = c(2,3), lwd =2)

Tree_S_Class_test <- predict(Tree_S, newdata = df_test,
    type = "c")

CM=confusionMatrix(as.factor(Tree_S_Class_test),
                   df_test$Depression,
                   positive = "Depression")

paste(round(CM$overall[1],3), "&", round(CM$byClass[1],3), "&",
      round(CM$byClass[2],3), "&", round(CM$byClass[3],3), "&",
      round(CM$byClass[4],3), "&", round(CM$byClass[7],3), "&",
      round(roc_Tree_S_test$auc,3))

DI <- data.frame(imp = Tree_S$variable.importance)
DI$variable <- names(Tree_S$variable.importance)
ggplot(DI) +
```

```
  geom_col(aes(x = variable, y = imp),
           col = "black", show.legend = F) +
  labs(x = NULL,
       y = "Feature Importance",
       title =NULL) +
  coord_flip() +
  scale_fill_grey() +
  theme_bw()

#########################################################
# Domains
Tree_S <- rpart(Depression ~ SMILE_DN + SMILE_SU + SMILE_PA +
    SMILE_SM + SMILE_RS + SMILE_SS + SMILE_EE + Gender +
    Age + Course + School + Displaced,
                data = df_train, method = "class",
                control=list(cp=0.0001),
                parms=list(split="information"))

rpart.plot(Tree_S)
fancyRpartPlot(Tree_S, caption = NULL)

Tree_S_prob_train <- predict(Tree_S, newdata = df_train,
    type = "prob")[,1]
Tree_S_prob_test <- predict(Tree_S, newdata = df_test,
    type = "prob")[,1]
roc_Tree_S_train = roc(df_train$Depression ~ Tree_S_prob_train)
roc_Tree_S_test = roc(df_test$Depression ~ Tree_S_prob_test)
roc.test(roc_Tree_S_train, roc_Tree_S_test)
#####################################
Tree_S <- rpart(Depression ~ SMILE_DN + SMILE_SU + SMILE_PA +
    SMILE_SM + SMILE_RS + SMILE_SS + SMILE_EE + Gender +
    Age + Course + School + Displaced,
                data = df_train, method = "class",
                control=list(cp=0.0001,
                             maxdepth=4),
                             parms=list(split="information"))

Tree_S_prob_train <- predict(Tree_S, newdata = df_train,
    type = "prob")[,1]
Tree_S_prob_test <- predict(Tree_S, newdata = df_test,
    type = "prob")[,1]
roc_Tree_S_train = roc(df_train$Depression ~ Tree_S_prob_train)
r3A = roc_Tree_S_test = roc(df_test$Depression ~ Tree_S_prob_test)
roc.test(roc_Tree_S_train, roc_Tree_S_test)

fancyRpartPlot(Tree_S, caption = NULL)

plot(roc_Tree_S_train, print.auc = TRUE, col = 2, lwd =3,
     print.auc.y=0.98, print.auc.x=0.98,
```

```
     auc.polygon = TRUE, max.auc.polygon = TRUE);
plot(roc_Tree_S_test, add=TRUE , print.auc = TRUE, col = 3,
    lwd =3, print.auc.y=0.90, print.auc.x=0.98)
legend(0.33,0.18, c("Train ","Test "),
       col = c(2,3), lwd =2)

Tree_S_Class_test <- predict(Tree_S, newdata = df_test,
    type = "c")

CM=confusionMatrix(as.factor(Tree_S_Class_test),
                   df_test$Depression,
                   positive = "Depression")

paste(round(CM$overall[1],3), "&", round(CM$byClass[1],3), "&",
      round(CM$byClass[2],3), "&", round(CM$byClass[3],3), "&",
      round(CM$byClass[4],3), "&", round(CM$byClass[7],3), "&",
      round(roc_Tree_S_test$auc,3))

DI <- data.frame(imp = Tree_S$variable.importance)
DI$variable <- names(Tree_S$variable.importance)
ggplot(DI) +
  geom_col(aes(x = variable, y = imp),
           col = "black", show.legend = F) +
  labs(x = NULL,
       y = "Feature Importance",
       title =NULL) +
  coord_flip() +
  scale_fill_grey() +
  theme_bw()

############################################################
# Decision trees - whith oversampling

############################################################
### Oversampling
table(df_train$Depression)
set.seed(123)
df_train_O <- RandOverClassif(Depression~., df_train, "balance")
table(df_train_O$Depression)

############################################################
# SMILE
Tree_S_O <- rpart(Depression ~ SMILE + Gender + Age + Course +
    School + Displaced,
                  data = df_train_O, method = "class",
                  control=list(cp=0.0001),
                  parms=list(split="information"))

rpart.plot(Tree_S_O)
```

```
fancyRpartPlot(Tree_S_O, caption = NULL)

Tree_S_O_prob_train <- predict(Tree_S_O, newdata = df_train_O,
    type = "prob")[,1]
Tree_S_O_prob_test <- predict(Tree_S_O, newdata = df_test,
    type = "prob")[,1]
roc_Tree_S_O_train = roc(df_train_O$Depression ~ Tree_S_O_prob_train)
roc_Tree_S_O_test = roc(df_test$Depression ~ Tree_S_O_prob_test)
roc.test(roc_Tree_S_O_train, roc_Tree_S_O_test)
#####################################
Tree_S_O <- rpart(Depression ~ SMILE + Gender + Age + Course +
    School + Displaced,
                  data = df_train_O, method = "class",
                  control=list(cp=0.0001,
                               maxdepth=2),
                               parms=list(split="information"))

Tree_S_O_prob_train <- predict(Tree_S_O, newdata = df_train_O,
    type = "prob")[,1]
Tree_S_O_prob_test <- predict(Tree_S_O, newdata = df_test,
    type = "prob")[,1]
roc_Tree_S_O_train = roc(df_train_O$Depression ~ Tree_S_O_prob_train)
r4A = roc_Tree_S_O_test = roc(df_test$Depression ~ Tree_S_O_prob_test)
roc.test(roc_Tree_S_O_train, roc_Tree_S_O_test)

fancyRpartPlot(Tree_S_O, caption = NULL)

plot(roc_Tree_S_O_train, print.auc = TRUE, col = 2, lwd =3,
     print.auc.y=0.98, print.auc.x=0.98,
     auc.polygon = TRUE, max.auc.polygon = TRUE);
plot(roc_Tree_S_O_test, add=TRUE , print.auc = TRUE, col = 3,
    lwd =3, print.auc.y=0.90, print.auc.x=0.98)
legend(0.33,0.18, c("Train ","Test "),
       col = c(2,3), lwd =2)

Tree_S_O_Class_test <- predict(Tree_S_O, newdata = df_test,
    type = "c")

CM=confusionMatrix(as.factor(Tree_S_O_Class_test),
                   df_test$Depression,
                   positive = "Depression")

paste(round(CM$overall[1],3), "&", round(CM$byClass[1],3), "&",
      round(CM$byClass[2],3), "&", round(CM$byClass[3],3), "&",
      round(CM$byClass[4],3), "&", round(CM$byClass[7],3), "&",
      round(roc_Tree_S_O_test$auc,3))

DI <- data.frame(imp = Tree_S_O$variable.importance)
DI$variable <- names(Tree_S_O$variable.importance)
```

```
ggplot(DI) +
  geom_col(aes(x = variable, y = imp),
           col = "black", show.legend = F) +
  labs(x = NULL,
       y = "Feature Importance",
       title =NULL) +
  coord_flip() +
  scale_fill_grey() +
  theme_bw()

#########################################################
# SMILE-C
Tree_S_O <- rpart(Depression ~ SMILEC + Gender + Age + Course +
    School + Displaced,
                  data = df_train_O, method = "class",
                  control=list(cp=0.0001),
                  parms=list(split="information"))

fancyRpartPlot(Tree_S_O, caption = NULL)

Tree_S_O_prob_train <- predict(Tree_S_O, newdata = df_train_O,
    type = "prob")[,1]
Tree_S_O_prob_test <- predict(Tree_S_O, newdata = df_test,
    type = "prob")[,1]
roc_Tree_S_O_train = roc(df_train_O$Depression ~ Tree_S_O_prob_train)
roc_Tree_S_O_test = roc(df_test$Depression ~ Tree_S_O_prob_test)
roc.test(roc_Tree_S_O_train, roc_Tree_S_O_test)
#####################################
Tree_S_O <- rpart(Depression ~ SMILEC + Gender + Age + Course +
    School + Displaced,
                  data = df_train_O, method = "class",
                  control=list(cp=0.0001,
                               maxdepth=3),
                               parms=list(split="information"))

Tree_S_O_prob_train <- predict(Tree_S_O, newdata = df_train_O,
    type = "prob")[,2]
Tree_S_O_prob_test <- predict(Tree_S_O, newdata = df_test,
    type = "prob")[,2]
roc_Tree_S_O_train = roc(df_train_O$Depression ~ Tree_S_O_prob_train)
r5A = roc_Tree_S_O_test = roc(df_test$Depression ~ Tree_S_O_prob_test)
roc.test(roc_Tree_S_O_train, roc_Tree_S_O_test)

fancyRpartPlot(Tree_S_O, caption = NULL)

plot(roc_Tree_S_O_train, print.auc = TRUE, col = 2, lwd =3,
     print.auc.y=0.98, print.auc.x=0.98,
     auc.polygon = TRUE, max.auc.polygon = TRUE);
plot(roc_Tree_S_O_test, add=TRUE , print.auc = TRUE, col = 3,
```

```
    lwd =3, print.auc.y=0.90, print.auc.x=0.98)
legend(0.33,0.18, c("Train ","Test "),
       col = c(2,3), lwd =2)

Tree_S_O_Class_test <- predict(Tree_S_O, newdata = df_test,
    type = "c")

CM=confusionMatrix(as.factor(Tree_S_O_Class_test),
                   df_test$Depression,
                   positive = "Depression")

paste(round(CM$overall[1],3), "&", round(CM$byClass[1],3), "&",
      round(CM$byClass[2],3), "&", round(CM$byClass[3],3), "&",
      round(CM$byClass[4],3), "&", round(CM$byClass[7],3), "&",
      round(roc_Tree_S_O_test$auc,3))

DI <- data.frame(imp = Tree_S_O$variable.importance)
DI$variable <- names(Tree_S_O$variable.importance)
ggplot(DI) +
  geom_col(aes(x = variable, y = imp),
           col = "black", show.legend = F) +
  labs(x = NULL,
       y = "Feature Importance",
       title =NULL) +
  coord_flip() +
  scale_fill_grey() +
  theme_bw()

