NOVA IMS Information Management School

MAA

Mestrado em Métodos Analíticos Avançados
Master Program in Advanced Analytics

Predicting academic performance

A practical study using Moodle log data and sociodemographic traits

Nuno Alexandre Lopes Do Rosário

Project Work presented as partial requirement for obtaining the Master's degree in Data Science and Advanced Analytics

NOVA Information Management School
Instituto Superior de Estatística e Gestão de Informação
Universidade Nova de Lisboa

NOVA Information Management School
Instituto Superior de Estatística e Gestão de Informação
Universidade Nova de Lisboa

PREDICTING ACADEMIC PERFORMANCE: A PRACTICAL STUDY USING MOODLE LOG DATA AND SOCIODEMOGRAPHIC TRAITS

by

Nuno Alexandre Lopes Do Rosário

Project Work presented as partial requirement for obtaining the Master's degree in Data Science and Advanced Analytics

Advisor / Co Advisor: Roberto Henriques, PhD

September 2021

ABSTRACT

With the increase of computational power, usage of IT systems and tools in several industries also increased the amounts of data generated and stored. Education is one of these fields. The opportunity to utilize analytical and data related techniques to the data generated and stored by computer-based educational systems is more significant than ever. Performance prediction is one of the most popular uses for all the data generated by educational systems.

In this line of thought, the main objective of this paper is to build a predictive model capable of classifying a student´s grade based on its Moodle system activity and several sociodemographic variables taken from the Netpa System. All the data used belongs to student´s that attended the first semester of 2019 at Nova Information Management School.

To achieve the objective, SEMMA Methodology was implemented. Python Language was used, with particular emphasis on the Scikit-Learn, pandas and Seaborn packages. Raw Moodle logs were processed and transformed into variables that represented the number of times a student navigated to a specific page in the platform. This information was then joined with Netpa variables, and a dataset was built. Exploratory data analysis was performed, and several model configurations were tested. The main differences that separate the models are outlier treatment, sampling techniques, feature scalers, feature engineering and type of algorithm – Logistic Regression, K-Neighbours Classifier, Random Forest Classifier and Multi-Layer Perceptron.

Using a K-Neighbours Classifier and the SMOTE sampling technique an F1-Score of 0.624 and a ROC AUC of 0.828 was obtained.

KEYWORDS

Performance Prediction; Educational Data Mining; Learning Analytics; Machine Learning.

INDEX

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

1.1. CONTEXT AND RELEVANCE

In the last decade, Information Technology systems had a big improvement in almost every performance metric. This means more computational power, storage space and more. Plus, has technology advances, more people have access to them. The use of Information Technology systems in education is not a new paradigm but with the evolution prior described, a lot more data is being collected and stored based on the students learning experience and interaction with the IT systems (Romero, Ventura & García, 2007) (Cristóbal Romero & Ventura, 2010) (Villagrá-Arnedo, Gallego-Durán, Compañ-Rosique, LLorens-Largo & Molina-Carmona, 2016).

Researchers soon found an opportunity to explore and use this data with the objective of studying educational questions. Therefore, two communities and research areas were created around the educational topic: Educational Data Mining and Learning Analytics (Peña-Ayala, 2014) (Popescu & Leon, 2018).

Performance prediction is one of the major topics researched by these two communities. The ability to predict a student's grade can provide support in course selection and design, help monitoring students in order to provide the needed support and integrate training programs to obtain the best results (Li & Liu, 2021). Several work has been developed in the area showing great results. Despite this, almost no work was developed with the general aspect of performance prediction in mind. This means that, although a lot of research has been developed and with great results, it can´t be easily replicated. Several studies use specific variables, like homework data, that are only obtainable by performing a close study (Minaei-Bidgoli, Kashy, Kortemeyer & Punch, 2003) (Lu, Anna Huang, Jeff Huang, Lin, Ogata & Yang, 2018). In addition, almost all the studies are done in a small student sample composed of only one course (Calvo-Flores, Galindo, Jiménez & Piñeiro, 2006) (Ibrahim & Rusli, 2007) (Macfadyen & Dawson, 2010) and with the label variable only assuming two possible values, pass and fail (Calvo-Flores, Galindo, Jiménez & Piñeiro, 2006).

1.2. OBJECTIVE

To address these issues this study aims to develop a predictive model capable of predicting student's grades based on their Moodle activity and sociodemographic traits.

Moodle log data and Netpa system sociodemographic data from 3226 Nova IMS first semester students was used. This way, the variables created are replicable by every institution that uses Moodle, one of the most used Learning Management Systems, and several courses were taken into consideration. The label variable was also classified according to 4 different labels, making it possible to separate the students that pass but have a bad performance and the ones that have an excellent performance.

This work can help understand if the types of data used have any predictive power in terms of grades and, if they do, can be used as a starting point by tutors in order to predict their student grades. Furthermore, an extensive exploratory data analysis was done that can bring detailed insights on the students. When preparing the data for modelling and building the model itself, several different techniques and algorithms were studied, the different results are documented in this report.

1.3. STUDY OUTLINE

The first chapter presents the context and relevance, main objective of this study and the outline where a brief description of the different chapters is done.

The second chapter presents the literature review on educational data mining, learning analytics and learning management systems. Furthermore, a review on performance prediction was done. This allows for identifying the most relevant techniques used and results obtained in the field of study.

The third chapter presents the methodology used in this study. It explains the different techniques used, from data collection to the assessment of the final model. All the relevant decisions made during this chapter are documented and explained in this report.

The fourth chapter analyses the results obtained by the implementation of the methodology mentioned in the third chapter. The best model configurations are examined as well as the different techniques that achieved better results.

The final two chapters are the conclusions and limitations. The first gives a summary of the work developed and the most relevant insights taken from it. The last one shows the several constraints faced while developing these types of projects.

2. LITERATURE REVIEW

2.1. EDUCATIONAL DATA MINING

Technology is in our lives more than ever. With the increased use of technological devices there is an even bigger increase in the amount of data that is generated and stored. This data can then be analyzed and interpreted to provide new insights to stakeholders across several industries.

Peña-Ayala (2014) describes Data Mining as the well-known process that extracts new, useful, and comprehensible knowledge from huge and heterogeneous data repositories.

Today even more industries try to apply Data Mining for the purpose of benefiting from it. One field that has expanded its use is education. There has been an increasing use of instrumental educational software by institutions that has created large repositories of data reflecting how students learn (Romero & Ventura, 2010). The data can then be used with Data Mining to create value. This particular application of Data Mining is known as Educational Data Mining.

Educational data mining is an emerging interdisciplinary research area field that applies data mining algorithms to solve educational issues using the large repositories of data generated by instrumental educational software (Hung, Hsu & Rice) (Romero C., Espejo, Zafra, Romero J. & Ventura, 2013). Its objective "is improving the learning process and guiding students in their learning" (Romero, Ventura & García, 2007).

Similar to Data Mining, the Educational Data Mining process analyses and converts raw largescale educational data, with a focal point on automated methods, into useful information that potentially could have a great impact on educational research and practice (Venkat, 2018) (Baker & Inventado, 2014).

Cristóbal Romero & Ventura (2010) presented a review in the state of the art of Educational Data Mining where they reviewed 306 references published until 2009.

The authors were able to group the survey references according to their categories/tasks that themselves established:

- Analysis & Visualization: 35 papers published.
- Providing Feedback: 40 papers published.
- Recommendation: 37 papers published.
- Predicting Performance: 37 papers published.
- Student Modeling: 28 papers published.
- Detecting Behavior: 23 papers published.
- Grouping Students: 26 papers published.
- Social Network Analysis: 15 papers published.
- Developing Concept Map: 10 papers published.
- Planning & Scheduling: 11 papers published.
- Constructing Courseware: 9 papers published.

From the analysis above we can see that the first 4 Educational Data Mining categories/tasks are the most popular ones with the number of papers published ranging from 35 to 40.

Moreover, they found that Educational Data Mining can be helpful in different ways to a lot of agents. The authors identified different groups of users that look at educational knowledge from different perspectives according to their own mission, vision, and objectives:

- Learners/Students/Pupils.
- Educators/Teachers/Instructors/Tutors.
- Course Developers/Educational Researchers.
- Organizations/Learning Providers/Universities/Private Training Companies.
- Administrators/School /Network Administrators/System Administrators.

Plus, the authors stated that "the number of publications about Educational Data Mining has grown exponentially in the last few years" (Cristóbal Romero & Ventura, 2010). Peña-Ayala (2014) also stated that Educational Data Mining is living its teenage period. These statements made me believe that Educational Data Mining is becoming throughout the years a more interesting research area. Three probable reasons behind this are: a) The use of Instrumental educational software has become more popular due to the advance of technology. As mentioned before, with the use of this type of software comes the data that is generated by them that represents the students learning path and there is an opportunity for the data to be used by researchers. b) A room for improving education not only on the educator's side by upgrading the course mapping, understanding student learning and behavior, predicting student performance, to find students most frequently made mistakes, etc. but also, on the student's side by personalizing e-learning, to get suggested hints, to recommend relevant discussions, etc. (Cristóbal Romero et al., 2007). c) More tools are being developed for this purpose. The tools are not only more powerful but also more interpretable which is an important feature to teachers that do not possess the technical skills to perform Data Mining.

An updated version of their 2010 state of the art paper has been published. They state that "today, one of the biggest challenges that educational institutions face is the exponential growth of educational data and the transformation of this data into new insights that can benefit students, teachers, and administrators" (Cristóbal Romero & Ventura, 2020). Furthermore, the main differences in Educational Data Mining, when compared to a decade ago are:

- New related terms used in the biography:
    a. Academic Analytics and Institutional Analytics - Focused on the political/economic challenge.
    b. Teaching Analytics - It is focused on the educational challenge from the instructors' point of view.
    c. Data-Driven Education and Data-Driven Decision-Making in Education - Refers to systematically collect and analyze various types of educational data, to guide a range of decisions to help improve the success of students and schools.
    d. Big Data in Education - Refers to apply Big Data techniques to data from educational environments.

e. Educational Data Science - Is defined as the use of data gathered from educational environments/settings for solving educational problems

- The number of published books and papers has grown exponentially.
- The interest on data related to new types of educational environments has increased.
- More tools and free datasets are available.
- The number of application problems or topics of interest is wider.

These differences show the wider interest in researching the area and make me believe that Educational Data Mining is close to achieving its mature period. Despite this, there is still some difficulty in bringing these techniques into a real-world scenario (Cristóbal Romero & Ventura, 2020).

2.2. LEARNING ANALYTICS

Learning analytics is an emerging, new and rapidly developing research area related to business intelligence, web analytics, academic analytics and predictive analytics that aims at collecting, measuring, analyzing and reporting student data in their interaction with computer based educational systems which allows teachers, course designers and administrators of such systems to find unobserved patterns in student behavior and display relevant information in interpretable formats (Papamitsiou & Economides, 2014) (Agudo-Peregrina, Iglesias-Pradas, Conde-González & Hernández-García, 2014) (Villagrá-Arnedo, Gallego-Durán, Compañ-Rosique, LLorens-Largo & Molina-Carmona, 2016) (Saqr, Fors & Tedre, 2017) (Popescu & Leon, 2018).

The purpose of Learning Analytics is to predict and explain student success (measured by performance, knowledge, score, or grade), optimize the educational platform and implement timely personalized interventions. Consequently, Predictive modeling for teaching and learning takes big part in Learning Analytics (Hwang, Chu & Yin, 2017) (Popescu & Leon, 2018) (Lu, Anna Huang, Jeff Huang, Lin, Ogata & Yang, 2018). The study of the underlying relations between interactions and students' academic success is another common research topic in the field (Agudo-Peregrina, Iglesias-Pradas, Conde-González & Hernández-García, 2014). Agudo-Peregrina, Iglesias-Pradas, Conde-González & Hernández-García (2014) inferred that interactions are the basic contextualized data units needed for learning analytics.

Hwang, Chu & Ying (2017) published a special issue where 9 quality papers were analysed and summarized the objectives, methodologies, and potential research issues of the Learning Analytics field. The most common methodologies mentioned are the Decision Tree, Clustering, Association Rules, Time Sequence Analysis and Visualization Techniques. Almost all these are used in the Educational Data Mining field as well. By looking at the two fields definition and objectives we can identify more similarities. Based on this, it is important to clarify their differences and similarities so, in the next section I will present a comparison between the two fields.

2.3. EDUCATIONAL DATA MINING VS LEARNING ANALYTICS

As stated before, both Educational Data Mining and Learning Analytics are "relatively new and promising fields of research that aim to improve educational experiences by helping stakeholders to make better decisions using data (Liñán & Pérez, 2015). There is a considerable thematic overlap between the two fields and share similarities such as goals, methodologies, and techniques (Baker & Inventado, 2014) (Liñán & Pérez, 2015). These fields are complementary of each other and one should follow their traces parallelly in order to benefit the most. (Papamitsiou & Economides, 2014).

Despite the similarities they share some differences that also need to be appointed.

Baker & Inventado (2014) stated that while the aim of Learning Analytics is human interpretation of data and visualization, Educational Data Mining focuses more on automated methods. But in their view, "the differences between the two communities are most based on focus and research questions."

Agudo-Peregrina, Iglesias-Pradas, Conde-González & Hernández-García (2014) stated that the differences between the two communities are based in terms of "purpose and scope. While Educational Data Mining is related to the development of methods for analysis of learning data from a technical point of view, Learning Analytics has to do with the interpretation and contextualization of that data for the improvement of learning".

Liñán & Pérez (2015) presented a paper where the similarities, differences and time evolution of both fields was discussed. They identified five key distinctions between Educational Data Mining and Learning Analytics which are:

- Discovery: Educational Data Mining researchers are interested in automated discovery while the aim of Learning Analytics is leveraging human judgement.
- Origins: Educational Data Mining is rooted in computer based educational systems and student modelling. Learning Analytics origins are related to the semantic web, intelligent curriculum, outcome prediction and systematic interventions.
- Adaption and Personalization: Educational Data Mining performs automated adaptation. Learning Analytics informs and empowers instructors and students.
- Techniques and Methods: Educational Data Mining employs more methods of classification, clustering, Bayesian modelling, relationship mining, discovery with models and visualization. On the other side, Learning Analytics focuses on social network analysis, sentiment analysis, influence analysis, discourse analysis, learner success prediction, concept analysis and sense- making models.