#########################################################
# Domains
Tree_S_O <- rpart(Depression ~ SMILE_DN + SMILE_SU + SMILE_PA +
    SMILE_SM + SMILE_RS + SMILE_SS + SMILE_EE + Gender +
    Age + Course + School + Displaced,
                  data = df_train_O, method = "class",
                  control=list(cp=0.0001),
                  parms=list(split="information"))

rpart.plot(Tree_S_O)
fancyRpartPlot(Tree_S_O, caption = NULL)

Tree_S_O_prob_train <- predict(Tree_S_O, newdata = df_train_O,
    type = "prob")[,1]
Tree_S_O_prob_test <- predict(Tree_S_O, newdata = df_test,
    type = "prob")[,1]
roc_Tree_S_O_train = roc(df_train_O$Depression ~ Tree_S_O_prob_train)
roc_Tree_S_O_test = roc(df_test$Depression ~ Tree_S_O_prob_test)
roc.test(roc_Tree_S_O_train, roc_Tree_S_O_test)
#####################################
Tree_S_O <- rpart(Depression ~ SMILE_DN + SMILE_SU + SMILE_PA +
```

```
    SMILE_SM + SMILE_RS + SMILE_SS + SMILE_EE + Gender +
    Age + Course + School + Displaced,
                  data = df_train_O, method = "class",
                  control=list(cp=0.0001,
                               maxdepth=2),
                               parms=list(split="information"))

Tree_S_O_prob_train <- predict(Tree_S_O, newdata = df_train_O,
    type = "prob")[,1]
Tree_S_O_prob_test <- predict(Tree_S_O, newdata = df_test,
    type = "prob")[,1]
roc_Tree_S_O_train = roc(df_train_O$Depression ~ Tree_S_O_prob_train)
r6A = roc_Tree_S_O_test = roc(df_test$Depression ~ Tree_S_O_prob_test)
roc.test(roc_Tree_S_O_train, roc_Tree_S_O_test)

fancyRpartPlot(Tree_S_O, caption = NULL)

plot(roc_Tree_S_O_train, print.auc = TRUE, col = 2, lwd =3,
     print.auc.y=0.98, print.auc.x=0.98,
     auc.polygon = TRUE, max.auc.polygon = TRUE);
plot(roc_Tree_S_O_test, add=TRUE , print.auc = TRUE, col = 3,
    lwd =3, print.auc.y=0.90, print.auc.x=0.98)
legend(0.33,0.18, c("Train ","Test "),
       col = c(2,3), lwd =2)

Tree_S_O_Class_test <- predict(Tree_S_O, newdata = df_test,
    type = "c")

CM=confusionMatrix(as.factor(Tree_S_O_Class_test),
                   df_test$Depression,
                   positive = "Depression")

paste(round(CM$overall[1],3), "&", round(CM$byClass[1],3), "&",
      round(CM$byClass[2],3), "&", round(CM$byClass[3],3), "&",
      round(CM$byClass[4],3), "&", round(CM$byClass[7],3), "&",
      round(roc_Tree_S_O_test$auc,3))

DI <- data.frame(imp = Tree_S_O$variable.importance)
DI$variable <- names(Tree_S_O$variable.importance)
ggplot(DI) +
  geom_col(aes(x = variable, y = imp),
           col = "black", show.legend = F) +
  labs(x = NULL,
       y = "Feature Importance",
       title =NULL) +
  coord_flip() +
  scale_fill_grey() +
  theme_bw()
```

```
# ROC summary
Cores <- c("red", "orange",  "purple", "firebrick",
    "blue", "darkturquoise")
plot(r1A, print.auc = TRUE, col = Cores[1], lwd =3,
     print.auc.y=0.98, print.auc.x=0.98,
     auc.polygon = TRUE, max.auc.polygon = TRUE);
plot(r2A, add=TRUE , print.auc = TRUE, col = Cores[2], lwd =3,
     print.auc.y=0.92, print.auc.x=0.98)
plot(r3A, add=TRUE , print.auc = TRUE, col = Cores[3], lwd =3,
     print.auc.y=0.86, print.auc.x=0.98)
plot(r4A, add=TRUE , print.auc = TRUE, col = Cores[4], lwd =3,
     print.auc.y=0.80, print.auc.x=0.98)
plot(r5A, add=TRUE , print.auc = TRUE, col = Cores[5], lwd =3,
     print.auc.y=0.74, print.auc.x=0.98)
plot(r6A, add=TRUE , print.auc = TRUE, col = Cores[6], lwd =3,
     print.auc.y=0.68, print.auc.x=0.98)
legend(0.42,0.3,
       c("D_DT_S ","D_DT_SC ","D_DT_SD",
         "D_DT_S_O ","D_DT_SC_O ","D_DT_SD_O"),
       col = Cores, lwd =3)

#########################################################
##### Multivariate analysis to screen for Anxiety and Depression

#########################################################
# Decision trees - whithout oversampling

#########################################################
# SMILE
Tree_S <- rpart(Anxiety_Depression ~ SMILE + Gender + Age +
    Course + School + Displaced,
                data = df_train, method = "class",
                control=list(cp=0.0001),
                parms=list(split="information"))

rpart.plot(Tree_S)
fancyRpartPlot(Tree_S, caption = NULL)

Tree_S_prob_train <- predict(Tree_S, newdata = df_train,
    type = "prob")[,1]
Tree_S_prob_test <- predict(Tree_S, newdata = df_test,
    type = "prob")[,1]
roc_Tree_S_train = roc(df_train$Anxiety_Depression ~ Tree_S_prob_train)
roc.test(roc_Tree_S_train, roc_Tree_S_test)
#####################################
Tree_S <- rpart(Anxiety_Depression ~ SMILE + Gender + Age +
    Course + School + Displaced,
                data = df_train, method = "class",
                control=list(cp=0.0001,
```

```
                                maxdepth=3),
                                parms=list(split="information"))

Tree_S_prob_train <- predict(Tree_S, newdata = df_train,
    type = "prob")[,1]
Tree_S_prob_test <- predict(Tree_S, newdata = df_test,
    type = "prob")[,1]
roc_Tree_S_train = roc(df_train$Anxiety_Depression ~ Tree_S_prob_train)
r1A = roc_Tree_S_test = roc(df_test$Anxiety_Depression ~ Tree_S_prob_test)
roc.test(roc_Tree_S_train, roc_Tree_S_test)

fancyRpartPlot(Tree_S, caption = NULL)

plot(roc_Tree_S_train, print.auc = TRUE, col = 2, lwd =3,
     print.auc.y=0.98, print.auc.x=0.98,
     auc.polygon = TRUE, max.auc.polygon = TRUE);
plot(roc_Tree_S_test, add=TRUE , print.auc = TRUE, col = 3, lwd =3,
     print.auc.y=0.90, print.auc.x=0.98)
legend(0.37,0.25, c("Train ","Test "),
       col = c(2,3), lwd =2)

Tree_S_Class_test <- predict(Tree_S, newdata = df_test, type = "c")

CM=confusionMatrix(as.factor(Tree_S_Class_test),
                   df_test$Anxiety_Depression,
                   positive = "Anxiety and Depression")

paste(round(CM$overall[1],3), "&", round(CM$byClass[1],3), "&",
      round(CM$byClass[2],3), "&", round(CM$byClass[3],3), "&",
      round(CM$byClass[4],3), "&", round(CM$byClass[7],3), "&",
      round(roc_Tree_S_test$auc,3))

DI <- data.frame(imp = Tree_S$variable.importance)
DI$variable <- names(Tree_S$variable.importance)
ggplot(DI) +
  geom_col(aes(x = variable, y = imp),
           col = "black", show.legend = F) +
  labs(x = NULL,
       y = "Feature Importance",
       title =NULL) +
  coord_flip() +
  scale_fill_grey() +
  theme_bw()

#########################################################
# SMILE-C
Tree_S <- rpart(Anxiety_Depression ~ SMILEC + Gender + Age +
    Course + School + Displaced,
                data = df_train, method = "class",
```

```
                control=list(cp=0.0001),
                parms=list(split="information"))

fancyRpartPlot(Tree_S, caption = NULL)

Tree_S_prob_train <- predict(Tree_S, newdata = df_train,
    type = "prob")[,1]
Tree_S_prob_test <- predict(Tree_S, newdata = df_test,
    type = "prob")[,1]
roc_Tree_S_train = roc(df_train$Anxiety_Depression ~ Tree_S_prob_train)
roc_Tree_S_test = roc(df_test$Anxiety_Depression ~ Tree_S_prob_test)
roc.test(roc_Tree_S_train, roc_Tree_S_test)
#####################################
Tree_S <- rpart(Anxiety_Depression ~ SMILEC + Gender + Age +
    Course + School + Displaced,
                data = df_train, method = "class",
                control=list(cp=0.0001,
                             maxdepth=6),
                             parms=list(split="information"))

Tree_S_prob_train <- predict(Tree_S, newdata = df_train,
    type = "prob")[,2]
Tree_S_prob_test <- predict(Tree_S, newdata = df_test,
    type = "prob")[,2]
roc_Tree_S_train = roc(df_train$Anxiety_Depression ~ Tree_S_prob_train)
r2A = roc_Tree_S_test = roc(df_test$Anxiety_Depression ~ Tree_S_prob_test)
roc.test(roc_Tree_S_train, roc_Tree_S_test)

fancyRpartPlot(Tree_S, caption = NULL)

plot(roc_Tree_S_train, print.auc = TRUE, col = 2, lwd =3,
     print.auc.y=0.98, print.auc.x=0.98,
     auc.polygon = TRUE, max.auc.polygon = TRUE);
plot(roc_Tree_S_test, add=TRUE , print.auc = TRUE, col = 3, lwd =3,
     print.auc.y=0.90, print.auc.x=0.98)
legend(0.33,0.18, c("Train ","Test "),
       col = c(2,3), lwd =2)

Tree_S_Class_test <- predict(Tree_S, newdata = df_test, type = "c")

CM=confusionMatrix(as.factor(Tree_S_Class_test),
                   df_test$Anxiety_Depression,
                   positive = "Anxiety and Depression")

paste(round(CM$overall[1],3), "&", round(CM$byClass[1],3), "&",
      round(CM$byClass[2],3), "&", round(CM$byClass[3],3), "&",
      round(CM$byClass[4],3), "&", round(CM$byClass[7],3), "&",
      round(roc_Tree_S_test$auc,3))
```

```
DI <- data.frame(imp = Tree_S$variable.importance)
DI$variable <- names(Tree_S$variable.importance)
ggplot(DI) +
  geom_col(aes(x = variable, y = imp),
           col = "black", show.legend = F) +
  labs(x = NULL,
       y = "Feature Importance",
       title =NULL) +
  coord_flip() +
  scale_fill_grey() +
  theme_bw()

#########################################################
# Domains
Tree_S <- rpart(Anxiety_Depression ~ SMILE_DN + SMILE_SU + SMILE_PA +
    SMILE_SM + SMILE_RS + SMILE_SS + SMILE_EE + Gender +
    Age + Course + School + Displaced,
                data = df_train, method = "class",
                control=list(cp=0.0001),
                parms=list(split="information"))

rpart.plot(Tree_S)
fancyRpartPlot(Tree_S, caption = NULL)

Tree_S_prob_train <- predict(Tree_S, newdata = df_train,
    type = "prob")[,1]
Tree_S_prob_test <- predict(Tree_S, newdata = df_test,
    type = "prob")[,1]
roc_Tree_S_train = roc(df_train$Anxiety_Depression ~ Tree_S_prob_train)
roc_Tree_S_test = roc(df_test$Anxiety_Depression ~ Tree_S_prob_test)
roc.test(roc_Tree_S_train, roc_Tree_S_test)
#####################################
Tree_S <- rpart(Anxiety_Depression ~ SMILE_DN + SMILE_SU +
    SMILE_PA + SMILE_SM +
                   SMILE_RS + SMILE_SS + SMILE_EE + Gender + Age +
                   Course + School + Displaced,
                data = df_train, method = "class",
                control=list(cp=0.0001,
                             maxdepth=4),
                             parms=list(split="information"))

Tree_S_prob_train <- predict(Tree_S, newdata = df_train,
    type = "prob")[,1]
Tree_S_prob_test <- predict(Tree_S, newdata = df_test,
    type = "prob")[,1]
roc_Tree_S_train = roc(df_train$Anxiety_Depression ~ Tree_S_prob_train)
r3A = roc_Tree_S_test = roc(df_test$Anxiety_Depression ~ Tree_S_prob_test)
roc.test(roc_Tree_S_train, roc_Tree_S_test)
```

```
fancyRpartPlot(Tree_S, caption = NULL)

plot(roc_Tree_S_train, print.auc = TRUE, col = 2, lwd =3,
     print.auc.y=0.98, print.auc.x=0.98,
     auc.polygon = TRUE, max.auc.polygon = TRUE);
plot(roc_Tree_S_test, add=TRUE , print.auc = TRUE, col = 3, lwd =3,
     print.auc.y=0.90, print.auc.x=0.98)
legend(0.33,0.18, c("Train ","Test "),
       col = c(2,3), lwd =2)

Tree_S_Class_test <- predict(Tree_S, newdata = df_test, type = "c")

CM=confusionMatrix(as.factor(Tree_S_Class_test),
                   df_test$Anxiety_Depression,
                   positive = "Anxiety and Depression")

paste(round(CM$overall[1],3), "&", round(CM$byClass[1],3), "&",
      round(CM$byClass[2],3), "&", round(CM$byClass[3],3), "&",
      round(CM$byClass[4],3), "&", round(CM$byClass[7],3), "&",
      round(roc_Tree_S_test$auc,3))

DI <- data.frame(imp = Tree_S$variable.importance)
DI$variable <- names(Tree_S$variable.importance)
ggplot(DI) +
  geom_col(aes(x = variable, y = imp),
           col = "black", show.legend = F) +
  labs(x = NULL,
       y = "Feature Importance",
       title =NULL) +
  coord_flip() +
  scale_fill_grey() +
  theme_bw()

############################################################
# Decision trees - whith oversampling

############################################################
### Oversampling
table(df_train$Anxiety_Depression)
set.seed(123)
df_train_O <- RandOverClassif(Anxiety_Depression~., df_train, "balance")
table(df_train_O$Anxiety_Depression)

############################################################
# SMILE
Tree_S_O <- rpart(Anxiety_Depression ~ SMILE + Gender + Age +
    Course + School + Displaced,
                  data = df_train_O, method = "class",
                  control=list(cp=0.0001),
```

```
                    parms=list(split="information"))

rpart.plot(Tree_S_O)
fancyRpartPlot(Tree_S_O, caption = NULL)

Tree_S_O_prob_train <- predict(Tree_S_O, newdata = df_train_O,
    type = "prob")[,1]
Tree_S_O_prob_test <- predict(Tree_S_O, newdata = df_test,
    type = "prob")[,1]
roc_Tree_S_O_train = roc(df_train_O$Anxiety_Depression ~
            Tree_S_O_prob_train)
roc_Tree_S_O_test = roc(df_test$Anxiety_Depression ~
            Tree_S_O_prob_test)
roc.test(roc_Tree_S_O_train, roc_Tree_S_O_test)
#####################################
Tree_S_O <- rpart(Anxiety_Depression ~ SMILE + Gender + Age +
    Course + School + Displaced,
                    data = df_train_O, method = "class",
                    control=list(cp=0.0001,
                                 maxdepth=3),
                                 parms=list(split="information"))

Tree_S_O_prob_train <- predict(Tree_S_O, newdata = df_train_O,
    type = "prob")[,1]
Tree_S_O_prob_test <- predict(Tree_S_O, newdata = df_test,
    type = "prob")[,1]
roc_Tree_S_O_train = roc(df_train_O$Anxiety_Depression ~
            Tree_S_O_prob_train)
r4A = roc_Tree_S_O_test = roc(df_test$Anxiety_Depression ~
            Tree_S_O_prob_test)
roc.test(roc_Tree_S_O_train, roc_Tree_S_O_test)

fancyRpartPlot(Tree_S_O, caption = NULL)

plot(roc_Tree_S_O_train, print.auc = TRUE, col = 2, lwd =3,
     print.auc.y=0.98, print.auc.x=0.98,
     auc.polygon = TRUE, max.auc.polygon = TRUE);
plot(roc_Tree_S_O_test, add=TRUE , print.auc = TRUE, col = 3, lwd =3,
     print.auc.y=0.90, print.auc.x=0.98)
legend(0.33,0.18, c("Train ","Test "),
       col = c(2,3), lwd =2)

Tree_S_O_Class_test <- predict(Tree_S_O, newdata = df_test, type = "c")

CM=confusionMatrix(as.factor(Tree_S_O_Class_test),
                   df_test$Anxiety_Depression,
                   positive = "Anxiety and Depression")

paste(round(CM$overall[1],3), "&", round(CM$byClass[1],3), "&",
```

```
      round(CM$byClass[2],3), "&", round(CM$byClass[3],3), "&",
      round(CM$byClass[4],3), "&", round(CM$byClass[7],3), "&",
      round(roc_Tree_S_O_test$auc,3))

DI <- data.frame(imp = Tree_S_O$variable.importance)
DI$variable <- names(Tree_S_O$variable.importance)
ggplot(DI) +
  geom_col(aes(x = variable, y = imp),
           col = "black", show.legend = F) +
  labs(x = NULL,
       y = "Feature Importance",
       title =NULL) +
  coord_flip() +
  scale_fill_grey() +
  theme_bw()

#########################################################
# SMILE-C
Tree_S_O <- rpart(Anxiety_Depression ~ SMILEC + Gender + Age +
    Course + School + Displaced,
                  data = df_train_O, method = "class",
                  control=list(cp=0.0001),
                  parms=list(split="information"))

fancyRpartPlot(Tree_S_O, caption = NULL)

Tree_S_O_prob_train <- predict(Tree_S_O, newdata = df_train_O,
    type = "prob")[,1]
Tree_S_O_prob_test <- predict(Tree_S_O, newdata = df_test,
    type = "prob")[,1]
roc_Tree_S_O_train = roc(df_train_O$Anxiety_Depression ~
            Tree_S_O_prob_train)
roc_Tree_S_O_test = roc(df_test$Anxiety_Depression ~
            Tree_S_O_prob_test)
roc.test(roc_Tree_S_O_train, roc_Tree_S_O_test)
#####################################
Tree_S_O <- rpart(Anxiety_Depression ~ SMILEC + Gender + Age +
    Course + School + Displaced,
                  data = df_train_O, method = "class",
                  control=list(cp=0.0001,
                               maxdepth=3),
                               parms=list(split="information"))

Tree_S_O_prob_train <- predict(Tree_S_O, newdata = df_train_O,
    type = "prob")[,2]
Tree_S_O_prob_test <- predict(Tree_S_O, newdata = df_test,
    type = "prob")[,2]
roc_Tree_S_O_train = roc(df_train_O$Anxiety_Depression ~
            Tree_S_O_prob_train)
```

```
r5A = roc_Tree_S_O_test = roc(df_test$Anxiety_Depression ~
            Tree_S_O_prob_test)
roc.test(roc_Tree_S_O_train, roc_Tree_S_O_test)

fancyRpartPlot(Tree_S_O, caption = NULL)

plot(roc_Tree_S_O_train, print.auc = TRUE, col = 2, lwd =3,
     print.auc.y=0.98, print.auc.x=0.98,
     auc.polygon = TRUE, max.auc.polygon = TRUE);
plot(roc_Tree_S_O_test, add=TRUE , print.auc = TRUE, col = 3, lwd =3,
     print.auc.y=0.90, print.auc.x=0.98)
legend(0.33,0.18, c("Train ","Test "),
       col = c(2,3), lwd =2)

Tree_S_O_Class_test <- predict(Tree_S_O, newdata = df_test, type = "c")

CM=confusionMatrix(as.factor(Tree_S_O_Class_test),
                   df_test$Anxiety_Depression,
                   positive = "Anxiety and Depression")
paste(round(CM$overall[1],3), "&", round(CM$byClass[1],3), "&",
      round(CM$byClass[2],3), "&", round(CM$byClass[3],3), "&",
      round(CM$byClass[4],3), "&", round(CM$byClass[7],3), "&",
      round(roc_Tree_S_O_test$auc,3))

DI <- data.frame(imp = Tree_S_O$variable.importance)
DI$variable <- names(Tree_S_O$variable.importance)
ggplot(DI) +
  geom_col(aes(x = variable, y = imp),
           col = "black", show.legend = F) +
  labs(x = NULL,
       y = "Feature Importance",
       title =NULL) +
  coord_flip() +
  scale_fill_grey() +
  theme_bw()