In my opinion the biggest gain in value happens when both these fields are used together. If this is the case, there is the opportunity to automate some of the learning processes while learning about them. Also, it is from my understanding that a lot of the differences between fields is due to the interest and values of researchers.

2.4. COMPUTER BASED EDUCATIONAL SYSTEMS

In today's reality there are different computer based educational systems each one with their own characteristics and purpose. Moreover, each one generates different types of data that account for different problems and tasks to be resolved (Cristóbal Romero & Ventura, 2010).

Some of the most popular Computer Based Educational Systems are:

- Web-Based Education/E-Learning - Systems used for distance learning where course material, assessment and support are all delivered online and face to face contact does not happen between students and teachers (McKimm, Jollie & Cantillon, 2003).
- Learning Management Systems – Systems aimed at supporting the lecturers teaching plan (Macfadyen & Dawson, 2010) (Peña-Ayala, 2014).
- Intelligent Tutoring Systems - Systems that use Artificial Intelligence to provide agents that know what they teach, who they teach and how to teach. They aim at providing an immediate and customized learning experience or feedback to learners, without the intervention of teachers (Alkhatlan & Kalita, 2018).
- Adaptive Educational Systems - Systems that monitor important learner characteristics and make appropriate adjustments to the learning plan to support and enhance the student experience (Shute & Rivera, 2012).

Peña-Ayala (2014) presented a survey that consisted in the analysis of 240 Educational Data Mining works from 2010 and the first quarter of 2013. During this analysis they identified "37 kinds of Computer Based Educational Systems, where Intelligent Tutoring Systems and Learning Management Systems were the most prominent with 49% of the sample". This shows that the majority of course providers pick one of these two has their institutions Computer Based Educational System.

Liñán & Pérez (2015) state that the use of these systems brings several positive aspects:

- Interactive communication between students and students and instructors is favored.
- Promotes continuous self-evaluation and collaborative activities.
- Contributes to the development of technical skills.
- Helps reducing the gap between theory and practice.

The positive aspects that the use of Computer Based Educational Systems bring to education indicate that their use is mandatory for a better learning experience not only on the student's side but also on the educator's side by facilitating several aspects that otherwise would be very difficult.

From Computer Based Educational Systems two types of data can be gathered. State based data is one of them. This type of data is concerned with demographics, psychological traits, past performance, etc. The other one is event driven data that is based on student's activity with the system and leaves a digital fingerprint (Popescu & Leon, 2018) (Liñán & Pérez, 2015).

2.4.1. Learning Management Systems

Learning Management Systems are a specific kind of Computer Based Educational Systems that allow educator's to "distribute information to students, produce content material, prepare assignments and tests, engage in discussions, manage distance classes and enable collaborative learning with forums, chats, file storage areas, news services, etc." (Cristóbal Romero et al., 2013).

Learning Management Systems are usually implemented in what is described as mixed-model and web-supported models. These models incorporate the use of Learning Management Systems with the objective of supporting the routine of teachers in traditional classroom-based courses (Macfadyen & Dawson, 2010) (Peña-Ayala, 2014).

Modern Learning Managements Systems "provide a large amount of data from the student activity" (Villagrá-Arnedo, Gallego-Durán, Compañ-Rosique, LLorens-Largo & Molina-Carmona, 2016). They record the student's navigation on the site in terms of records, logs, interactions, and other digital footprints as well as dates and timestamps. They keep detailed information of every student´s interaction in the system such as the number and duration of online sessions, tools accessed, content pages visited, opening course materials, taking tests and communicating with peers (McCuaig & Baldwin, 2012) (Macfadyen & Dawson, 2010) (Cristóbal Romero et al., 2013) (Saqr et al., 2017).

Currently, one of the most popular and used Learning Management System is Modular Object-Oriented Developmental Learning Environment, or Moodle. It is a free system "enabling the creation of powerful, flexible and engaging online courses and experiences" (Cristóbal Romero et al., 2013). It is designed using pedagogical principles to help educators (Calvo-Flores, Galindo, Jiménez & Piñeiro, 2006).

2.5. PERFORMANCE PREDICTION RELATED WORK

I have reviewed several papers where the main objective is Performance Prediction. This study has proved to be very beneficial in terms of knowing what procedures have been used and the ones that got better results in the field. Decisions such as what variables to use, what algorithms perform better and pre-processing tasks that prove the best results are some of the procedures covered by these papers.

Minaei-Bidgoli, Kashy, Kortemeyer & Punch (2003) studied the prediction of student performance in an educational web-based system developed at Michigan State University called LON-CAPA. They used logged homework data of an introductory physics course held in 2002 with 227 participants. This course included 12 online homework sets and 7 features were extracted such as success rate, success at the first try, number of attempts before correct answer is delivered, etc. The authors labeled the dependent variable, grade, in 3 possible ways: a) 9 class labels corresponding to the possible 9 grade values the students could obtain; b) 3 class labels representing high, middle, and low grades; c) 2 class labels representing passed and failed grades. Plus, they compared the performance of different classifiers (Quadratic Bayesian, 1-nearest neighbor, k-nearest neighbor, Parzen-window, multi-layer perceptron and several different kinds of decision trees). Regarding the case of 2-class labels, k-nearest neighbor performed better with 82.3% accuracy. The case of 3-class and 9-class labels CART decision tree had the best accuracy of about 60% and 43% respectively.

Calvo-Flores, Galindo, Jiménez & Piñeiro (2006) applied a radial basis function artificial neural network Model for prediction of performance using Moodle log data from 240 Computer Science students at the university of Cordoba. 11 features were obtained such as the number of examination sessions, total of accesses to “resource view”, percentage of “resource view” accesses from the total accesses, number of different “resource view” of each type have been visited, etc. The dependent variable was labelled to pass or fail. The authors studied if it was possible to predict performance based only on the interactions between the students and the Moodle platform. The results showed that this is possible by obtaining 80.2% accuracy.

Ibrahim & Rusli (2007) performed a model comparison between artificial neural network, decision tree and linear regression for performance prediction in the end of the semester using data from 206 students. The features used were the information technology application knowledge, previous school, programming knowledge and family financial status. The artificial neural network had the best performance, followed by decision tree and linear regression.

Macfadyen & Dawson (2010) developed an early warning system for educators that integrated performance prediction. The authors used 118 students learning management system tracking data from three terms of a full online undergraduate biology course at the university of British Columbia during 2008. The authors used features such as number of messages read, total number of discussion messages posted, number of uses of the “compile” tool, number of files viewed, number of assessments started, number of assessments submitted, etc. First, they studied simple correlations of learning management system tracking variables with final grade. A positive and statistically significant correlation happened between 13 of the 22 total variables and student final grade. The results show that engaged and discursive students are more likely to complete the course successfully than their fewer interactive peers. Secondly, a linear multiple regression analysis was performed to identify the variables with more predictive power. The model is a linear combination of only 3 online activity variables: total number of discussion messages posted, total number of mail messages sent, and total number of assessments completed. Finally, a logistic regression model was built with the same variables as before. Plus, the authors labelled the independent variable as “at risk” if the final grade value was below 60% or “performing adequately or better” if final grade value was higher or equal than 60%. The model presented an accuracy of 73.7% and effectively identified most of the students who failed or almost failed the course.

McCuaig & Baldwin (2012) presented a study where it was concluded that successful learners exhibit different interaction behaviours with the learning management system than less successful learners. Students who are more engaged and show more activity in the learning management system have better results. Moreover, the results indicated that formal assessment results are unnecessary to accurately predict which learners will be successful. The study was conducted on 122 computer science students during the winter of 2011 in a single semester programming class and used several features concerning homework assessments, days active in the learning management systems, number of views of course page, etc.

Hung, Hsu & Rice (2012) conducted a study with 7538 students (23,854,527 learning logs in 883 courses) where an approach of program evaluation through analyses of student learning logs, demographic data and end-of-course evaluation surveys in an online K-12 supplemental program was performed. After the data coming from different sources was merged, the authors started by doing a

cluster analysis. From there on, an exploratory data analysis was performed where several unknown patterns were discovered. The authors studied the average clicks per course in different subject areas, the student's subject preferences, the pass rate in different subject areas and the student performance and engagement by course. Following that, predictive analysis was implemented using three CRT decision tree algorithms were the following three variables were adopted as dependent variables: a) Average course grade; b) Average course satisfaction; c) Average instructor satisfaction. In final grade prediction the results indicate a positive (higher engaged –> higher performance) and negative (lower engaged –> lower performance) correlation between engagement level and performance.

Cristóbal Romero, Espejo, Zafra, José Romero & Ventura (2013) compared 21 different data mining techniques to predict the final exam grade using 438 students tracking data in 7 engineering Cordoba university Moodle courses. Plus, they developed a specific Moodle mining tool for this purpose. They used several variables such as the number of assignments done, the number of quizzes passed, the number of quizzes failed, the number of messages sent to the forum, total time used on assignments, total tune used on forums, etc. The first experiment was done to evaluate if original data performed better than data filtered by rows (only rows without any missing values are used) or data filtered by columns (only 4 input variables are used, these were selected prior using feature engineering methods). The results show that 14 out of 21 algorithms obtained their better results using original data (CART, GAO, GGP and NNEP were the ones with better accuracy) and 7 out of 21 obtained better results with data filtered by columns (CART, SAP and GAP were the ones with better accuracy). Considering all, the best results were proven to be obtained with original data. The second experiment used the original data but now applying discretization or rebalancing. 12 out of 21 algorithms obtained better performance using original data (CART, GAP, GGP and NNEP) and 13 out of 21 obtained better performance using categorical data (CART and C4.5). Both experiments showed that with different algorithms different procedures obtain the best results and show the importance of experimentation. The authors also reinforce the idea that a good classifier model in education "has to be both accurate and comprehensible for instructors".

Agudo-Peregrina, Iglesias-Pradas, Conde-González & Hernández-García (2014) studied three different interaction classifications and evaluated their relationship with academic performance. The classifications are based on agent (student-to-student, student-to-teacher, or student-content interactions), frequency of use (most used, moderately used or rarely used) and participation mode (active or passive). The study was conducted on 138 students from six online courses and 218 students from two Moodle-supported F2F courses. First and to assess predictive power, bivariate correlations and multiple linear regression were performed between the different interactions and student performance measured by grade. The bivariate correlation analysis showed that there is a significant relation between different types of interactions and the students' academic performance. Regarding the multiple linear regression analysis and focusing on Moodle-supported F2F courses, the authors were not able to establish any predictive model contrarily to online courses were the three different classifications showed to be useful predictors of student achievement. According to the authors' results, academic performance is explained by: a) the interactions they have in the VLE with their peers and mainly with the teachers; 2) the interactions related to student assessment; 3) interactions involving active participation.

Saqr, Fors & Tedre (2017) studied the use of learning analytics to predict under-achieving students in a blended medical education course. The study included 133 students enrolled in a blended medical

course where they used Moodle at their will. Activity data was then extracted (logins, views, forums time, formative assessment, and communications at different points of time) and 64 features were built. Plus, five engagements indicators (login, course views, forum post, time, and formative assessment) were calculated to reflect self-regulation and engagement. For predicting students' performance, the authors tested automatic linear modelling that obtained a 63.5% accuracy were the most important features were login sub-indicator, views of course information, formative sub-indicator, posting sub-indicator, total views at mid-course and total post at mid-course. Binary logistic regression was also tested and obtained an accuracy of 80.8%.

Villagrá-Arnedo, Gallego-Durán, Compañ-Rosique, Llorens-Largo & Molina-Carmona (2016) developed a learning platform used by 336 students to collect behavioural (Number of frontpage visits, Number of exercises downloads and Number of submissions per stage) and learning data (several exercise related data). They studied the effects of using these types of data for prediction of performance. The predictive model was based on a standard C-parameterized margin, SVM classifier. The best results were obtained when both behavioural and learning data were used, as expected. Although the worst results were obtained when using only behavioural data an accuracy of around 60% was obtained.

Lu, Anna Huang, Jeff Huang, Lin, Ogata & Yang (2018) applied learning analytics for the early prediction of students' final academic performance in a blended calculus course attended by 59 students. 21 variables were collected such as the Number of days the student exhibits activity per week, Number of days a student watches videos per week, Number of times a student clicks "Play" per week, Number of times a student engages in online practice per week, Students' weekly homework score, etc. Accumulated and duration datasets were built to represent accumulated data from the start of the study to that week or represent data from specific weeks individually, respectively. For prediction performance the authors used PCR algorithm. The R-squared and adjusted R-squared scales are higher in all accumulated data sets comparing to the duration datasets. Plus, the accumulated datasets had regression models with better goodness of fit and prediction ability than those of the duration datasets. Moreover, the results suggest that students' final academic performance could be predicted when only six weeks have passed since the enrolment of the semester and seven critical factors affected students' final academic performance, namely: Number of activities a student engages in per week, Number of times a student clicks "Play" during video viewing per week, Number of times a student clicks "Backward seek" during video viewing per week, Student's weekly practice score, Student's weekly homework score, Student's weekly quiz score and Number of times a student participates in after-school tutoring per week.

Popescu & Leon (2019) studied the effect of students' social media traces on performance. Data was obtained from 343 web applications design course student's where wiki, blog and microblogging tools were used for communication and collaboration activities in a project-based learning scenario. 14 numeric features were computed for each student such as the number of blog posts, the average length of a blog comment, the number of files uploaded on the wiki, the number of wiki page revisions, etc. For the prediction task the authors compared the performance of large-margin nearest neighbour regression, random forest, and k-nearest neighbours. The results showed that students' actions on social media tools are good predictors of performance. Large-margin nearest neighbour regression outperformed both random forest and k-nearest neighbours, obtaining good correlation coefficients and 85% of predictions were within 1 point of the actual grade.