#########################################################
# Domains
Tree_S_O <- rpart(Anxiety_Depression ~ SMILE_DN + SMILE_SU +
    SMILE_PA + SMILE_SM +
                    SMILE_RS + SMILE_SS + SMILE_EE + Gender + Age +
                    Course + School + Displaced,
                  data = df_train_O, method = "class",
                  control=list(cp=0.0001),
                  parms=list(split="information"))

rpart.plot(Tree_S_O)
fancyRpartPlot(Tree_S_O, caption = NULL)
```

```
Tree_S_O_prob_train <- predict(Tree_S_O, newdata = df_train_O,
    type = "prob")[,1]
Tree_S_O_prob_test <- predict(Tree_S_O, newdata = df_test,
    type = "prob")[,1]
roc_Tree_S_O_train = roc(df_train_O$Anxiety_Depression ~
            Tree_S_O_prob_train)
roc_Tree_S_O_test = roc(df_test$Anxiety_Depression ~
            Tree_S_O_prob_test)
roc.test(roc_Tree_S_O_train, roc_Tree_S_O_test)
#####################################
Tree_S_O <- rpart(Anxiety_Depression ~ SMILE_DN + SMILE_SU +
    SMILE_PA + SMILE_SM +
                    SMILE_RS + SMILE_SS + SMILE_EE + Gender + Age +
                    Course + School + Displaced,
                data = df_train_O, method = "class",
                control=list(cp=0.0001,
                             maxdepth=3),
                             parms=list(split="information"))

Tree_S_O_prob_train <- predict(Tree_S_O, newdata = df_train_O,
    type = "prob")[,1]
Tree_S_O_prob_test <- predict(Tree_S_O, newdata = df_test,
    type = "prob")[,1]
roc_Tree_S_O_train = roc(df_train_O$Anxiety_Depression ~
            Tree_S_O_prob_train)
r6A = roc_Tree_S_O_test = roc(df_test$Anxiety_Depression ~
            Tree_S_O_prob_test)
roc.test(roc_Tree_S_O_train, roc_Tree_S_O_test)

fancyRpartPlot(Tree_S_O, caption = NULL)

plot(roc_Tree_S_O_train, print.auc = TRUE, col = 2, lwd =3,
     print.auc.y=0.98, print.auc.x=0.98,
     auc.polygon = TRUE, max.auc.polygon = TRUE);
plot(roc_Tree_S_O_test, add=TRUE , print.auc = TRUE, col = 3, lwd =3,
     print.auc.y=0.90, print.auc.x=0.98)
legend(0.33,0.18, c("Train ","Test "),
       col = c(2,3), lwd =2)

Tree_S_O_Class_test <- predict(Tree_S_O, newdata = df_test, type = "c")

CM=confusionMatrix(as.factor(Tree_S_O_Class_test),
                   df_test$Anxiety_Depression,
                   positive = "Anxiety and Depression")

paste(round(CM$overall[1],3), "&", round(CM$byClass[1],3), "&",
      round(CM$byClass[2],3), "&", round(CM$byClass[3],3), "&",
      round(CM$byClass[4],3), "&", round(CM$byClass[7],3), "&",
      round(roc_Tree_S_O_test$auc,3))
```

```
DI <- data.frame(imp = Tree_S_O$variable.importance)
DI$variable <- names(Tree_S_O$variable.importance)
ggplot(DI) +
  geom_col(aes(x = variable, y = imp),
           col = "black", show.legend = F) +
  labs(x = NULL,
       y = "Feature Importance",
       title =NULL) +
  coord_flip() +
  scale_fill_grey() +
  theme_bw()

# ROC summary
Cores <- c("red", "orange",  "purple", "firebrick",
    "blue", "darkturquoise")
plot(r1A, print.auc = TRUE, col = Cores[1], lwd =3,
     print.auc.y=0.98, print.auc.x=0.98,
     auc.polygon = TRUE, max.auc.polygon = TRUE);
plot(r2A, add=TRUE , print.auc = TRUE, col = Cores[2], lwd =3,
     print.auc.y=0.92, print.auc.x=0.98)
plot(r3A, add=TRUE , print.auc = TRUE, col = Cores[3], lwd =3,
     print.auc.y=0.86, print.auc.x=0.98)
plot(r4A, add=TRUE , print.auc = TRUE, col = Cores[4], lwd =3,
     print.auc.y=0.80, print.auc.x=0.98)
plot(r5A, add=TRUE , print.auc = TRUE, col = Cores[5], lwd =3,
     print.auc.y=0.74, print.auc.x=0.98)
plot(r6A, add=TRUE , print.auc = TRUE, col = Cores[6], lwd =3,
     print.auc.y=0.68, print.auc.x=0.98)
legend(0.45,0.3,
       c("AD_DT_S ","AD_DT_SC ","AD_DT_SD",
         "AD_DT_S_O ","AD_DT_SC_O ","AD_DT_SD_O"),
       col = Cores, lwd =3)


#############################################################
##### Multivariate analysis to screen for Anxiety

#############################################################
# Logistic regression - whithout oversampling

#############################################################
# SMILE
LR_A1_1 <- glm(Anxiety ~ SMILE + Gender + Age + Course +
                 School + Displaced,
               family=binomial(link="logit"), data = df_train)
summary(LR_A1_1)
LR_A1_2 = update(LR_A1_1,~. - Displaced)
anova(LR_A1_1,LR_A1_2, test = "Chisq")
```

```
summary(LR_A1_2)
LR_A1_3 = update(LR_A1_2,~. - Age)
anova(LR_A1_2,LR_A1_3, test = "Chisq")
summary(LR_A1_3)
LR_A1_4 = update(LR_A1_3,~. - Course)
anova(LR_A1_3,LR_A1_4, test = "Chisq")
anova(LR_A1_1,LR_A1_4, test = "Chisq")
LR_A1 <- LR_A1_4
summary(LR_A1)
coefficients(LR_A1)
exp(LR_A1$coefficients)

hoslem.test(LR_A1$y, LR_A1$fitted.values, g=10)

LR_prob_train <- LR_A1$fitted.values
LR_prob_test <- predict(LR_A1, df_test, type="response")
roc_LR_train = roc(df_train$Anxiety ~ LR_prob_train)
roc_LR_test = roc(df_test$Anxiety ~ LR_prob_test)
rla1 = roc(df_test$Anxiety ~ LR_prob_test)
plot(roc_LR_train, print.auc = TRUE, col = 2, lwd =3,
     print.auc.y=0.98, print.auc.x=0.98,
     auc.polygon = TRUE, max.auc.polygon = TRUE);
plot(roc_LR_test, add=TRUE , print.auc = TRUE, col = 3, lwd =3,
     print.auc.y=0.90, print.auc.x=0.98);
legend(0.33,0.18, c("Train ","Test "),
       col = c(2,3), lwd =2)
roc.test(roc_LR_train, roc_LR_test)

LR_Class_test <- factor(ifelse(LR_prob_test >=0.5,
    "Anxiety", "Negative"),
    levels= c("Negative","Anxiety"))
CM=confusionMatrix(as.factor(LR_Class_test),
                   df_test$Anxiety,
                   positive = "Anxiety")
paste(round(CM$overall[1],3), "&", round(CM$byClass[1],3), "&",
      round(CM$byClass[2],3), "&", round(CM$byClass[3],3), "&",
      round(CM$byClass[4],3), "&", round(CM$byClass[7],3), "&",
      round(roc_LR_test$auc,3))

#########################################################
# SMILE-C
LR_A2_1 <- glm(Anxiety ~ SMILEC + Gender + Age + Course +
                 School + Displaced,
               family=binomial(link="logit"), data = df_train)
summary(LR_A2_1)
LR_A2_2 = update(LR_A2_1,~. - Displaced)
anova(LR_A2_1,LR_A2_2, test = "Chisq")
summary(LR_A2_2)
LR_A2_3 = update(LR_A2_2,~. - School)
```

```
anova(LR_A2_2,LR_A2_3, test = "Chisq")
summary(LR_A2_3)
LR_A2_4 = update(LR_A2_3,~. - Course)
anova(LR_A2_3,LR_A2_4, test = "Chisq")
LR_A2 <- LR_A2_4
summary(LR_A2)
coefficients(LR_A2)
exp(LR_A2$coefficients)

hoslem.test(LR_A2$y, LR_A2$fitted.values, g=10)

LR_prob_train <- LR_A2$fitted.values
LR_prob_test <- predict(LR_A2, df_test, type="response")
roc_LR_train = roc(df_train$Anxiety ~ LR_prob_train)
roc_LR_test = roc(df_test$Anxiety ~ LR_prob_test)
rla2 = roc(df_test$Anxiety ~ LR_prob_test)
plot(roc_LR_train, print.auc = TRUE, col = 2, lwd =3,
     print.auc.y=0.98, print.auc.x=0.98,
     auc.polygon = TRUE, max.auc.polygon = TRUE);
plot(roc_LR_test, add=TRUE , print.auc = TRUE, col = 3, lwd =3,
     print.auc.y=0.90, print.auc.x=0.98);
legend(0.33,0.18, c("Train ","Test "),
       col = c(2,3), lwd =2)
roc.test(roc_LR_train, roc_LR_test)

LR_Class_test <- factor(ifelse(LR_prob_test >=0.5,
    "Anxiety", "Negative"),
                        levels= c("Negative","Anxiety"))
CM=confusionMatrix(as.factor(LR_Class_test),
                   df_test$Anxiety,
                   positive = "Anxiety")
paste(round(CM$overall[1],3), "&", round(CM$byClass[1],3), "&",
      round(CM$byClass[2],3), "&", round(CM$byClass[3],3), "&",
      round(CM$byClass[4],3), "&", round(CM$byClass[7],3), "&",
      round(roc_LR_test$auc,3))

#########################################################
# SMILE Domains
LR_A3_1 <- glm(Anxiety ~ SMILE_DN + SMILE_SU + SMILE_PA + SMILE_SM +
                  SMILE_RS + SMILE_SS + SMILE_EE + Gender +
                  Age + Course + School + Displaced,
                family=binomial(link="logit"), data = df_train)
summary(LR_A3_1)
LR_A3_2 = update(LR_A3_1,~. - Displaced)
anova(LR_A3_1,LR_A3_2, test = "Chisq")
summary(LR_A3_2)
LR_A3_3 = update(LR_A3_2,~. - Course)
anova(LR_A3_2,LR_A3_3, test = "Chisq")
summary(LR_A3_3)
```

```
LR_A3_4 = update(LR_A3_3,~. - SMILE_PA)
anova(LR_A3_3,LR_A3_4, test = "Chisq")
summary(LR_A3_4)
LR_A3_5 = update(LR_A3_4,~. - SMILE_EE)
anova(LR_A3_4,LR_A3_5, test = "Chisq")
summary(LR_A3_5)
LR_A3_6 = update(LR_A3_5,~. - Age)
anova(LR_A3_5,LR_A3_6, test = "Chisq")
summary(LR_A3_6)
LR_A3_7 = update(LR_A3_6,~. - SMILE_SS)
anova(LR_A3_6,LR_A3_7, test = "Chisq")
summary(LR_A3_7)
LR_A3_8 = update(LR_A3_7,~. - SMILE_SU)
anova(LR_A3_7,LR_A3_8, test = "Chisq")
anova(LR_A3_1,LR_A3_8, test = "Chisq")
LR_A3 <- LR_A3_8
summary(LR_A3)
coefficients(LR_A3)
exp(LR_A3$coefficients)

hoslem.test(LR_A3$y, LR_A3$fitted.values, g=10)

LR_prob_train <- LR_A3$fitted.values
LR_prob_test <- predict(LR_A3, df_test, type="response")
roc_LR_train = roc(df_train$Anxiety ~ LR_prob_train)
roc_LR_test = roc(df_test$Anxiety ~ LR_prob_test)
rla3 = roc(df_test$Anxiety ~ LR_prob_test)
plot(roc_LR_train, print.auc = TRUE, col = 2, lwd =3,
     print.auc.y=0.98, print.auc.x=0.98,
     auc.polygon = TRUE, max.auc.polygon = TRUE);
plot(roc_LR_test, add=TRUE , print.auc = TRUE, col = 3, lwd =3,
     print.auc.y=0.90, print.auc.x=0.98);
legend(0.33,0.18, c("Train ","Test "),
       col = c(2,3), lwd =2)
roc.test(roc_LR_train, roc_LR_test)

LR_Class_test <- factor(ifelse(LR_prob_test >=0.5,
    "Anxiety", "Negative"),
                        levels= c("Negative","Anxiety"))
CM=confusionMatrix(as.factor(LR_Class_test),
                   df_test$Anxiety,
                   positive = "Anxiety")
paste(round(CM$overall[1],3), "&", round(CM$byClass[1],3), "&",
      round(CM$byClass[2],3), "&", round(CM$byClass[3],3), "&",
      round(CM$byClass[4],3), "&", round(CM$byClass[7],3), "&",
      round(roc_LR_test$auc,3))

#########################################################
# Logistic regression - whith oversampling
```

```
#########################################################
### Oversampling
table(df_train$Anxiety)
set.seed(123)
df_train_O <- RandOverClassif(Anxiety~., df_train, "balance")
table(df_train_O$Anxiety)

#########################################################
# SMILE: Logistic regression - whith oversampling
LR_A1_1 <- glm(Anxiety ~ SMILE + Gender + Age + Course +
                 School + Displaced,
               family=binomial(link="logit"), data = df_train_O)
summary(LR_A1_1)
LR_A1_2 = update(LR_A1_1,~. - Displaced)
anova(LR_A1_1,LR_A1_2, test = "Chisq")
summary(LR_A1_2)
LR_A1_3 = update(LR_A1_2,~. - Course)
anova(LR_A1_2,LR_A1_3, test = "Chisq")
summary(LR_A1_3)
LR_A1_4 = update(LR_A1_3,~. - Age)
anova(LR_A1_3,LR_A1_4, test = "Chisq")
anova(LR_A1_1,LR_A1_4, test = "Chisq")
LR_A1 <- LR_A1_4
summary(LR_A1)
coefficients(LR_A1)
exp(LR_A1$coefficients)

hoslem.test(LR_A1$y, LR_A1$fitted.values, g=10)

LR_prob_train <- LR_A1$fitted.values
LR_prob_test <- predict(LR_A1, df_test, type="response")
roc_LR_train = roc(df_train_O$Anxiety ~ LR_prob_train)
roc_LR_test = roc(df_test$Anxiety ~ LR_prob_test)
rla1_O = roc(df_test$Anxiety ~ LR_prob_test)
plot(roc_LR_train, print.auc = TRUE, col = 2, lwd =3,
     print.auc.y=0.98, print.auc.x=0.98,
     auc.polygon = TRUE, max.auc.polygon = TRUE);
plot(roc_LR_test, add=TRUE , print.auc = TRUE, col = 3, lwd =3,
     print.auc.y=0.90, print.auc.x=0.98);
legend(0.33,0.18, c("Train ","Test "),
       col = c(2,3), lwd =2)
roc.test(roc_LR_train, roc_LR_test)

LR_Class_test <- factor(ifelse(LR_prob_test >=0.5,
    "Anxiety", "Negative"),
                        levels= c("Negative","Anxiety"))
CM=confusionMatrix(as.factor(LR_Class_test),
                   df_test$Anxiety,
```

```
                    positive = "Anxiety")
paste(round(CM$overall[1],3), "&", round(CM$byClass[1],3), "&",
      round(CM$byClass[2],3), "&", round(CM$byClass[3],3), "&",
      round(CM$byClass[4],3), "&", round(CM$byClass[7],3), "&",
      round(roc_LR_test$auc,3))

#########################################################
# SMILE-C: Logistic regression - whith oversampling
LR_A2_1 <- glm(Anxiety ~ SMILEC + Gender + Age + Course +
                 School + Displaced,
               family=binomial(link="logit"), data = df_train_O)
summary(LR_A2_1)
LR_A2_2 = update(LR_A2_1,~. - Displaced)
anova(LR_A2_1,LR_A2_2, test = "Chisq")
summary(LR_A2_2)
LR_A2_3 = update(LR_A2_2,~. - Course)
anova(LR_A2_2,LR_A2_3, test = "Chisq")
summary(LR_A2_3)
LR_A2 <- LR_A2_3
summary(LR_A2)
coefficients(LR_A2)
exp(LR_A2$coefficients)

hoslem.test(LR_A2$y, LR_A2$fitted.values, g=10)

LR_prob_train <- LR_A2$fitted.values
LR_prob_test <- predict(LR_A2, df_test, type="response")
roc_LR_train = roc(df_train_O$Anxiety ~ LR_prob_train)
roc_LR_test = roc(df_test$Anxiety ~ LR_prob_test)
rla2_O = roc(df_test$Anxiety ~ LR_prob_test)
plot(roc_LR_train, print.auc = TRUE, col = 2, lwd =3,
     print.auc.y=0.98, print.auc.x=0.98,
     auc.polygon = TRUE, max.auc.polygon = TRUE);
plot(roc_LR_test, add=TRUE , print.auc = TRUE, col = 3, lwd =3,
     print.auc.y=0.90, print.auc.x=0.98);
legend(0.33,0.18, c("Train ","Test "),
       col = c(2,3), lwd =2)
roc.test(roc_LR_train, roc_LR_test)

LR_Class_test <- factor(ifelse(LR_prob_test >=0.5,
    "Anxiety", "Negative"),
                        levels= c("Negative","Anxiety"))
CM=confusionMatrix(as.factor(LR_Class_test),
                   df_test$Anxiety,
                   positive = "Anxiety")
paste(round(CM$overall[1],3), "&", round(CM$byClass[1],3), "&",
      round(CM$byClass[2],3), "&", round(CM$byClass[3],3), "&",
      round(CM$byClass[4],3), "&", round(CM$byClass[7],3), "&",
      round(roc_LR_test$auc,3))
```

```
#######################################################
# SMILE Domains: Logistic regression - whith oversampling
LR_A3_1 <- glm(Anxiety ~ SMILE_DN + SMILE_SU + SMILE_PA + SMILE_SM +
                 SMILE_RS + SMILE_SS + SMILE_EE + Gender +
                 Age + Course + School + Displaced,
               family=binomial(link="logit"), data = df_train_O)
summary(LR_A3_1)
LR_A3_2 = update(LR_A3_1,~. - Displaced)
anova(LR_A3_1,LR_A3_2, test = "Chisq")
summary(LR_A3_2)
LR_A3_3 = update(LR_A3_2,~. - SMILE_EE)
anova(LR_A3_2,LR_A3_3, test = "Chisq")
summary(LR_A3_3)
LR_A3_4 = update(LR_A3_3,~. - SMILE_PA)
anova(LR_A3_3,LR_A3_4, test = "Chisq")
summary(LR_A3_4)
LR_A3_5 = update(LR_A3_4,~. - SMILE_SU)
anova(LR_A3_4,LR_A3_5, test = "Chisq")
summary(LR_A3_5)
LR_A3_6 = update(LR_A3_5,~. - Age)
anova(LR_A3_5,LR_A3_6, test = "Chisq")
summary(LR_A3_6)
LR_A3_7 = update(LR_A3_6,~. - SMILE_SS)
anova(LR_A3_6,LR_A3_7, test = "Chisq")
summary(LR_A3_7)
LR_A3_8 = update(LR_A3_7,~. - Course)
anova(LR_A3_7,LR_A3_8, test = "Chisq")
summary(LR_A3_8)
anova(LR_A3_1,LR_A3_8, test = "Chisq")
LR_A3 <- LR_A3_8
summary(LR_A3)
coefficients(LR_A3)
exp(LR_A3$coefficients)

hoslem.test(LR_A3$y, LR_A3$fitted.values, g=10)

LR_prob_train <- LR_A3$fitted.values
LR_prob_test <- predict(LR_A3, df_test, type="response")
roc_LR_train = roc(df_train_O$Anxiety ~ LR_prob_train)
roc_LR_test = roc(df_test$Anxiety ~ LR_prob_test)
rla3_O = roc(df_test$Anxiety ~ LR_prob_test)
plot(roc_LR_train, print.auc = TRUE, col = 2, lwd =3,
     print.auc.y=0.98, print.auc.x=0.98,
     auc.polygon = TRUE, max.auc.polygon = TRUE);
plot(roc_LR_test, add=TRUE , print.auc = TRUE, col = 3, lwd =3,
     print.auc.y=0.90, print.auc.x=0.98);
legend(0.33,0.18, c("Train ","Test "),
       col = c(2,3), lwd =2)
```

```
roc.test(roc_LR_train, roc_LR_test)

LR_Class_test <- factor(ifelse(LR_prob_test >=0.5,
    "Anxiety", "Negative"),
                        levels= c("Negative","Anxiety"))
CM=confusionMatrix(as.factor(LR_Class_test),
                   df_test$Anxiety,
                   positive = "Anxiety")
paste(round(CM$overall[1],3), "&", round(CM$byClass[1],3), "&",
      round(CM$byClass[2],3), "&", round(CM$byClass[3],3), "&",
      round(CM$byClass[4],3), "&", round(CM$byClass[7],3), "&",
      round(roc_LR_test$auc,3))


# ROC summary
Cores <- c("red", "orange",  "purple", "firebrick",
    "blue", "darkturquoise")
plot(rla1, print.auc = TRUE, col = Cores[1], lwd =3,
     print.auc.y=0.98, print.auc.x=0.98,
     auc.polygon = TRUE, max.auc.polygon = TRUE);
plot(rla2, add=TRUE , print.auc = TRUE, col = Cores[2], lwd =3,
     print.auc.y=0.93, print.auc.x=0.98)
plot(rla3, add=TRUE , print.auc = TRUE, col = Cores[3], lwd =3,
     print.auc.y=0.88, print.auc.x=0.98)
plot(rla1_O, add=TRUE , print.auc = TRUE, col = Cores[4], lwd =3,
     print.auc.y=0.83, print.auc.x=0.98)
plot(rla2_O, add=TRUE , print.auc = TRUE, col = Cores[5], lwd =3,
     print.auc.y=0.78, print.auc.x=0.98)
plot(rla3_O, add=TRUE , print.auc = TRUE, col = Cores[6], lwd =3,
     print.auc.y=0.73, print.auc.x=0.98)
legend(0.45,0.3,
       c("AD_LR_S ","AD_LR_SC ","AD_LR_SD",
         "AD_LR_S_O ","AD_LR_SC_O ","AD_LR_SD_O "),
       col = Cores, lwd =3)