Several conclusions were taken while making the literature review:

- Most of the papers reviewed only take into consideration a small sample of students, usually only 1 course. From a general perspective there is a lot of value in considering different courses to train a model, especially considering that different courses may indicate a different interaction between the system and the student.
- Several papers use variables that are not generally replicable, e.g., specific homework programming data. While these types of variables may represent a lot of information to the model developed by the researchers, they are not accessible in all computer based educational systems thus making it hard for the construction of a general performance prediction model.
- The predicted variable is usually labelled “pass” or “fail”. This translates to a better model score, but it is impossible to distinguish between students that pass and have minimum grade performance and the ones who have excellent performance.
- Moodle frequency variables like the number of times a student accessed a specific resource in the system have shown to be good predictors of performance.
- Successful students show different interaction patterns with the system when compared to less successful learners. Additionally, students who are more engaged show better results.
- Formal assessment results are unnecessary to predict performance.
- A lot of different techniques prove to be efficient. The ones used the most are K-Neighbours Classifier, Artificial Neural Networks and Linear Regression.

3. METHODOLOGY

This study was conducted using the Python coding language supported by Jupyter Notebook technology. Python is the world's third most popular programming language and is described as an interpreted, high-level and general-purpose programming language. In order to accomplish the desired result, make coding, data analysis and modeling more efficient, several Data Science packages for Python were used in this study:

- **Pandas** - Pandas is a software library written for the Python programming language for data manipulation and analysis. In particular, it offers data structures and operations for manipulating numerical tables and time series. One of those structures is called Data frame, which is a two-dimensional data structure, i.e., data is aligned in a tabular fashion in rows and columns like a spreadsheet or SQL table (Pandas, 2021).
- **Scikit-Learn** - Scikit-Learn is a software machine learning library for the Python programming language. It features various classification, regression and clustering algorithms including support vector machines, random forest, gradient boosting, k-means and DBSCAN (Scikit-Learn, 2020).
- **Seaborn** - Seaborn is a Python data visualization library. It provides a high-level interface for drawing attractive and informative statistical graphics (Seaborn, 2021).

Jupyter Notebook is an open-source web application that allows users to create and share documents that contain live code, equations, visualizations, and narrative text (Project Jupyter, 2021).

3.1. SEMMA OVERVIEW

This study follows the well-known Data Mining and Machine Learning methodology SEMMA (which stands for Sample, Explore, Modify, Model, and Assess). This methodology was introduced by SAS Institute with the objective of creating a standard process for the development of data mining projects regardless of industry. Figure 1 below gives a visual representation on the workflow of the methodology and the tasks, generally, involved in each phase.

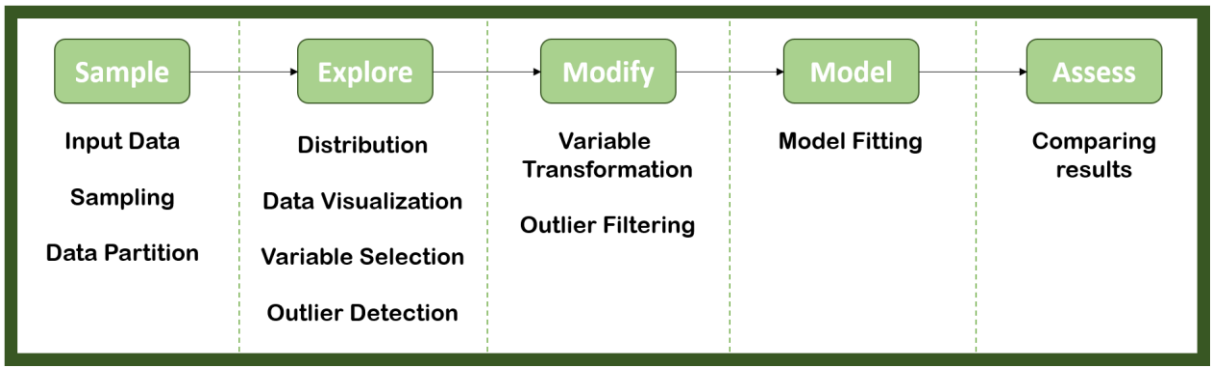

Figure 1 - SEMMA Phases and Tasks.

In the following sections (3.1.1 to 3.1.5) a more in-depth explanation of each phase has been developed.

It is important to state that all the data used in this study was only made available to me after being anonymized. This study uses extrapolation to reach general results and personal data is not important to the task at hand. All RGPD rules have been followed.

3.1.1. Sample

The sample phase, as the name indicates, is concerned with sampling raw input data into a usable dataset fit for the purpose of the project. In this study, raw data was extracted from 2 source systems, pre-processed and aggregated into one final working dataset. Figure 2 gives a general overview of the process.

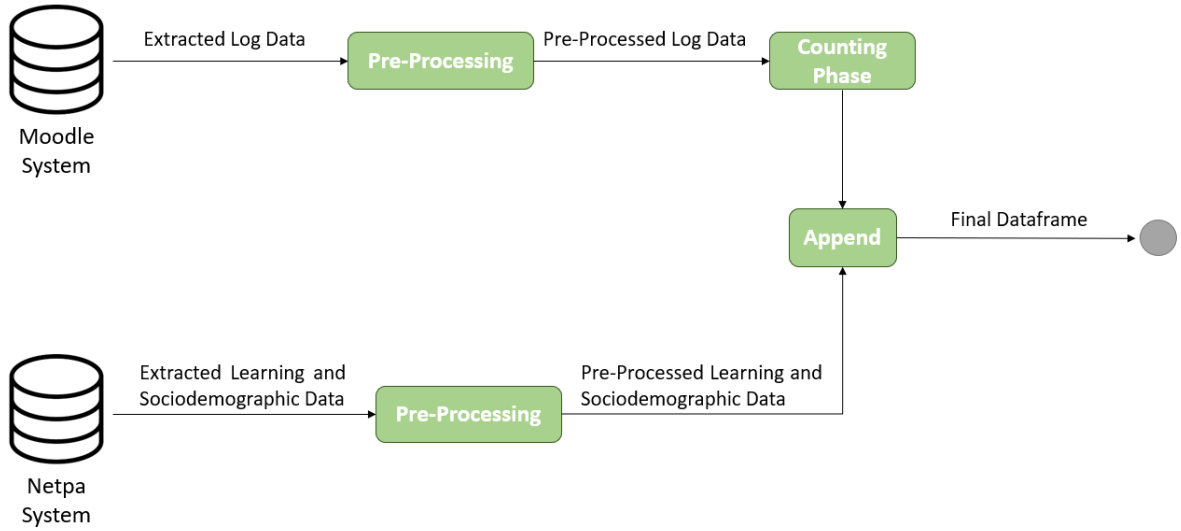

Figure 2 - Sample Phase Workflow.

3.1.1.1. Log Data

Log Data used in this study was obtained from Nova IMS Moodle from September 2019 throughout January 2020, corresponding to a full semester. Each month raw log data was extracted from the system in a .csv file format and imported into 6 different Pandas Dataframes (one for each month) in a Jupyter Notebook. Although some of these files had error columns and log data is classified as unstructured, Table 1 describes the general structure and description of each field of the Moodle System logs exported.

The error columns mentioned above appeared in some of the files because error log messages do not follow the general log structure. This way, when an error message contained a comma a new column named “Unnamed:” was added to the Dataframe. These rows and columns where deleted.

Field Name	Description	Value Example
Id	Log identifier.	13444875
Timecreated	Log creation Date and Time.	2019-09-01 00:03:27
Eventname	Description of the user action.	\core\event\course_information_viewed
Component	Moodle component. Part of the Eventname.	core
Action	Moodle action. Part of the Eventname.	Viewed
Target	Moodle target. Part of the Eventname.	Course_information
Objecttable	Object table name in Moodle System.	course
Objectid	Object identifier in Moodle System.	775
Crud	CRUD operation performed.	r
Edulevel	---	2
Contextid	---	
Contextlevel	---	50
Contextinstanceid	---	775
Userid	User identifier.	5490
Courseid	Course subject identifier.	775
Relateduserid	If the action affects another user this field identifies the related user.	817
Anonymous	If the action is anonymous.	0
Other	---	N
Origin	Origin of the log.	web
Realuserid	---	NaN
ip	User´s ip address.	xxx.xxx.xxx.xx

Table 1 - Log Structure.

Looking at the data, several fields were not important to the task at hand and to the purpose of the study. For this reason, the columns “timecreated”, “objecttable”, “objectid”, “crud”, “edulevel”, “contextid”, “contextlevel”, “contextinstanceid”, “relateduserid”, “anonymous”, “other”, “origin”, “ip” and “realuserid” were deleted.

To the relevant fields, “id”, “eventname”, “component”, “action”, “target”, “userid” and “courseid” the rows with missing values were discarded.

3.1.1.2. Counting Phase

The objective of the counting phase is to count how many occurrences of certain types of logs occurred for each student in each subject and afterwards map them to a variable in the final Dataframe.

Figure 3 below gives a visual representation of the counting phase.

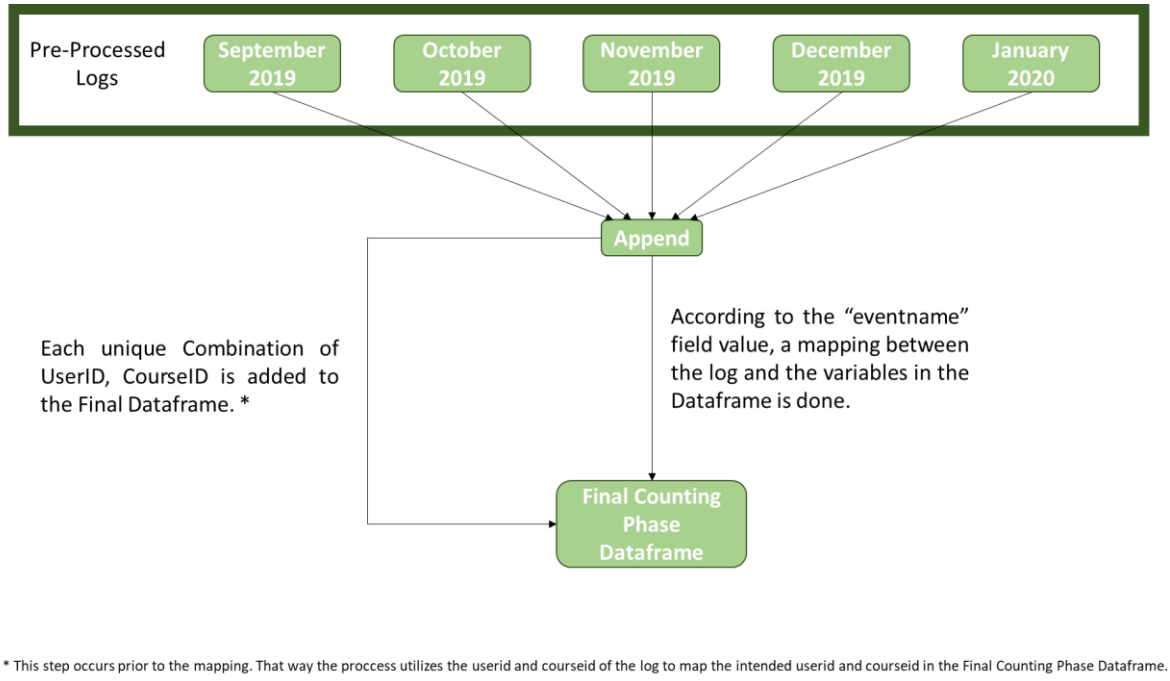

Figure 3 - Counting Phase WorkFlow

Has seen in the figure above, the pre-processed log Dataframes are grouped together. From here the unique combination of UserId, CourseId values are obtained and added to the Final Counting Phase Dataframe under the same variables.

After that, the aggregated pre-processed log Dataframe is counted and the fields in the Final Counting Phase Dataframe are updated. The way this counting occurs is through the “eventname” field in the log Dataframe. For every row in the log Dataframe the “UserId”, “CourseID” combination is matched to the same variables in the Final Counting Phase Dataframe. After that, the value in the “eventname” field is checked and if it contains an important value a variable in the Final Counting Phase Dataframe is updated with a plus one count.

In the bullet points below is identified which “eventname” values are directionally related to variables in the Final Counting Phase Dataframe.

- # of forum messages read
  - \\mod_forum\\event\\discussion_viewed

- # of forum messages posted
  - \\mod_forum\\event\\post_created
  - \\mod_forum\\event\\subscription_created
  - \\mod_forum\\event\\discussion_created
  - \\mod_forum\\event\\discussion_subscription_created

- # of pages accessed
  - \\core\\event\\course_viewed
  - \\mod_folder\\event\\course_module_viewed
  - \\mod_forum\\event\\course_module_viewed
  - \\mod_turnitintolltwo\\event\\list_submissions
  - \\mod_assign\\event\\submission_status_viewed
  - \\mod_page\\event\\course_module_viewed

- # of files accessed
  - \\mod_resouce\\event\\course_module_viewed

- # of submission
  - \\mod_turnitintooltwo\\event\\add_submission
  - \\mod_assign\\event\\assessable_submitted
  - \\assignsubmission_file\\event\\assessable_uploaded

- # of total clicks
  - All the events

These values, in the eventname field, correspond to specific pages accessed by the student in the course page.

3.1.1.3. Netpa System Data

Sociodemographic and Learning Data used in this study was obtained from Nova IMS Netpa system. The data originates from the students that attended the Portuguese college in the academic year of 2019/2020.

- **Sociodemographic Data**

The data was extracted from the system in a .csv file and imported into a Pandas Dataframe in a Jupyter notebook. Table 2 gives the field name, description and value example for each variable imported.