#########################################################
##### Multivariate analysis to screen for Depresson

#########################################################
# Logistic regression - whithout oversampling

#########################################################
# SMILE
LR_D1_1 <- glm(Depression ~ SMILE + Gender + Age + Course +
                 School + Displaced,
               family=binomial(link="logit"), data = df_train)
summary(LR_D1_1)
LR_D1_2 = update(LR_D1_1,~. - Displaced)
```

```
anova(LR_D1_1,LR_D1_2, test = "Chisq")
summary(LR_D1_2)
LR_D1 <- LR_D1_2
summary(LR_D1)
coefficients(LR_D1)
exp(LR_D1$coefficients)

hoslem.test(LR_D1$y, LR_D1$fitted.values, g=10)

LR_prob_train <- LR_D1$fitted.values
LR_prob_test <- predict(LR_D1, df_test, type="response")
roc_LR_train = roc(df_train$Depression ~ LR_prob_train)
roc_LR_test = roc(df_test$Depression ~ LR_prob_test)
rld1 = roc(df_test$Depression ~ LR_prob_test)
plot(roc_LR_train, print.auc = TRUE, col = 2, lwd =3,
     print.auc.y=0.98, print.auc.x=0.98,
     auc.polygon = TRUE, max.auc.polygon = TRUE);
plot(roc_LR_test, add=TRUE , print.auc = TRUE, col = 3, lwd =3,
     print.auc.y=0.90, print.auc.x=0.98);
legend(0.33,0.18, c("Train ","Test "),
       col = c(2,3), lwd =2)
roc.test(roc_LR_train, roc_LR_test)

LR_Class_test <- factor(ifelse(LR_prob_test >=0.5,
    "Depression", "Negative"),
                        levels= c("Negative","Depression"))
CM=confusionMatrix(as.factor(LR_Class_test),
                   df_test$Depression,
                   positive = "Depression")
paste(round(CM$overall[1],3), "&", round(CM$byClass[1],3), "&",
      round(CM$byClass[2],3), "&", round(CM$byClass[3],3), "&",
      round(CM$byClass[4],3), "&", round(CM$byClass[7],3), "&",
      round(roc_LR_test$auc,3))

#########################################################
# SMILE-C
LR_D2_1 <- glm(Depression ~ SMILEC + Gender + Age + Course +
                 School + Displaced,
               family=binomial(link="logit"), data = df_train)
summary(LR_D2_1)
LR_D2_2 = update(LR_D2_1,~. - Displaced)
anova(LR_D2_1,LR_D2_2, test = "Chisq")
summary(LR_D2_2)
LR_D2_3 = update(LR_D2_2,~. - Gender)
anova(LR_D2_2,LR_D2_3, test = "Chisq")
summary(LR_D2_3)
LR_D2 <- LR_D2_3
summary(LR_D2)
coefficients(LR_D2)
```

```
exp(LR_D2$coefficients)

hoslem.test(LR_D2$y, LR_D2$fitted.values, g=10)

LR_prob_train <- LR_D2$fitted.values
LR_prob_test <- predict(LR_D2, df_test, type="response")
roc_LR_train = roc(df_train$Depression ~ LR_prob_train)
roc_LR_test = roc(df_test$Depression ~ LR_prob_test)
rld2 = roc(df_test$Depression ~ LR_prob_test)
plot(roc_LR_train, print.auc = TRUE, col = 2, lwd =3,
     print.auc.y=0.98, print.auc.x=0.98,
     auc.polygon = TRUE, max.auc.polygon = TRUE);
plot(roc_LR_test, add=TRUE , print.auc = TRUE, col = 3, lwd =3,
     print.auc.y=0.90, print.auc.x=0.98);
legend(0.33,0.18, c("Train ","Test "),
       col = c(2,3), lwd =2)
roc.test(roc_LR_train, roc_LR_test)

LR_Class_test <- factor(ifelse(LR_prob_test >=0.5,
    "Depression", "Negative"),
                        levels= c("Negative","Depression"))
CM=confusionMatrix(as.factor(LR_Class_test),
                   df_test$Depression,
                   positive = "Depression")
paste(round(CM$overall[1],3), "&", round(CM$byClass[1],3), "&",
      round(CM$byClass[2],3), "&", round(CM$byClass[3],3), "&",
      round(CM$byClass[4],3), "&", round(CM$byClass[7],3), "&",
      round(roc_LR_test$auc,3))

#########################################################
# SMILE Domains
LR_D3_1 <- glm(Depression ~ SMILE_DN + SMILE_SU + SMILE_PA +
    SMILE_SM + SMILE_RS + SMILE_SS + SMILE_EE +
    Gender + Age + Course + School + Displaced,
                family=binomial(link="logit"), data = df_train)
summary(LR_D3_1)
LR_D3_2 = update(LR_D3_1,~. - Displaced)
anova(LR_D3_1,LR_D3_2, test = "Chisq")
summary(LR_D3_2)
LR_D3_3 = update(LR_D3_2,~. - SMILE_DN)
anova(LR_D3_2,LR_D3_3, test = "Chisq")
summary(LR_D3_3)
LR_D3_4 = update(LR_D3_3,~. - SMILE_SU)
anova(LR_D3_3,LR_D3_4, test = "Chisq")
summary(LR_D3_4)
LR_D3_5 = update(LR_D3_4,~. - SMILE_PA)
anova(LR_D3_4,LR_D3_5, test = "Chisq")
summary(LR_D3_5)
LR_D3 <- LR_D3_5
```

```
summary(LR_D3)
coefficients(LR_D3)
exp(LR_D3$coefficients)

hoslem.test(LR_D3$y, LR_D3$fitted.values, g=10)

LR_prob_train <- LR_D3$fitted.values
LR_prob_test <- predict(LR_D3, df_test, type="response")
roc_LR_train = roc(df_train$Depression ~ LR_prob_train)
roc_LR_test = roc(df_test$Depression ~ LR_prob_test)
rld3 = roc(df_test$Depression ~ LR_prob_test)
plot(roc_LR_train, print.auc = TRUE, col = 2, lwd =3,
     print.auc.y=0.98, print.auc.x=0.98,
     auc.polygon = TRUE, max.auc.polygon = TRUE);
plot(roc_LR_test, add=TRUE , print.auc = TRUE, col = 3, lwd =3,
     print.auc.y=0.90, print.auc.x=0.98);
legend(0.33,0.18, c("Train ","Test "),
       col = c(2,3), lwd =2)
roc.test(roc_LR_train, roc_LR_test)

LR_Class_test <- factor(ifelse(LR_prob_test >=0.5,
    "Depression", "Negative"),
                          levels= c("Negative","Depression"))
CM=confusionMatrix(as.factor(LR_Class_test),
                   df_test$Depression,
                   positive = "Depression")
paste(round(CM$overall[1],3), "&", round(CM$byClass[1],3), "&",
      round(CM$byClass[2],3), "&", round(CM$byClass[3],3), "&",
      round(CM$byClass[4],3), "&", round(CM$byClass[7],3), "&",
      round(roc_LR_test$auc,3))

#########################################################
# Logistic regression - whith oversampling
#########################################################
### Oversampling
table(df_train$Depression)
set.seed(123)
df_train_O <- RandOverClassif(Depression~., df_train, "balance")
table(df_train_O$Depression)

#########################################################
# SMILE: Logistic regression - whith oversampling
LR_D1_1 <- glm(Depression ~ SMILE + Gender + Age + Course +
                 School + Displaced,
               family=binomial(link="logit"), data = df_train_O)
summary(LR_D1_1)
LR_D1 <- LR_D1_1
summary(LR_D1)
coefficients(LR_D1)
```

```
exp(LR_D1$coefficients)

hoslem.test(LR_D1$y, LR_D1$fitted.values, g=10)

LR_prob_train <- LR_D1$fitted.values
LR_prob_test <- predict(LR_D1, df_test, type="response")
roc_LR_train = roc(df_train_O$Depression ~ LR_prob_train)
roc_LR_test = roc(df_test$Depression ~ LR_prob_test)
rld1_O = roc(df_test$Depression ~ LR_prob_test)
plot(roc_LR_train, print.auc = TRUE, col = 2, lwd =3,
     print.auc.y=0.98, print.auc.x=0.98,
     auc.polygon = TRUE, max.auc.polygon = TRUE);
plot(roc_LR_test, add=TRUE , print.auc = TRUE, col = 3, lwd =3,
     print.auc.y=0.90, print.auc.x=0.98);
legend(0.33,0.18, c("Train ","Test "),
       col = c(2,3), lwd =2)
roc.test(roc_LR_train, roc_LR_test)

LR_Class_test <- factor(ifelse(LR_prob_test >=0.5,
    "Depression", "Negative"),
                        levels= c("Negative","Depression"))
CM=confusionMatrix(as.factor(LR_Class_test),
                   df_test$Depression,
                   positive = "Depression")
paste(round(CM$overall[1],3), "&", round(CM$byClass[1],3), "&",
      round(CM$byClass[2],3), "&", round(CM$byClass[3],3), "&",
      round(CM$byClass[4],3), "&", round(CM$byClass[7],3), "&",
      round(roc_LR_test$auc,3))

#########################################################
# SMILE-C: Logistic regression - whith oversampling
LR_D2_1 <- glm(Depression ~ SMILEC + Gender + Age + Course +
                 School + Displaced,
               family=binomial(link="logit"), data = df_train_O)
summary(LR_D2_1)
LR_D2_2 = update(LR_D2_1,~. - Gender)
anova(LR_D2_1,LR_D2_2, test = "Chisq")
summary(LR_D2_2)
LR_D2 <- LR_D2_2
summary(LR_D2)
coefficients(LR_D2)
exp(LR_D2$coefficients)

hoslem.test(LR_D2$y, LR_D2$fitted.values, g=10)

LR_prob_train <- LR_D2$fitted.values
LR_prob_test <- predict(LR_D2, df_test, type="response")
roc_LR_train = roc(df_train_O$Depression ~ LR_prob_train)
roc_LR_test = roc(df_test$Depression ~ LR_prob_test)
```