Field Name	Description	Value Example
ID_INDIVIDUO	User identifier.	17250
DT_NASCIMENTO	Birthday date.	22/06/1978 00:00
SEXO	Gender.	M
ESTADO_CIVIL	Marital status Code.	2
DS_EST_CIVIL	Marital status Description.	Casado(a)
CD_NACIONA	Nationality Code.	1
DS_NACIONA	Nationality Description.	Portugal
CD_NATURAL	Naturality Code.	900086
DS_NATURAL	Naturality Description.	Brasil
DS_PAIS	Country of address.	Portugal
CD_POSTAL	Postal Code of address.	2775
DS_POSTAL	Postal Code Description.	Parede
CD_SUBPOS	Sub Postal Code.	42
DS_UPPOST	Postal Code of address with UPPER case.	Parede

Table 2 - Sociodemographic Variables.

Looking at the data, the variables “ESTADO_CIVIL”, “CD_NACIONA”, “CD_NATURAL” were dropped because they were not important to the task at hand.

The sociodemographic data was joined to the log data through the Moodle variable userid and Netpa variable ID_INDIVIDUO.

- **Learning Data**

The Learning data was extracted from the system in a .csv file and imported into a Pandas Dataframe in a Jupyter notebook. Table 3 gives the field name, description and value example for each variable imported.

Field Name	Description	Value Example
ID_INDIVIDUO	User Identifier.	16609
CD_LECTIVO	Academic year.	201920
CD_CURSO	Course ID.	7512
CD_DISCIP	Course Subject Identifier.	200134
CD_DURACAO	Semester/Quarter Identifier.	S1
CD_GRU_AVA	---	1
CD_AVALIA	Assessment Code.	99
DS_AVALIA	Assessment Description.	Normal
DT_AVALIA	Assessment Date.	29/03/2020 00:00
NR_AVALIA	Assessment Grade.	20
CD_STA_EPO	Assessment State Code.	3
DS_STA_EPO	Assessment State Description.	Aprovado
CD_FINAL	Parcial/Final Grade Identifier.	15

Table 3 - Learning Data.

Being the log data only from the first semester of the 2019/2020 academic year, data was filtered so only rows with "CD_DURACAO" equal to S1, T1 OR T2 would be considered. These represent the first semester, first trimester and second trimester. Plus, only final grades are of interest to this study. For that reason, the rows were filtered so that "CD_FINAL" equals to "Nota final". Moreover, the columns "CD_CURSO", "CD_GRU_AVA", "CD_AVALIA" and "CD_STA_EPO" were deleted.

The learning data was joined to the log and sociodemographic data through the Moodle variables userid and courseid and Netpa variables ID_INDIVIDUO and CD_DISCIP.

At the end of the counting phase, the dataframe was 3226 rows and 28 variables. Some of these variables will be used on the explore phase but not as input to the model. Plus, several modifications were implemented and documented to the same variables on the modify phase.

3.1.2. Explore

In the SEMMA methodology, after the counting phase comes the explore phase. This next section has the purpose of getting to know the data I had collected to this study. Data visualization and descriptive statistics were used to enhance and facilitate the exploration and reach a deep understanding of the data. Moreover, this is an important step in understanding what types of modifications need to be done to the dataset, in order to get a better predictive result.

3.1.2.1. Descriptive statistics

Descriptive statistics were computed and analysed for the numerical variables. Table 4 below shows the results of the analysis. It was possible to access the sum, average, median, mode, standard deviation, variance, minimum and maximum value. Plus, Age was computed from the birthday date.

This analysis will be used in addition with data visualization techniques to better estimate outliers in the sample.

Variable	Unit of measure	Sum	Average	Median	Mode	Std. Dev.	Varience	Minimum	Maximum
# of forum messages read	occurrences	15,413	4.78	0	0	13.49	182.11	0	138
# of forum messages posted	occurrences	234	0.07	0	0	0.51	0.26	0	8
# of pages accessed	occurrences	357,001	110.66	90	84	84.17	7,085.41	0	644
# of files accessed	occurrences	179,930	55.77	44	0	53.53	2,865.37	0	439
# of total clicks	occurrences	572,438	177.45	154	131 and 64	123.92	15,355.46	1	967
# of submissions	occurrences	2,698	0.84	0	0	2.10	4.43	0	22
NR_AVALIA	Grade from 0 to 20	--	14.00	15	16	4.56	20.83	0	20
Age	Years	--	25.98	24	19	6.44	41.47	18	58

Table 4 - Descriptive statistics for numerical variables

Regarding categorical variables, an analysis will be done using data visualization. These types of variables are a lot harder to analyse and modify because of the different levels they may have. Some of the techniques to modify categorical variables and make them usable in machine learning algorithms tend to extend the number of variables in the dataframe. This may cause what is known as the curse of dimensionality. Introduced by Bellman, refers to the "explosive nature of spatial dimensions and its resulting effect, such as, an exponential increase in computational effort, large waste of space and poor visualization capabilities"(Venkat et al., 2018). With a small sample, like the one in this study, the decision to use or modify categorical variables is an important one.

3.1.2.2. Data visualization

Zhao Kaidi (2000) defines visualization as "the graphical presentation of information, with the goal of providing the viewer with a qualitative understanding of the information contents. It is also the process of transforming objects, concepts, and numbers into a form that is visible to the human eye".

To explore this important concept, several different plots were used in order to visually understand the facts hidden in the data and reach a deep understanding of it.

In this study, has said before in this report, there are 3226 students that attended 83 different subjects/courses, throughout the first semester at Nova IMS, during the 2019/2020 academic year.

From these 83 subjects, 63 occurred in the first semester, 9 occurred in the first trimester and 11 occurred in the second trimester, as seen on figure 4.

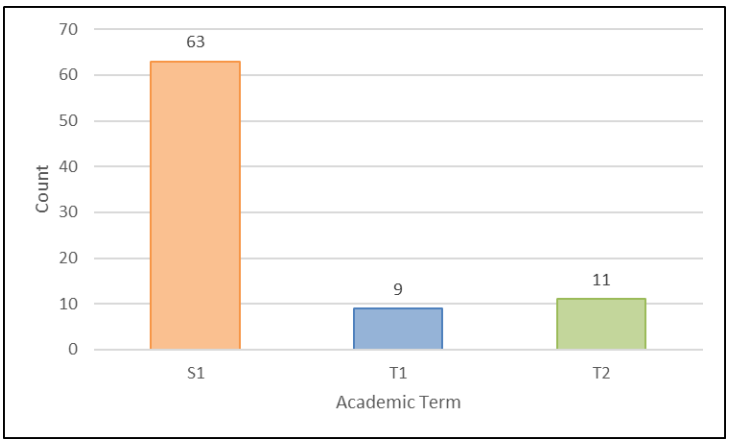

Figure 4 - Count of courses per Academic Term

Regarding the final evaluation results, figure 5 below shows the number of occurrences of approved, failed, quit, or missed final grades for the 3226 students in the 83 subjects. As expected, most students passed the subject by obtaining a final grade of 10 or higher.

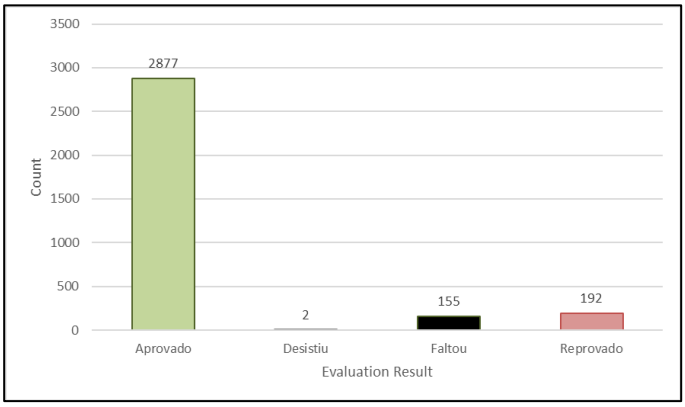

Figure 5 - Evaluation results

The final grades presented in this study were obtained at different types of assessments. Figure 6 below shows that 2747 students got their final grade from normal evaluation. 320 students from "Recurso" (failed to get approved in the first epoch, tried in the second), 146 from improvement (approved in the first epoch, tried to improve the grade in the second), 12 from "Creditação" (Students that were approved in a similar subject from other course) and 1 from the special epoch.

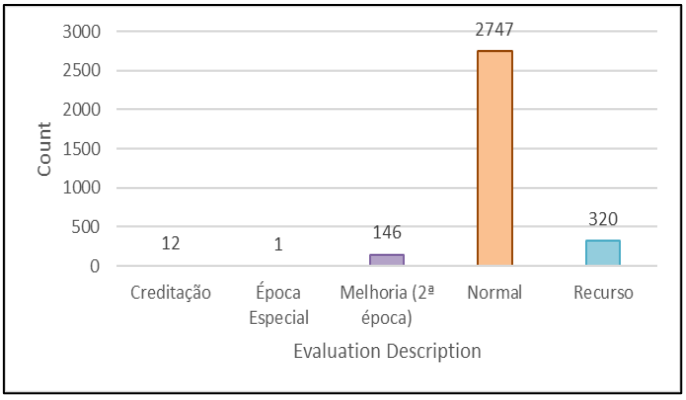

Figure 6 - Evaluation description

Data distribution describes the grouping or the density of the observations in the sample. To visually observe the distribution of the several Moodle activity, age, and grade variables, Kernel density estimation plots (kde) were created.

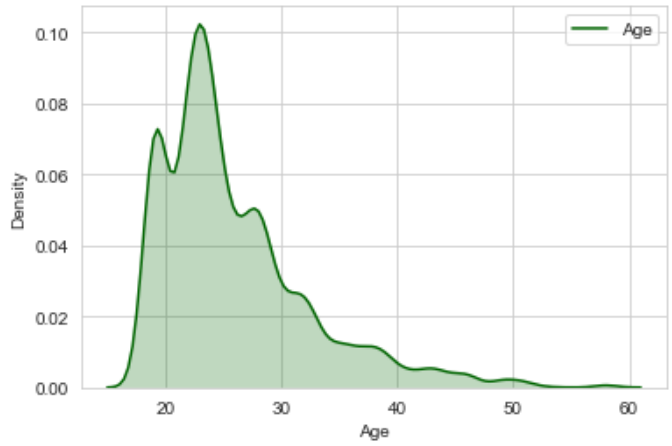

Figure 7 - Kde plot for Age variable

From figure 7, represented above, we can observe that the age variable has more density in the range of 20 to 30 years, with some occurrences in the values that follow. Moreover, and as seen in table 4, the maximum age value in this sample is 58.

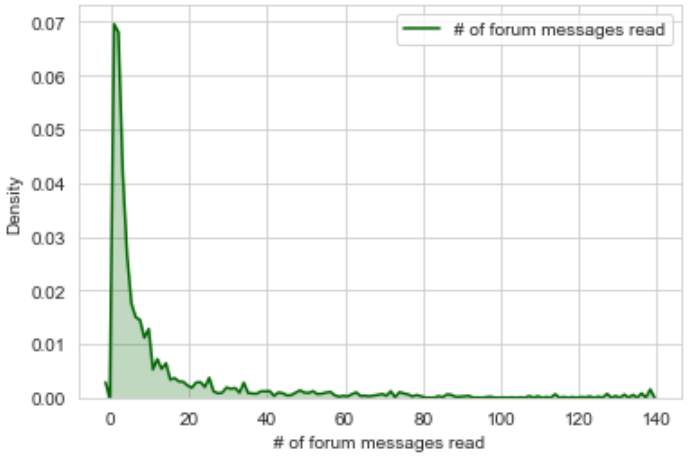

Figure 8 - Kde plot for # of forum messages read variable

Figure 8 shows the density of the # of forum messages read variable. Has seen above, the density in this variable is high around the 0 to 10 range, showing not a lot of students read forum messages. The forum tool, from my personal experience using it, is not used by the course designers extensively and, in some cases, not used at all. This explains why the occurrences have low values. Despite this, the maximum number of forum messages read was 138.

Similarly, figure 9 shows the density of the # of forum messages posted variable. This activity is almost not used at all by the students, showing big density around the value 0. The maximum value of this variable is only 8. Again, from my personal experience, students only use the forum to read messages if necessary and prefer other ways of communication with their peers and course instructors. Moreover, being that a small portion of teachers use the tool it is expected that the values of posted messages by students is almost none.

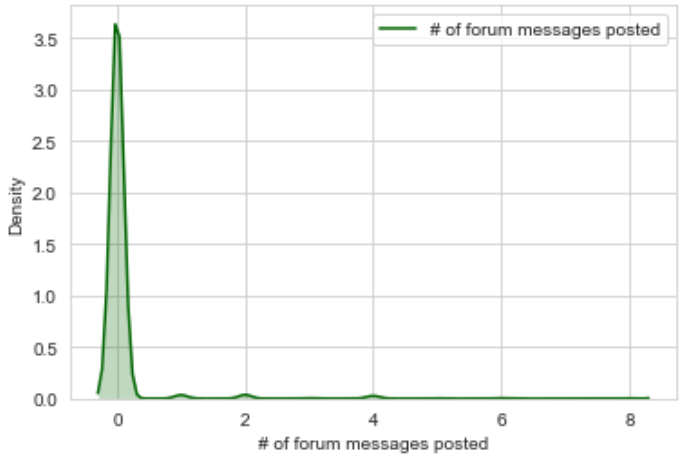

Figure 9 - Kde plot for # of forum messages posted variable

Figure 10 shows the density of the # of pages accessed variable. We observe greater density in the 50 to 150 range and, the plot, is more extended than the last 2 plots, indicating a bigger difference in values for the sample. The maximum value in this variable is 644. Accessing pages is the primary purpose of the Moodle system and the one student's use the most. With this in mind, it is not surprising that this variable shows the second most activity, only behind the # of total clicks.