```
rld2_O = roc(df_test$Depression ~ LR_prob_test)
plot(roc_LR_train, print.auc = TRUE, col = 2, lwd =3,
     print.auc.y=0.98, print.auc.x=0.98,
     auc.polygon = TRUE, max.auc.polygon = TRUE);
plot(roc_LR_test, add=TRUE , print.auc = TRUE, col = 3, lwd =3,
     print.auc.y=0.90, print.auc.x=0.98);
legend(0.33,0.18, c("Train ","Test "),
       col = c(2,3), lwd =2)
roc.test(roc_LR_train, roc_LR_test)

LR_Class_test <- factor(ifelse(LR_prob_test >=0.5,
    "Depression", "Negative"),
                        levels= c("Negative","Depression"))
CM=confusionMatrix(as.factor(LR_Class_test),
                   df_test$Depression,
                   positive = "Depression")
paste(round(CM$overall[1],3), "&", round(CM$byClass[1],3), "&",
      round(CM$byClass[2],3), "&", round(CM$byClass[3],3), "&",
      round(CM$byClass[4],3), "&", round(CM$byClass[7],3), "&",
      round(roc_LR_test$auc,3))

#########################################################
# SMILE Domains: Logistic regression - whith oversampling
LR_D3_1 <- glm(Depression ~ SMILE_DN + SMILE_SU + SMILE_PA +
    SMILE_SM + SMILE_RS + SMILE_SS + SMILE_EE +
    Gender + Age + Course + School + Displaced,
                family=binomial(link="logit"), data = df_train_O)
summary(LR_D3_1)
LR_D3_2 = update(LR_D3_1,~. - Displaced)
anova(LR_D3_1,LR_D3_2, test = "Chisq")
summary(LR_D3_2)
LR_D3_3 = update(LR_D3_2,~. - SMILE_DN)
anova(LR_D3_2,LR_D3_3, test = "Chisq")
summary(LR_D3_3)
LR_D3_4 = update(LR_D3_3,~. - SMILE_PA)
anova(LR_D3_3,LR_D3_4, test = "Chisq")
summary(LR_D3_4)
LR_D3_5 = update(LR_D3_4,~. - SMILE_SU)
anova(LR_D3_4,LR_D3_5, test = "Chisq")
summary(LR_D3_5)
LR_D3 <- LR_D3_5
summary(LR_D3)
coefficients(LR_D3)
exp(LR_D3$coefficients)

hoslem.test(LR_D3$y, LR_D3$fitted.values, g=10)

LR_prob_train <- LR_D3$fitted.values
LR_prob_test <- predict(LR_D3, df_test, type="response")
```

```
roc_LR_train = roc(df_train_O$Depression ~ LR_prob_train)
roc_LR_test = roc(df_test$Depression ~ LR_prob_test)
rld3_O = roc(df_test$Depression ~ LR_prob_test)
plot(roc_LR_train, print.auc = TRUE, col = 2, lwd =3,
     print.auc.y=0.98, print.auc.x=0.98,
     auc.polygon = TRUE, max.auc.polygon = TRUE);
plot(roc_LR_test, add=TRUE , print.auc = TRUE, col = 3, lwd =3,
     print.auc.y=0.90, print.auc.x=0.98);
legend(0.33,0.18, c("Train ","Test "),
       col = c(2,3), lwd =2)
roc.test(roc_LR_train, roc_LR_test)

LR_Class_test <- factor(ifelse(LR_prob_test >=0.5,
    "Depression", "Negative"),
                        levels= c("Negative","Depression"))
CM=confusionMatrix(as.factor(LR_Class_test),
                   df_test$Depression,
                   positive = "Depression")
paste(round(CM$overall[1],3), "&", round(CM$byClass[1],3), "&",
      round(CM$byClass[2],3), "&", round(CM$byClass[3],3), "&",
      round(CM$byClass[4],3), "&", round(CM$byClass[7],3), "&",
      round(roc_LR_test$auc,3))


# ROC summary
Cores <- c("red", "orange",  "purple", "firebrick",
    "blue", "darkturquoise")
plot(rld1, print.auc = TRUE, col = Cores[1], lwd =3,
     print.auc.y=0.98, print.auc.x=0.98,
     auc.polygon = TRUE, max.auc.polygon = TRUE);
plot(rld2, add=TRUE , print.auc = TRUE, col = Cores[2], lwd =3,
     print.auc.y=0.93, print.auc.x=0.98)
plot(rld3, add=TRUE , print.auc = TRUE, col = Cores[3], lwd =3,
     print.auc.y=0.88, print.auc.x=0.98)
plot(rld1_O, add=TRUE , print.auc = TRUE, col = Cores[4], lwd =3,
     print.auc.y=0.83, print.auc.x=0.98)
plot(rld2_O, add=TRUE , print.auc = TRUE, col = Cores[5], lwd =3,
     print.auc.y=0.78, print.auc.x=0.98)
plot(rld3_O, add=TRUE , print.auc = TRUE, col = Cores[6], lwd =3,
     print.auc.y=0.73, print.auc.x=0.98)
legend(0.45,0.3,
       c("AD_LR_S ","AD_LR_SC ","AD_LR_SD",
         "AD_LR_S_O ","AD_LR_SC_O ","AD_LR_SD_O "),
       col = Cores, lwd =3)


##########################################################
##### Multivariate analysis to screen for Anxiety and Depression
```

```
#########################################################
# Logistic regression - whithout oversampling

#########################################################
# SMILE
LR_AD1_1 <- glm(Anxiety_Depression ~ SMILE + Gender + Age + Course +
                  School + Displaced,
                family=binomial(link="logit"), data = df_train)
summary(LR_AD1_1)
LR_AD1_2 = update(LR_AD1_1,~. - Displaced)
anova(LR_AD1_1,LR_AD1_2, test = "Chisq")
summary(LR_AD1_2)
LR_AD1 <- LR_AD1_2
summary(LR_AD1)
coefficients(LR_AD1)
exp(LR_AD1$coefficients)

hoslem.test(LR_AD1$y, LR_AD1$fitted.values, g=10)

LR_prob_train <- LR_AD1$fitted.values
LR_prob_test <- predict(LR_AD1, df_test, type="response")
roc_LR_train = roc(df_train$Anxiety_Depression ~ LR_prob_train)
roc_LR_test = roc(df_test$Anxiety_Depression ~ LR_prob_test)
rlad1 = roc(df_test$Anxiety_Depression ~ LR_prob_test)
plot(roc_LR_train, print.auc = TRUE, col = 2, lwd =3,
     print.auc.y=0.98, print.auc.x=0.98,
     auc.polygon = TRUE, max.auc.polygon = TRUE);
plot(roc_LR_test, add=TRUE , print.auc = TRUE, col = 3, lwd =3,
     print.auc.y=0.90, print.auc.x=0.98);
legend(0.33,0.18, c("Train ","Test "),
       col = c(2,3), lwd =2)
roc.test(roc_LR_train, roc_LR_test)

LR_Class_test <- factor(ifelse(LR_prob_test >=0.5,
    "Anxiety and Depression", "Negative"),
                        levels= c("Negative","Anxiety and Depression"))
CM=confusionMatrix(as.factor(LR_Class_test),
                   df_test$Anxiety_Depression,
                   positive = "Anxiety and Depression")
paste(round(CM$overall[1],3), "&", round(CM$byClass[1],3), "&",
      round(CM$byClass[2],3), "&", round(CM$byClass[3],3), "&",
      round(CM$byClass[4],3), "&", round(CM$byClass[7],3), "&",
      round(roc_LR_test$auc,3))

#########################################################
# SMILE-C
LR_AD2_1 <- glm(Anxiety_Depression ~ SMILEC + Gender + Age + Course +
                  School + Displaced,
                family=binomial(link="logit"), data = df_train)
```

```
summary(LR_AD2_1)
LR_AD2_2 = update(LR_AD2_1,~. - Displaced)
anova(LR_AD2_1,LR_AD2_2, test = "Chisq")
summary(LR_AD2_2)
LR_AD2 <- LR_AD2_2
summary(LR_AD2)
coefficients(LR_AD2)
exp(LR_AD2$coefficients)

hoslem.test(LR_AD2$y, LR_AD2$fitted.values, g=10)

LR_prob_train <- LR_AD2$fitted.values
LR_prob_test <- predict(LR_AD2, df_test, type="response")
roc_LR_train = roc(df_train$Anxiety_Depression ~ LR_prob_train)
roc_LR_test = roc(df_test$Anxiety_Depression ~ LR_prob_test)
rlad2 = roc(df_test$Anxiety_Depression ~ LR_prob_test)
plot(roc_LR_train, print.auc = TRUE, col = 2, lwd =3,
     print.auc.y=0.98, print.auc.x=0.98,
     auc.polygon = TRUE, max.auc.polygon = TRUE);
plot(roc_LR_test, add=TRUE , print.auc = TRUE, col = 3, lwd =3,
     print.auc.y=0.90, print.auc.x=0.98);
legend(0.33,0.18, c("Train ","Test "),
       col = c(2,3), lwd =2)
roc.test(roc_LR_train, roc_LR_test)

LR_Class_test <- factor(ifelse(LR_prob_test >=0.5,
    "Anxiety and Depression", "Negative"),
                        levels= c("Negative","Anxiety and Depression"))
CM=confusionMatrix(as.factor(LR_Class_test),
                   df_test$Anxiety_Depression,
                   positive = "Anxiety and Depression")
paste(round(CM$overall[1],3), "&", round(CM$byClass[1],3), "&",
      round(CM$byClass[2],3), "&", round(CM$byClass[3],3), "&",
      round(CM$byClass[4],3), "&", round(CM$byClass[7],3), "&",
      round(roc_LR_test$auc,3))

#########################################################
# SMILE Domains
LR_AD3_1 <- glm(Anxiety_Depression ~ SMILE_DN + SMILE_SU +
    SMILE_PA + SMILE_SM + SMILE_RS + SMILE_SS + SMILE_EE +
    Gender + Age + Course + School + Displaced,
                family=binomial(link="logit"), data = df_train)
summary(LR_AD3_1)
LR_AD3_2 = update(LR_AD3_1,~. - Displaced)
anova(LR_AD3_1,LR_AD3_2, test = "Chisq")
summary(LR_AD3_2)
LR_AD3_3 = update(LR_AD3_2,~. - SMILE_PA)
anova(LR_AD3_2,LR_AD3_3, test = "Chisq")
summary(LR_AD3_3)
```