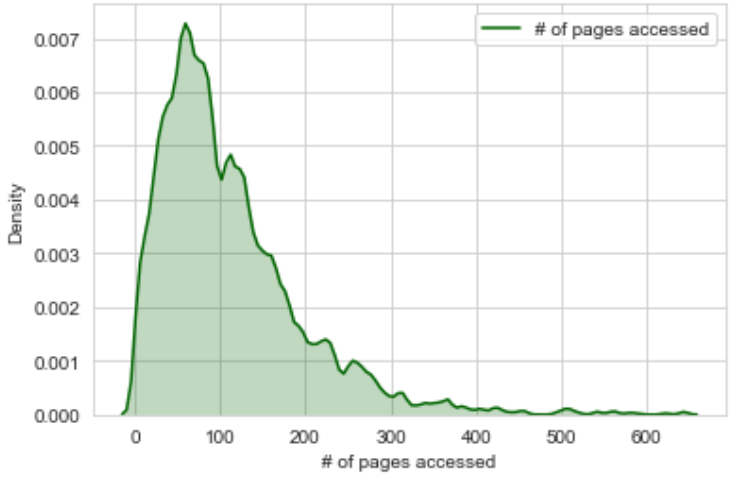

Figure 10 - Kde plot for # of pages accessed variable

Figure 11 show the kde plot for # of files accessed variable. The density in this variable shows a big spike near the 0-value followed by a descending curve from the range of 50 to 439, although there are low values of density from 200 onwards. Again, from my experience, the high spike in density around the 0 value comes from the fact that most of the students accesses a particular file one time and then proceeds to download it, in order to physically store it in their personal machines. File sharing between students may be other reason for this spike at low values.

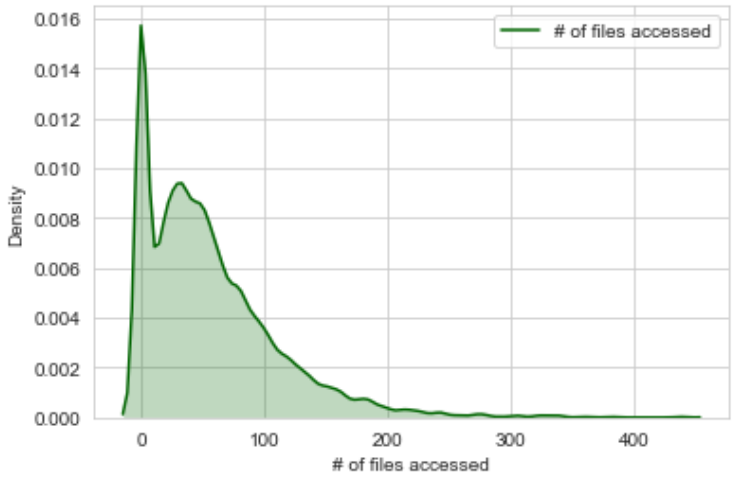

Figure 11 - Kde plot for # of files accessed variable

From figure 12 below, we can observe that the values for the # of total clicks variable range from 1 to 967 with the biggest density around the 100 to 200 range. This the variable with most activity, as expected, because it represents all the clicks done by a student in the Moodle platform.

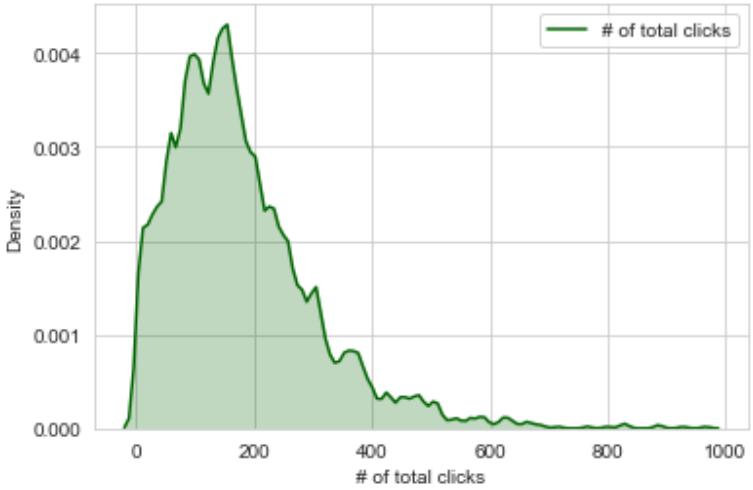

Figure 12 - Kde plot for # of total clicks variable

Figure 13 shows the kde plot for # of submissions variable. Like Figure 9, most of the observations have 0 submissions, represented by the big density spike. From my personal experience as a student, most of the subjects do not require students to do submissions and if they had, different channels, like email, could be used. Additionally, a small number of submissions, in the range of 1 to 3 is normally what is required in subjects. These submissions usually represent projects or homework's students need to perform. In order to get evaluated through continuous evaluation.

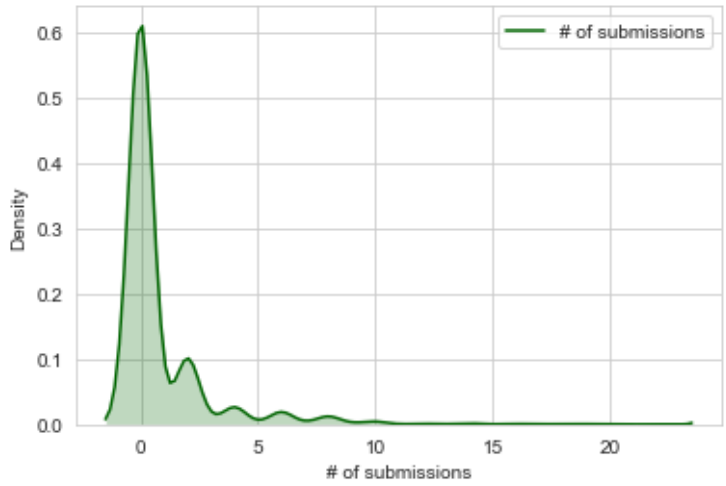

Figure 13 - Kde plot for # of submission variable

Regarding the grade variable, we can observe that the values that have greater density are in the range of 13 to 17. Moreover, we can see that there is some density in the 0 value. This is explained, not only, by the poor performance of some students, but by the students that quitted or missed the assessment. To plot the density of the grade variable, missing values from the types of students mentioned prior were filled with 0. These students will be handled later in the modify phase, since 0 is not a "real" grade and doesn´t reflect their Moodle activity.

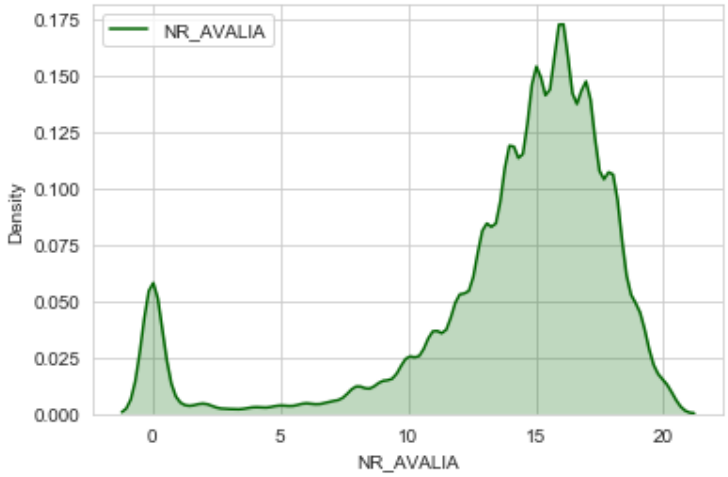

Figure 14 - Kde plot for Grade variable

Gender variable was explored by counting the different occurrences in the dataset. Moreover, Kernel density estimation plots were computed for the Moodle activity variables for each gender value.

From figure 15 below, we can observe that there are more female students than male students, in this study. 1679 students are female, and 1547 students are male.

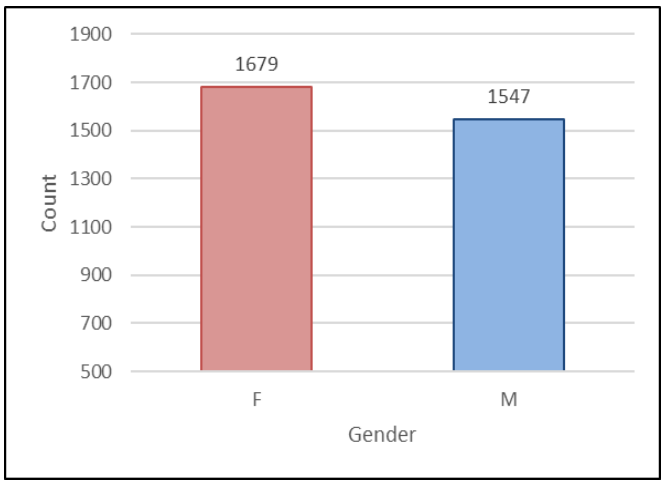

Figure 15 – Number of occurrences per Gender

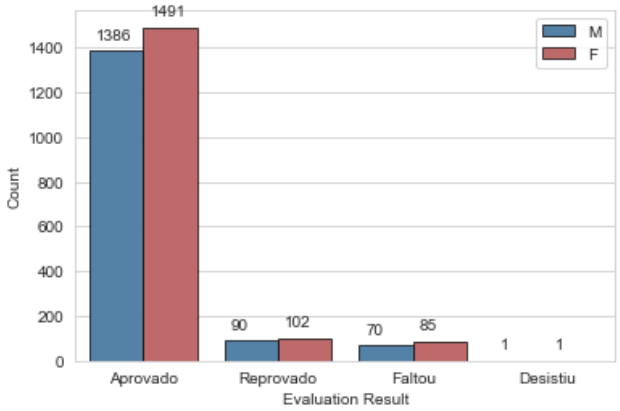

Figure 16 - Evaluation Result occurrences per Gender

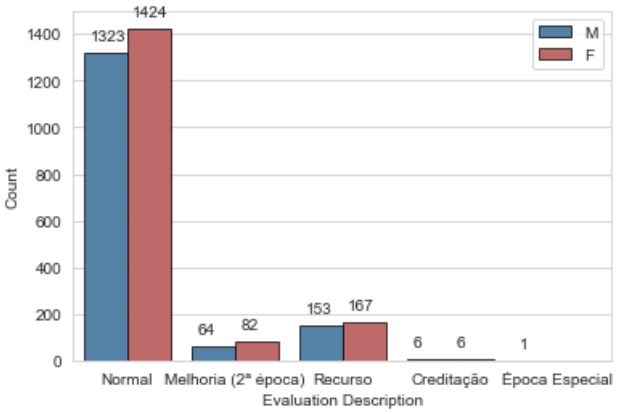

Figure 17 - Evaluation Description occurrences per Gender

Figures 16 and 17 show the evaluation result and evaluation description per gender, respectively. As expected, there are more female students in each of the categories.

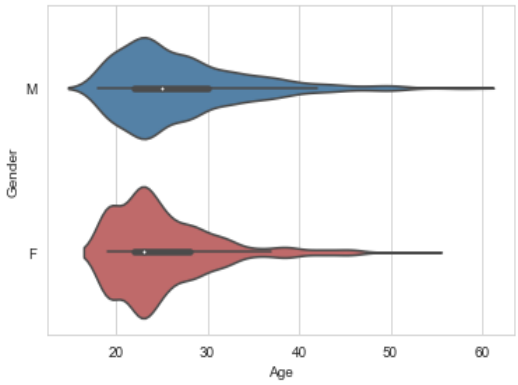

Figure 18 - Kde plot for Age variable per Gender

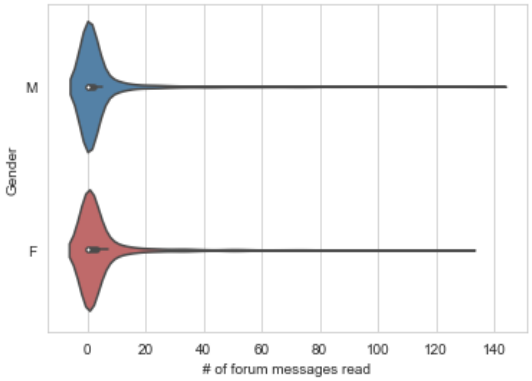

Figure 19 - Kde plot for # of forum messages read per Gender

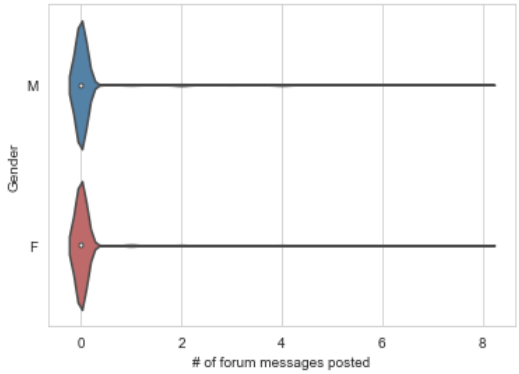

Figure 20 - Kde plot for # of forum messages posted per Gender

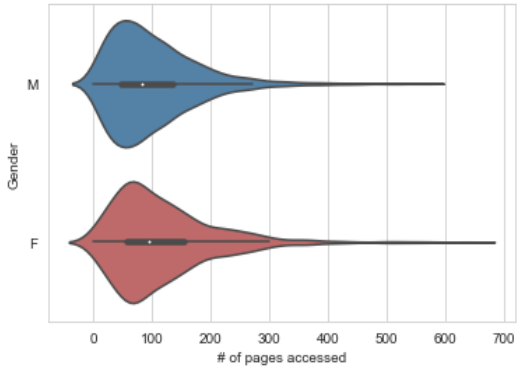

Figure 21 - Kde plot for # of pages accessed per Gender

Figures 18 to 21 show several Kernel Density plots for age and Moodle activity variables per gender. All of these follow the same structure of the ones previously analysed. Considering the Age variable, it is possible to observe that a male student is the oldest of all students in the sample. Plus, female students seem to have more density around the 20 to 30 range than male students. The distribution of age in the male students seems to be more varied in the sense that the curve is smoother from start to end. The female age distribution curve is way less smooth, having big density in the ranges mentioned previously and thinning before the male one does.

In terms of the # of pages accessed, the female students seem to have a more varied distribution. Not only the maximum value is achieved by a female student, but the density is slightly bigger than the male distribution, all around.

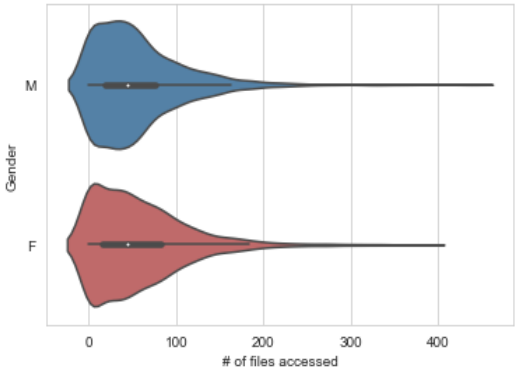

Figure 22 - Kde plot for # of files accessed per Gender

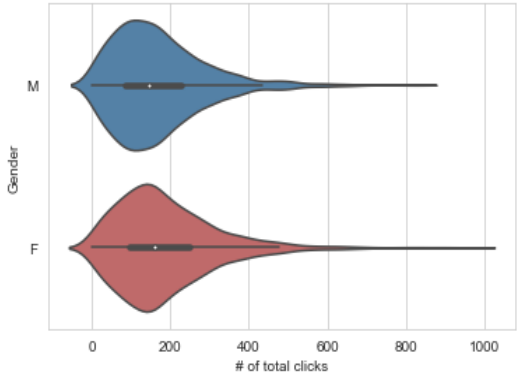

Figure 23 - Kde plot for # of total clicks per Gender

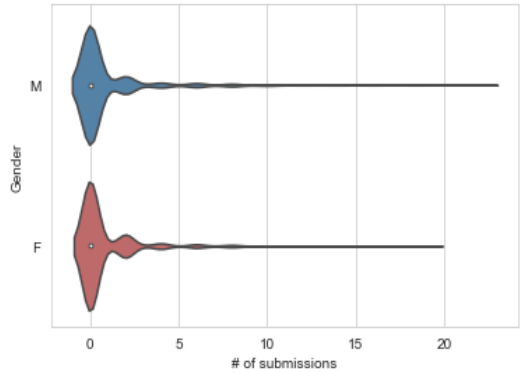

Figure 24 - Kde plot for # of submissions per Gender

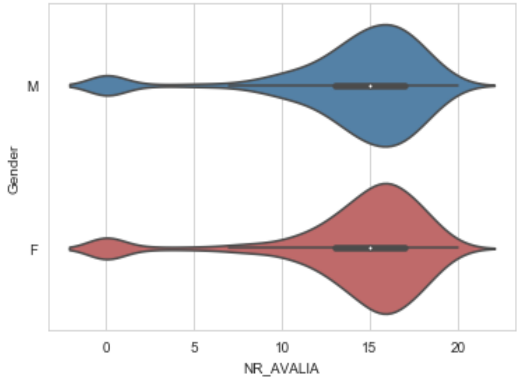

Figure 25 - Kde plot for Grade variable per Gender

Figures 22 to 25 show several Kernel Density plots for grade and the remainder of Moodle activity variables per gender. All of these follow the same structure of the ones previously analysed. In terms of the # of files accessed, we can observe that the maximum value is obtained by a male student. In contrary to the # of total clicks, which is maximum is obtained by a female student. The # of submissions is almost the same in both genders, except for the maximum value, 20 for the female gender and 22 for the male gender. To conclude, the grade variable distribution is almost identical in both genders.

Marital Status variable was explored by counting the different occurrences in the dataset. Moreover, Kernel density estimation plots were computed for age, Moodle activity and grade variables for each Marital Status value.

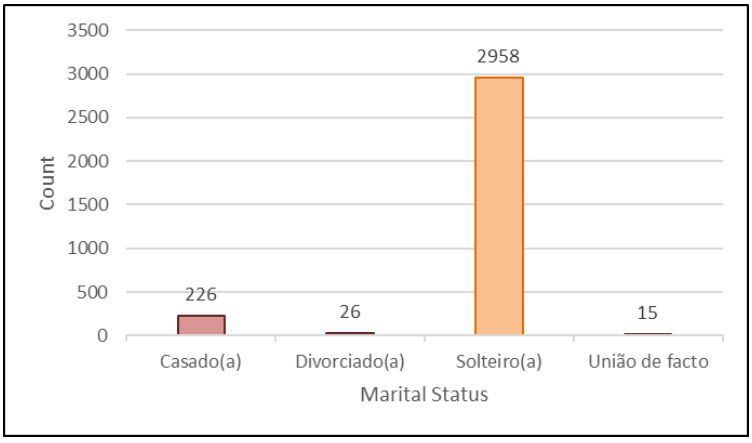

Figure 26 - Number of occurrences per Marital Status

Most of the students in this sample are single, represented by “Solteiro(a)”. In fact, only 226 are married (“Casado(a)”), 26 divorced (“Divorciado(a)”), and 15 in a consensual union (“União de facto”).

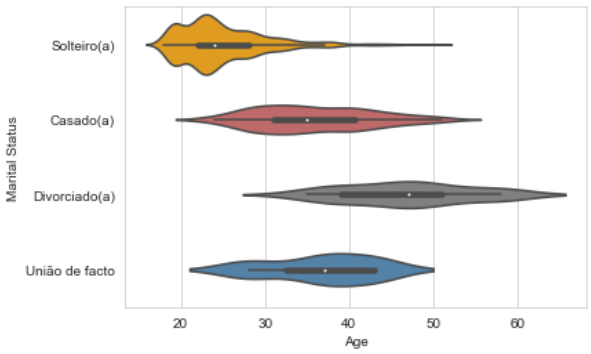

Figure 27 - kde plot for Age per Marital Status

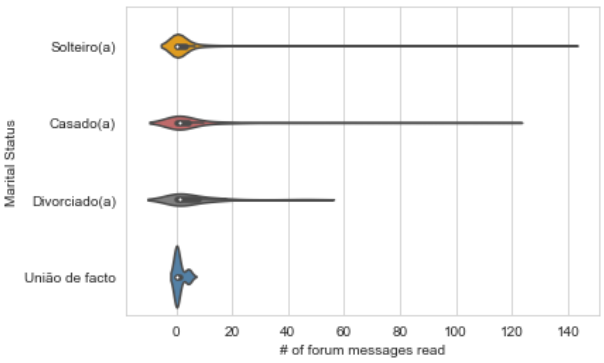

Figure 28 - kde plot for # of forum messages read per Marital Status

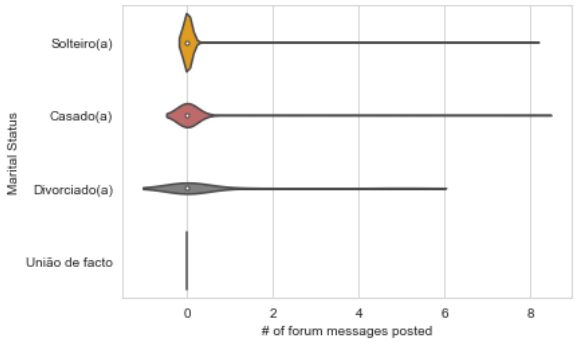

Figure 29 - kde plot for # of forum messages posted per Marital Status

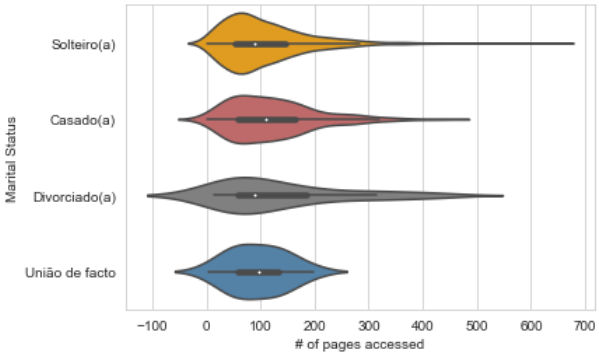

Figure 30 - kde plot for # of pages accessed per Marital Status

Figures 27 to 30 show several Kernel Density plots for age and Moodle activity variables per marital status. In terms of age, we can observe that single students are the youngest, as expected, and the oldest student is divorced. Additionally, a single student read the maximum number of messages in a course forum. Regarding the # of forum messages posted in a course forum, the maximum value of this forum doesn´t belong to a single student, but to a married student. Again, not a lot of analysis can be done in the two Moodle forum activity variables because of the low values it holds. Concerning the # of pages accessed per Marital Status, single students show a big density around the values of 50 to 100, with the line extending nearly to the 700 value. 644 is the maximum value obtained in this variable and, again, a single student was the one to show this kind of activity. Married students show greater density around the 30 to 200 range, while divorced students show a more extended but thinner line, that ranges from 0, all the way to 500. Lastly, the students in a consensual union show values ranging from 0 to 200, showing a consistent density all around, just thinning on the edges.

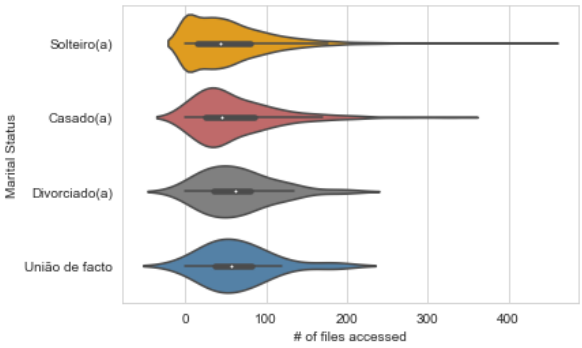

Figure 31 - kde plot for # of files accessed per Marital Status

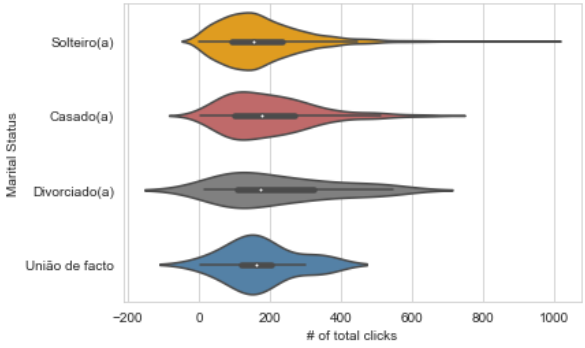

Figure 32 - kde plot for # of total clicks per Marital Status

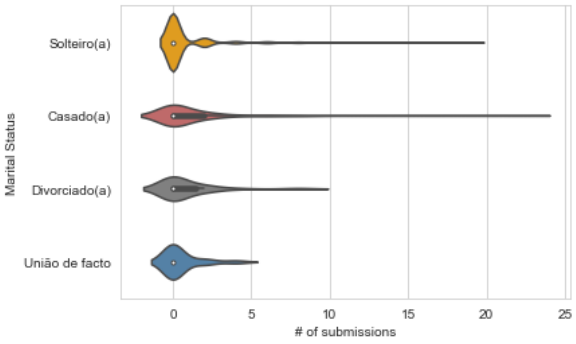

Figure 33 - kde plot for # of submissions per Marital Status

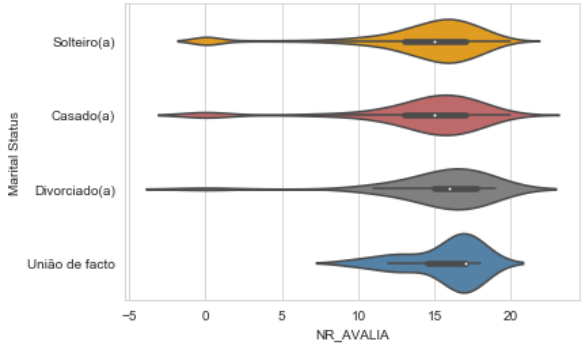

Figure 34 - kde plot for Grade per Marital Status

Figures 31 to 34 show several Kernel Density plots for the remainder of Moodle activity variables and grade per marital status. Regarding the number of files accessed, the maximum value was obtained by a single student. Moreover, we can see that divorced students tend to access more files than the rest of the marital status categories. In terms of the number of total clicks single, married, and divorced students have a similar density curve, although the divorced category density line is thinner and longer. Students in a consensual union show bigger density in the value range of 100 to 200.

The number of submissions, again, is hard to analyse due to the low values that this variable has. It is possible to observe that all the marital status categories show the biggest density around the 0 to 1 value. The maximum number of submissions is obtained by a married student.

In the matter of grade, all the marital status categories show similar curves. The students in consensual union obtained the biggest mean out of all categories. Additionally, these students appear to have more density in the grade range of 16 to 18 than their peers.

Nationality is another variable present in the Netpa data made available. Figure 35 below, shows the number of students per nationality in the sample.