```
LR_AD3_4 = update(LR_AD3_3,~. - SMILE_EE)
anova(LR_AD3_3,LR_AD3_4, test = "Chisq")
summary(LR_AD3_4)
LR_AD3_5 = update(LR_AD3_4,~. - SMILE_DN)
anova(LR_AD3_4,LR_AD3_5, test = "Chisq")
summary(LR_AD3_5)
LR_AD3 <- LR_AD3_5
summary(LR_AD3)
coefficients(LR_AD3)
exp(LR_AD3$coefficients)

hoslem.test(LR_AD3$y, LR_AD3$fitted.values, g=10)

LR_prob_train <- LR_AD3$fitted.values
LR_prob_test <- predict(LR_AD3, df_test, type="response")
roc_LR_train = roc(df_train$Anxiety_Depression ~ LR_prob_train)
roc_LR_test = roc(df_test$Anxiety_Depression ~ LR_prob_test)
rlad3 = roc(df_test$Anxiety_Depression ~ LR_prob_test)
plot(roc_LR_train, print.auc = TRUE, col = 2, lwd =3,
     print.auc.y=0.98, print.auc.x=0.98,
     auc.polygon = TRUE, max.auc.polygon = TRUE);
plot(roc_LR_test, add=TRUE , print.auc = TRUE, col = 3, lwd =3,
     print.auc.y=0.90, print.auc.x=0.98);
legend(0.33,0.18, c("Train ","Test "),
       col = c(2,3), lwd =2)
roc.test(roc_LR_train, roc_LR_test)

LR_Class_test <- factor(ifelse(LR_prob_test >=0.5,
    "Anxiety and Depression", "Negative"),
                        levels= c("Negative","Anxiety and Depression"))
CM=confusionMatrix(as.factor(LR_Class_test),
                   df_test$Anxiety_Depression,
                   positive = "Anxiety and Depression")
paste(round(CM$overall[1],3), "&", round(CM$byClass[1],3), "&",
      round(CM$byClass[2],3), "&", round(CM$byClass[3],3), "&",
      round(CM$byClass[4],3), "&", round(CM$byClass[7],3), "&",
      round(roc_LR_test$auc,3))

#########################################################
# Logistic regression - whith oversampling

#########################################################
### Oversampling
table(df_train$Anxiety_Depression)
set.seed(123)
df_train_O <- RandOverClassif(Anxiety_Depression~., df_train, "balance")
table(df_train_O$Anxiety_Depression)

#########################################################
```

```
# SMILE: Logistic regression - whith oversampling
LR_AD1_1 <- glm(Anxiety_Depression ~ SMILE + Gender + Age +
    Course + School + Displaced,
                family=binomial(link="logit"), data = df_train_O)
summary(LR_AD1_1)
LR_AD1_2 = update(LR_AD1_1,~. - Displaced)
anova(LR_AD1_1,LR_AD1_2, test = "Chisq")
summary(LR_AD1_2)
LR_AD1 <- LR_AD1_2
summary(LR_AD1)
coefficients(LR_AD1)
exp(LR_AD1$coefficients)

hoslem.test(LR_AD1$y, LR_AD1$fitted.values, g=10)

LR_prob_train <- LR_AD1$fitted.values
LR_prob_test <- predict(LR_AD1, df_test, type="response")
roc_LR_train = roc(df_train_O$Anxiety_Depression ~ LR_prob_train)
roc_LR_test = roc(df_test$Anxiety_Depression ~ LR_prob_test)
rlad1_O = roc(df_test$Anxiety_Depression ~ LR_prob_test)
plot(roc_LR_train, print.auc = TRUE, col = 2, lwd =3,
     print.auc.y=0.98, print.auc.x=0.98,
     auc.polygon = TRUE, max.auc.polygon = TRUE);
plot(roc_LR_test, add=TRUE , print.auc = TRUE, col = 3, lwd =3,
     print.auc.y=0.90, print.auc.x=0.98);
legend(0.33,0.18, c("Train ","Test "),
       col = c(2,3), lwd =2)
roc.test(roc_LR_train, roc_LR_test)

LR_Class_test <- factor(ifelse(LR_prob_test >=0.5,
    "Anxiety and Depression", "Negative"),
                        levels= c("Negative","Anxiety and Depression"))
CM=confusionMatrix(as.factor(LR_Class_test),
                   df_test$Anxiety_Depression,
                   positive = "Anxiety and Depression")
paste(round(CM$overall[1],3), "&", round(CM$byClass[1],3), "&",
      round(CM$byClass[2],3), "&", round(CM$byClass[3],3), "&",
      round(CM$byClass[4],3), "&", round(CM$byClass[7],3), "&",
      round(roc_LR_test$auc,3))

#########################################################
# SMILE-C: Logistic regression - whith oversampling
LR_AD2_1 <- glm(Anxiety_Depression ~ SMILEC + Gender + Age +
    Course + School + Displaced,
                family=binomial(link="logit"), data = df_train_O)
summary(LR_AD2_1)
LR_AD2_2 = update(LR_AD2_1,~. - Displaced)
anova(LR_AD2_1,LR_AD2_2, test = "Chisq")
summary(LR_AD2_2)
```

```
LR_AD2 <- LR_AD2_2
summary(LR_AD2)
coefficients(LR_AD2)
exp(LR_AD2$coefficients)

hoslem.test(LR_AD2$y, LR_AD2$fitted.values, g=10)

LR_prob_train <- LR_AD2$fitted.values
LR_prob_test <- predict(LR_AD2, df_test, type="response")
roc_LR_train = roc(df_train_O$Anxiety_Depression ~ LR_prob_train)
roc_LR_test = roc(df_test$Anxiety_Depression ~ LR_prob_test)
rlad2_O = roc(df_test$Anxiety_Depression ~ LR_prob_test)
plot(roc_LR_train, print.auc = TRUE, col = 2, lwd =3,
     print.auc.y=0.98, print.auc.x=0.98,
     auc.polygon = TRUE, max.auc.polygon = TRUE);
plot(roc_LR_test, add=TRUE , print.auc = TRUE, col = 3, lwd =3,
     print.auc.y=0.90, print.auc.x=0.98);
legend(0.33,0.18, c("Train ","Test "),
       col = c(2,3), lwd =2)
roc.test(roc_LR_train, roc_LR_test)

LR_Class_test <- factor(ifelse(LR_prob_test >=0.5,
    "Anxiety and Depression", "Negative"),
                        levels= c("Negative","Anxiety and Depression"))
CM=confusionMatrix(as.factor(LR_Class_test),
                   df_test$Anxiety_Depression,
                   positive = "Anxiety and Depression")
paste(round(CM$overall[1],3), "&", round(CM$byClass[1],3), "&",
      round(CM$byClass[2],3), "&", round(CM$byClass[3],3), "&",
      round(CM$byClass[4],3), "&", round(CM$byClass[7],3), "&",
      round(roc_LR_test$auc,3))

#########################################################
# SMILE Domains: Logistic regression - whith oversampling
LR_AD3_1 <- glm(Anxiety_Depression ~ SMILE_DN + SMILE_SU +
    SMILE_PA + SMILE_SM + SMILE_RS + SMILE_SS + SMILE_EE +
    Gender + Age + Course + School + Displaced,
                family=binomial(link="logit"), data = df_train_O)
summary(LR_AD3_1)
LR_AD3_2 = update(LR_AD3_1,~. - SMILE_PA)
anova(LR_AD3_1,LR_AD3_2, test = "Chisq")
summary(LR_AD3_2)
LR_AD3_3 = update(LR_AD3_2,~. - SMILE_EE)
anova(LR_AD3_2,LR_AD3_3, test = "Chisq")
summary(LR_AD3_3)
LR_AD3_4 = update(LR_AD3_3,~. - Displaced)
anova(LR_AD3_3,LR_AD3_4, test = "Chisq")
summary(LR_AD3_4)
LR_AD3 <- LR_AD3_4
```

```
summary(LR_AD3)
coefficients(LR_AD3)
exp(LR_AD3$coefficients)

hoslem.test(LR_AD3$y, LR_AD3$fitted.values, g=10)

LR_prob_train <- LR_ADD3$fitted.values
LR_prob_test <- predict(LR_AD3, df_test, type="response")
roc_LR_train = roc(df_train_O$Anxiety_Depression ~ LR_prob_train)
roc_LR_test = roc(df_test$Anxiety_Depression ~ LR_prob_test)
rlad3_O = roc(df_test$Anxiety_Depression ~ LR_prob_test)
plot(roc_LR_train, print.auc = TRUE, col = 2, lwd =3,
     print.auc.y=0.98, print.auc.x=0.98,
     auc.polygon = TRUE, max.auc.polygon = TRUE);
plot(roc_LR_test, add=TRUE , print.auc = TRUE, col = 3, lwd =3,
     print.auc.y=0.90, print.auc.x=0.98);
legend(0.33,0.18, c("Train ","Test "),
       col = c(2,3), lwd =2)
roc.test(roc_LR_train, roc_LR_test)

LR_Class_test <- factor(ifelse(LR_prob_test >=0.5,
    "Anxiety and Depression", "Negative"),
                        levels= c("Negative","Anxiety and Depression"))
CM=confusionMatrix(as.factor(LR_Class_test),
                   df_test$Anxiety_Depression,
                   positive = "Anxiety and Depression")
paste(round(CM$overall[1],3), "&", round(CM$byClass[1],3), "&",
      round(CM$byClass[2],3), "&", round(CM$byClass[3],3), "&",
      round(CM$byClass[4],3), "&", round(CM$byClass[7],3), "&",
      round(roc_LR_test$auc,3))


# ROC summary
Cores <- c("red", "orange",  "purple", "firebrick",
    "blue", "darkturquoise")
plot(rlad1, print.auc = TRUE, col = Cores[1], lwd =3,
     print.auc.y=0.98, print.auc.x=0.98,
     auc.polygon = TRUE, max.auc.polygon = TRUE);
plot(rlad2, add=TRUE , print.auc = TRUE, col = Cores[2], lwd =3,
     print.auc.y=0.93, print.auc.x=0.98)
plot(rlad3, add=TRUE , print.auc = TRUE, col = Cores[3], lwd =3,
     print.auc.y=0.88, print.auc.x=0.98)
plot(rlad1_O, add=TRUE , print.auc = TRUE, col = Cores[4], lwd =3,
     print.auc.y=0.83, print.auc.x=0.98)
plot(rlad2_O, add=TRUE , print.auc = TRUE, col = Cores[5], lwd =3,
     print.auc.y=0.78, print.auc.x=0.98)
plot(rlad3_O, add=TRUE , print.auc = TRUE, col = Cores[6], lwd =3,
     print.auc.y=0.73, print.auc.x=0.98)
legend(0.45,0.3,
```

```
c("AD_LR_S ","AD_LR_SC ","AD_LR_SD",
  "AD_LR_S_O ","AD_LR_SC_O ","AD_LR_SD_O "),
col = Cores, lwd =3)
```