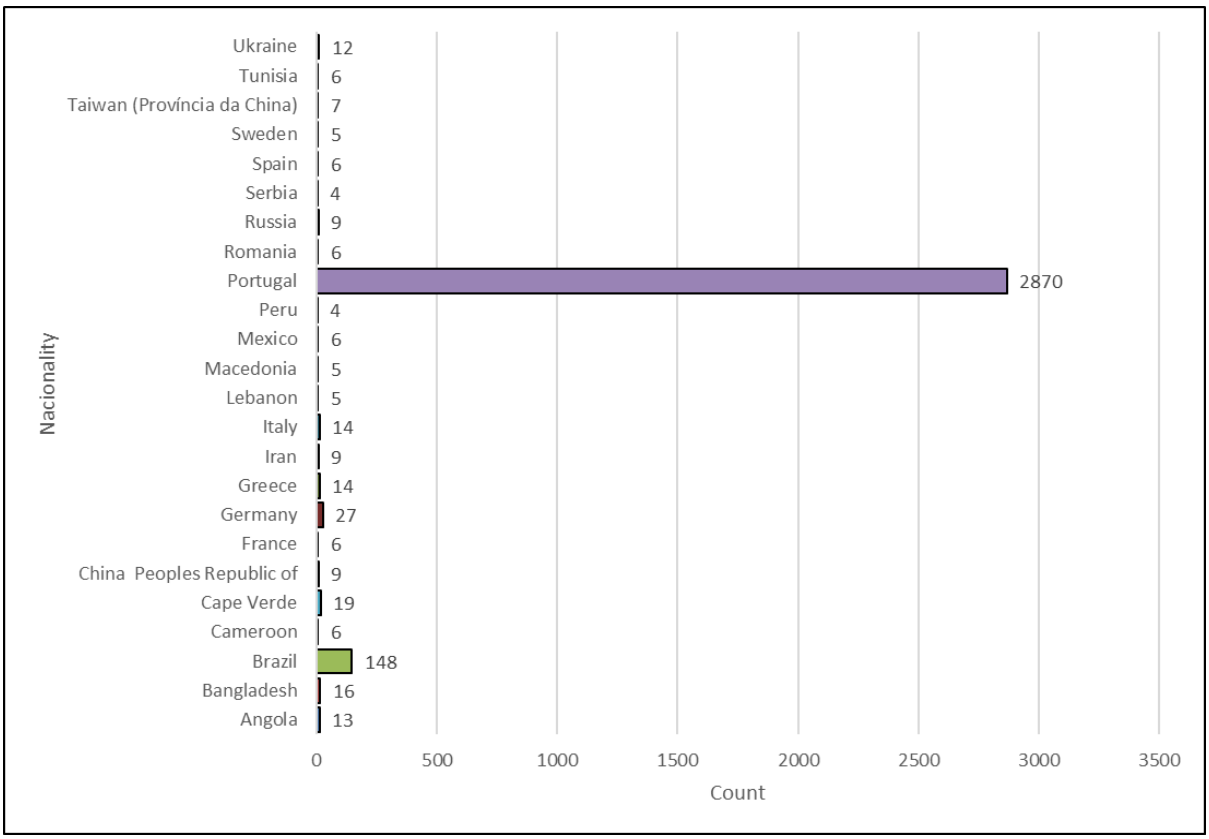

Figure 35 - Occurrences per Nationality

There are 24 different nationalities. As expected, Portugal is the most common value in the sample, as Nova IMS is a Portuguese college, with 2870 occurrences, Brazil comes next with 148 occurrences.

This next set of visualizations, show several Kernel Density plots for the Moodle activity, age, and grade variables, with the evaluation result as hue. Although evaluation result is just a descriptive variable and will not be used for modelling, by plotting it in combination with the Moodle activity variables we can understand if different results show different patterns of activity. The students who passed the course are represent by the green colour, and the ones who didn´t are represented by red. Students who quit or missed the evaluation will not be analysed.

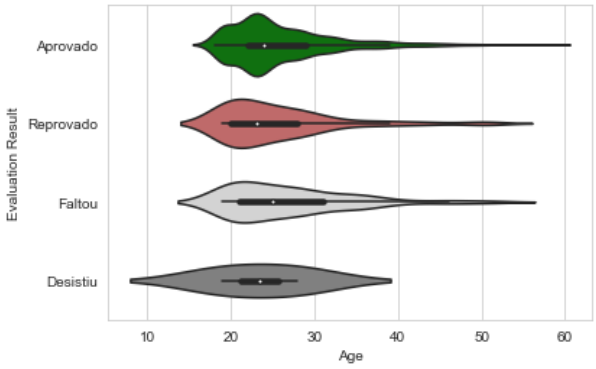

Figure 36 – kde plot for Age per Evaluation Result

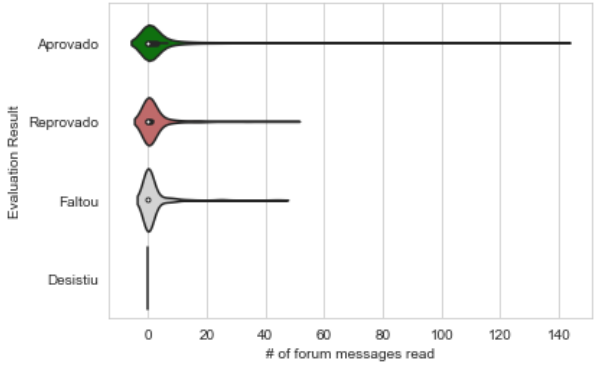

Figure 37 - kde plot for # of forum messages read per Evaluation Result

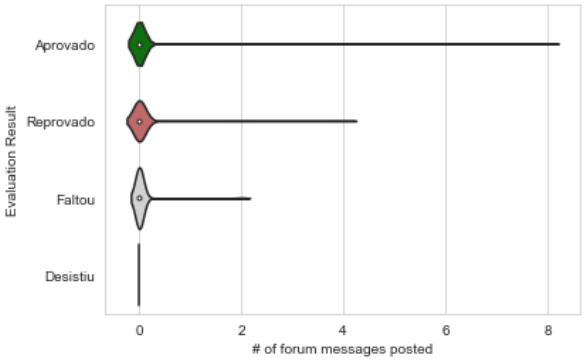

Figure 38 - kde plot for # of forum messages posted per Evaluation Result

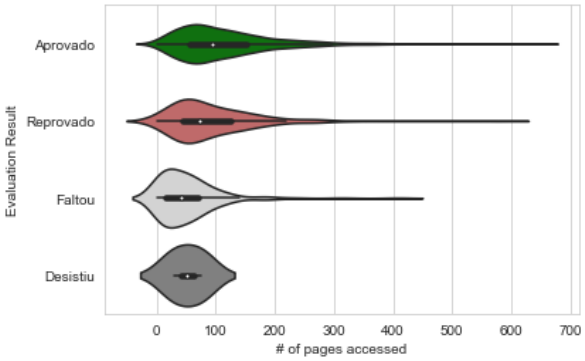

Figure 39 - kde plot for # of pages accessed per Evaluation Result

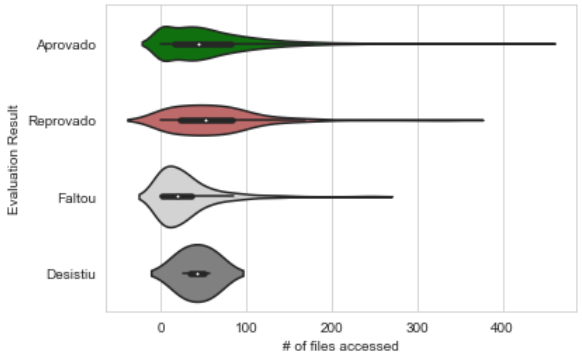

Figure 40 - kde plot for # of files accessed per Evaluation Result

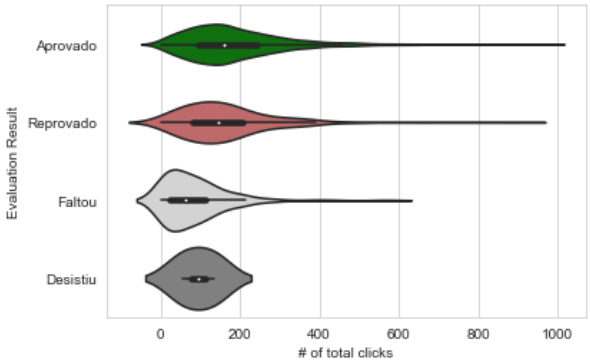

Figure 41 - kde plot for # of total clicks per Evaluation Result

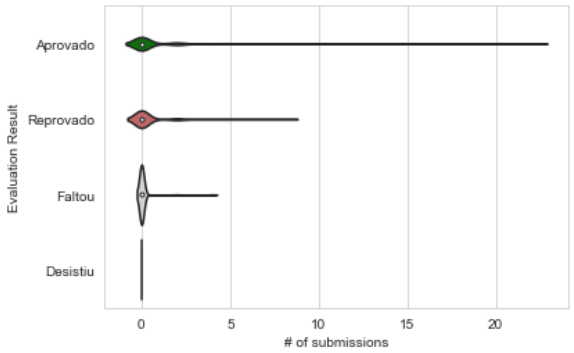

Figure 42 - kde plot for # of submission per Evaluation Result

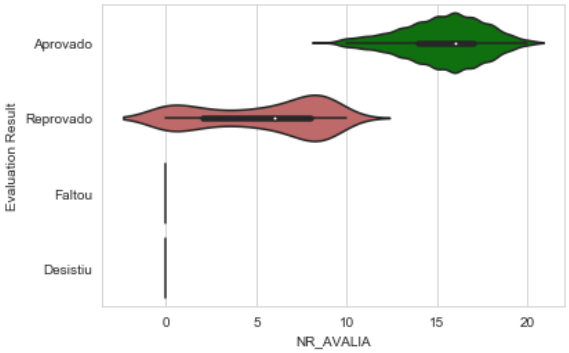

Figure 43 - kde plot for grade variable per Evaluation Result

Looking at the figures, we can observe that several variables show different densities in approved or failed students. Most of the differences are very subtle, but we can see that the majority of the graphics show that approved students tend to have a bigger value mean than failed students. Despite this, the information displayed in figure 40 is an exception. Although the maximum value for the number of files accessed is obtained by an approved student, the mean value for the failed students, represented by the white dot, is bigger than the mean value of their peers.

Moreover, we can observe that all the maximum values for the variables were obtained by approved students.

The visual difference between the density curves of approved and failed students seem to indicate that there is a difference in the engagement with the system between these 2 groups. Previously on this report was stated that McCuaig & Baldwin (2012) studied this effect and concluded that students who were more engaged and showed more activity in the learning management systems had better results.

3.1.2.3. Outlier detection

Outlier detection is a mandatory task in any machine learning problem. The NIST/SEMATECH e-Handbook of Statistical Methods (2003) defines an outlier as "an observation that lies an abnormal distance from other values in a random sample from a population". In other words, it is a value that is extremely distant from the rest in the sample with a low number of occurrences.

Most machine learning algorithms do not work well with outliers, since they are sensitive to the range and distribution of attribute values. Extreme values can mislead the training process resulting in longer training times, less accurate models, and ultimately poorer results.

Outliers can occur in many ways:

- Measurement or input error.
- Data corruption.
- True outlier observation.

Boxplots allow for outlier analysis and visualization. Figure 44 below shows an example boxplot. The dots represented in green are considered outliers by this method. These values are below the first quartile minus 1,5 times the size of the interquartile range or over the third quartile plus 1,5 times the size of the interquartile range.

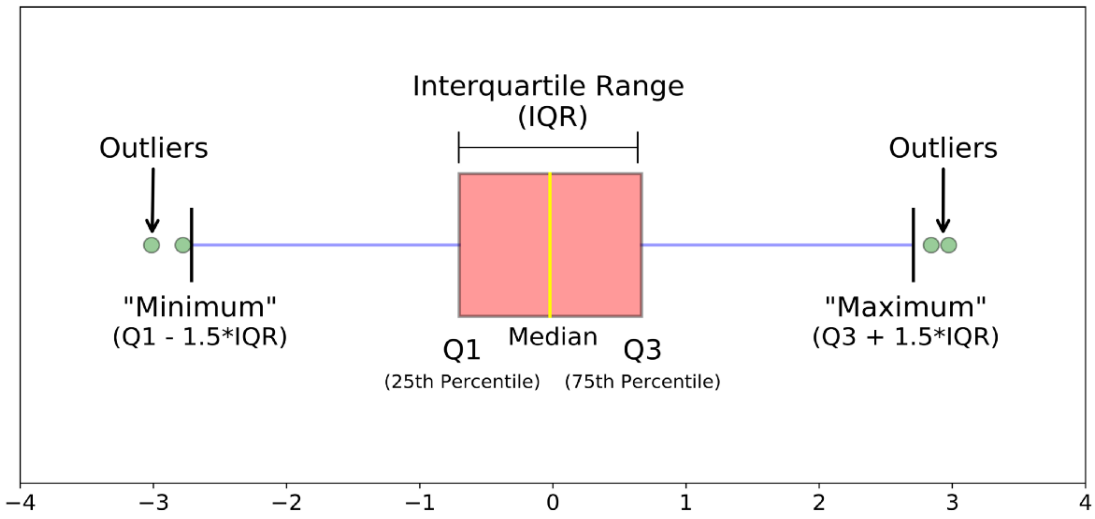

Figure 44 - Boxplot example

Despite this, sometimes while constructing boxplots, long lines of outliers will form. These lines can be misleading and hard to analyse because the values may not be extremely separated from the rest, thus not being considered outliers.

With that being said, only the most extreme and visually separated values, according to the variable boxplot, were considered outliers and removed from the sample.

Note that, due to the uncertainty of the outlier analysis, every model was tested with or without outliers.

3.1.2.4. Variable correlation

Finally, Pearson´s correlation was calculated. According to Peter Samuels (2014), Pearson correlation measures the existence and strength of a linear relationship between two variables.

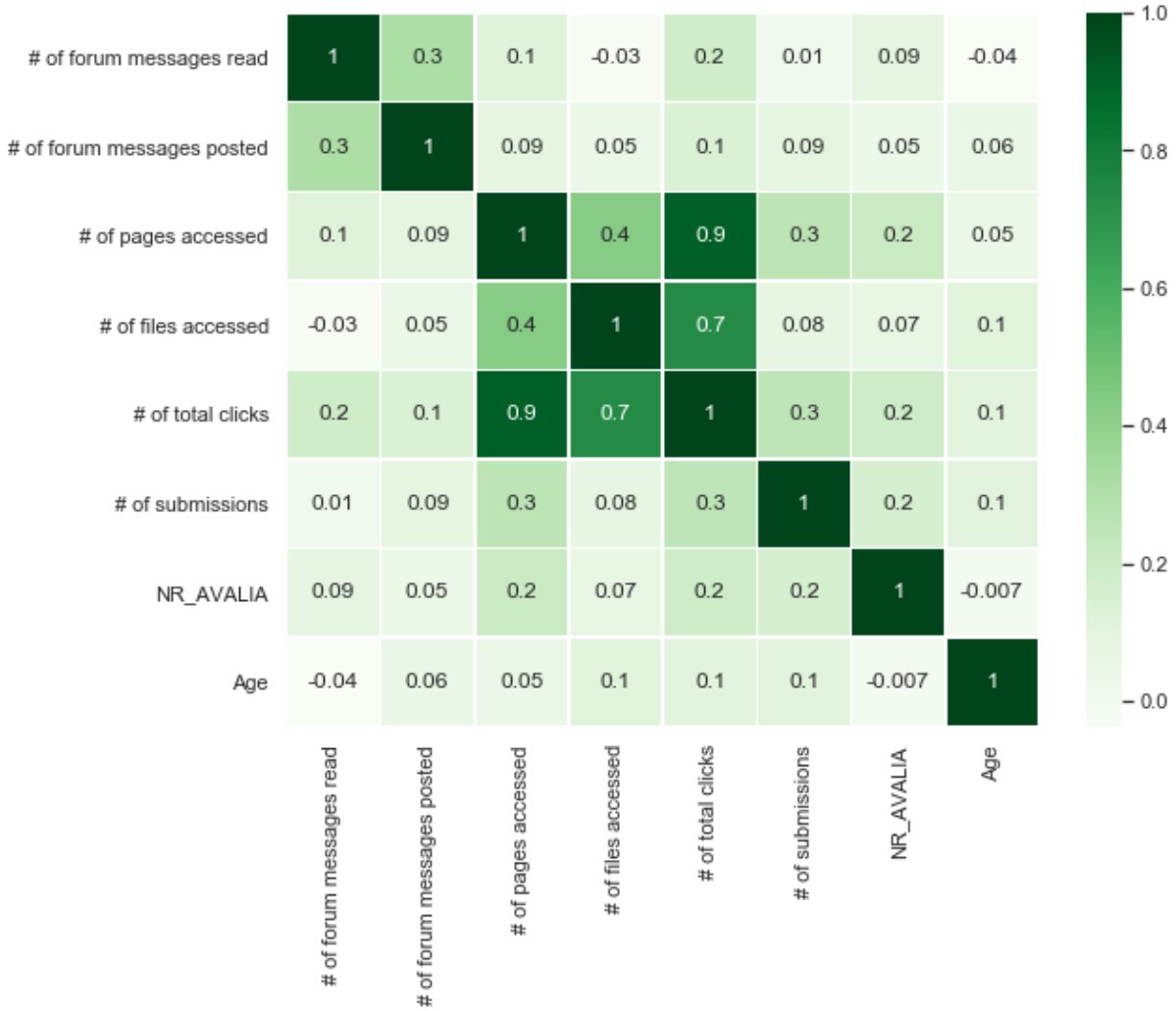

Figure 45 - Pearson´s correlation Heatmap

Looking at the figure above, we can observe that there is no variable highly correlated with the label variable (NR_AVALIA). Moreover, we see that there is a high correlation between the variables # of total clicks and # of pages accessed (0.9), and # of total clicks and # of files accessed (0.7). Knowing what these variables represent, it was expected that their correlation is high.

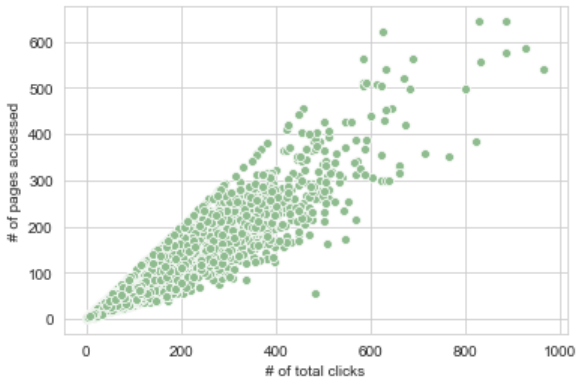

Figure 46 - Scatterplot for # of total clicks and # of pages accessed variables

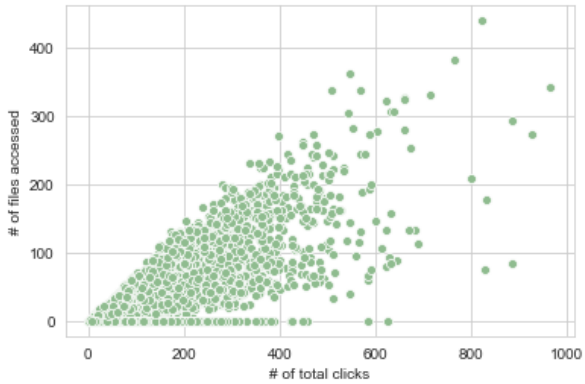

Figure 47 - Scatterplot for # of total clicks and # of files accessed variables

Observing the scatterplots for the pairs of variables mentioned above, we can confirm the strong linear relationships between them.

It is known that high correlated variables, in general, don´t improve models. Plus, a simpler model is preferable, and, in some sense, a model with fewer features is simpler. Because of this, feature selection methods will be tested in the modify phase.

3.1.3. Modify

The objective of the modify phase is to transform the data, in order to reach a better end result, after training a model. Data transformation might refer to techniques for handling missing data, outliers, and scaling. The outcome of this phase is a clean dataset that can be passed to the machine learning algorithm.

3.1.3.1. Missing Data

Missing data, has defined by Allison (2001), is “data that are missing for some (but not all) variables”. Handling missing data is a mandatory and regular task in any data related project. The majority of real-world datasets have missing data due to the anomalies or the natural flow of information in the gathering of said data. Most of the machine learning algorithms do not work with missing values.

In the context of this study and because I was the one creating the dataset used, not a lot of missing data can be found. The only occurrence is the grade variable for the students that either quitted or missed the assessment. Because the missing data occurs in the label variable and it represents a small percentage (4,8%), the decision made was to drop these observations.

3.1.3.2. Dummy Variable Encoding Categorical variables

Most machine learning algorithms, has the ones used in this study, cannot operate on categorical data directly. All data needs to be in a numeric form (Machine Learning Mastery, 2020).

Dummy Variable encoding identifies the distinct values of a variable and transforms them into binary columns. For N distinct values you need N-1 columns to represent them.

This technique was used to transform the categorical variables in this study, Gender and Marital Status, into a numeric form.

Table 5 below, shows an example of how the encoding transforms the gender variable.

Original Gender Value	Dummy variable encoded Gender variable
M	1
F	0

Table 5 - Example of Dummy variable encoding for the gender variable

Although Nationality variable is also a categorical variable, the same technique was impossible to use, due to the number of distinct values the variable presented. For this reason, the decision was to encode the variable with a "1", if the student was from Portugal and a "0" otherwise. Transforming the variable this way will show if a student from the university home country has better performance than a foreign student.

Table 6 below, shows an example of how the encoding transforms the nationality variable.

Original Nationality Value	Encoded Nationality Value
Portugal	1
Germany	0
Macedonia	0
Bangladesh	0

Table 6 - Example of encoding for the nationality variable

3.1.3.3. Label Encoding Grade variable

Similar to input variables, the label variable in this study also needs to be transformed. Grade variable was collected in the form of an integer (e.g., 8, 15, 19), making it impossible to use in a classification problem without proper modification. The first task implemented was defining performance classes.

Table 7 below, demonstrates the mapping used in the definition of said classes.

Grade Variable Range	Performance Class
>= 9	Poor
>=10 && <=14	Medium
>=15 && <=17	Good
>=18	Excellent

Table 7 - Mapping used between numeric grade and performance classes

Lastly, a label encoder was used. To transform the performance classes into an integer value. According to Scikit-learn (2020 a.), this technique encodes target labels with value between 0 and n_classes-1 and it should be used on the target variable, not on the input.

Table 8 below, displays how the technique transformed the performance class values.

Performance Classes	Label encoded performance classes
Poor	0
Medium	1
Good	2
Excellent	3

Table 8 - Label encoding for performance classes

3.1.3.4. Outlier Removal

The technique used for outlier detection and removal was explained in the explore section of this report. Has said before because outlier removal is a difficult task, only extreme values will be removed. Both datasets, with and without outliers will be tested in the model phase.

Dataset	Number of Observations	% of Removed Observations
With Outliers	3069	--
Without Outliers	3054	0.49%

Table 9 - Dataset Observations

3.1.3.5. Data Sampling

Class imbalance is a very common problem in machine learning applications. This means that each class is unevenly represented in the dataset (Chawla, Bowyer, Hall & Kegelmeyer, 2002). Classifiers trained on imbalanced data, are prone to make a biased learning model, with a predictive accuracy that performs worse on the minority class (Labels that do not have the biggest representation in the dataset), when compared to the majority (Label with the biggest representation in the dataset) (Zheng & Jin, 2020).

Analysing table 10 below, both datasets created until now, show this kind of disparity between labels. The label “Good” is the majority class in both instances, as well.

Class	Number of occurrences	%
Poor	192	6.3%
Medium	947	30.9%
Good	1429	46.6%
Excellent	501	16.3%

Table 11 - Class Distribution for the whole dataset

Class	Number of occurrences	%
Poor	191	6.3%
Medium	947	31.0%
Good	1420	46.5%
Excellent	496	16.2%

Table 10 - Class Distribution for the dataset without outliers

One of the most used techniques to deal with this problem is re-sampling the original dataset, “either by oversampling the minority class and/or under-sampling the majority class” (Chawla et al., 2002).

In the context of this problem and taking into consideration the small sample of data available, under sampling was excluded from being a feasible solution. A lot of data would have to be removed in order to reach a balanced dataset.

For this reason, the technique used was over sampling, in the form of the synthetic minority oversampling technique.

Smote was proposed by Chawla, Bowyer, Hall & Kegelmeyer in 2002. It introduces “synthetic” observations that are a close approximation of neighbour samples in the training data. The authors describe the process of oversampling the minority class “by taking each minority class sample and introducing synthetic examples along the line segments joining any/all of the k minority class nearest neighbours”.

Technically Smote generates samples by:

1. Selecting a random example from the minority class.
2. K-nearest neighbours for the sample are defined.
3. One of the neighbours is randomly chosen and a new synthetic observation is created at a randomly selected point between the two examples in feature space.

Following the same line as before, both datasets referred in the previous section were tested with and without SMOTE, in the model phase.

3.1.3.6. Scaling

Variables, most of the time, have different scales. This means that a variable can range from 0 to 100, while other can range from 2 to 5. For some data mining algorithms, such differences in the ranges will lead to biased results towards the variables with the greater range (Larose & Larose, 2015).

In general, learning algorithms benefit from data scaling and practitioners should scale numeric variables, in order to standardize the scale of effect each variable has on the results (Scikit-Learn, 2020 b.) (Larose & Larose, 2015).

The two most common techniques for scaling are normalization (bound values between two numbers) and standardization (data transformed to have zero mean and a variance of 1). In this study 4 different scalers were taken into consideration. For every model and configuration tested a function was executed to assess which scaler presented the best result. Not scaling the data was also an option considered by the function.

The sub-sections below explain how the scalers used transform the data.

- **Min-Max Scaler**

Min-Max Scaler transforms variables by scaling each feature to a given range (Scikit-Learn, 2020 c.). Two ranges were considered, 0 to 1 and -1 to 1.

Figure 48 below, shows the formula used to transform each observation.

$$x_{scaled} = \frac{x - x_{min}}{x_{max} - x_{min}}$$

Figure 48 – Min-Max Normalization formula

- **Standard Scaler**

The standard scaler transforms the variables by removing the mean and scaling to unit variance (Scikit-Learn, 2020 d.).

Figure 49 below, shows the formula used to transform each observation, where μ is the mean and σ the standard deviation.

$$z = \frac{x_i - \mu}{\sigma}$$

Figure 49 – Standard Scaler Formula

- **Robust Scaler**

Robust scaler works by removing the median and scaling the data according to the interquartile range (IQR). The IQR is between the 1st quartile (25th quantile) and the 3rd quartile (75th quantile). This scaler is not affected by outliers as their values are not considered for scaling (Scikit-Learn, 2020 e.).

Figure 50 below, shows the formula used to transform each observation.

$$x'_i = \frac{x_i - Q1}{Q3 - Q1}$$

Figure 50 – Robust Scaler Formula

3.1.3.7. Feature Selection

Feature selection is the process of reducing the number of input variables fed to a model by selecting a subset of the most relevant features from the original ones. It helps in reducing overfitting and improving classification performance in datasets that have small training samples and high dimensionality (Chen & Jeong, 2007).

Feature selection methods can be categorized in 3 ways, Filter methods, Wrapper methods and embedded methods. Filter methods work by selecting subsets of variables based on scores given by various statistical tests for their correlation with the label variable. Pearson´s correlation coefficient is

an example of a filter method. Wrapper and embedded methods work by training a machine learning algorithm and assessing the performance of different subsets of features. The difference between the two is that wrapper methods use a machine learning algorithm independently from the final algorithm trained, that way the feature selection process and the learning process occur in different stages, while in embedded methods the final machine learning model performs feature selection itself as part of the learning process (Chen & Jeong, 2007) (Analytics Vidhya, 2016 a.).

In this study, Recursive Feature Elimination (RFE) was the method used for feature selection. RFE is a wrapper method that works by iteratively training a model, computing the ranking criterion for all features such that the best ranking is the top contributor to the model and eliminating the lowest scoring feature from the subset. This process occurs until all features are eliminated. The subset of features that gives the best overall assessment score is the one selected (Guyon, Weston & Barnhill, 2002). Note that the model used was a Random Forest Classifier with default parameters and the way of assessing each subset of features was through the F1-Score with a stratified 10-fold cross validation.

Once more, the datasets created until now were all tested with RFE and without any of the original variables removed.

At the end of the modify phase, 8 different datasets have been created:

- One with no outlier removal, no Smote applied and no RFE applied.
- One with no outlier removal, no Smote applied and RFE applied.
- One with no outlier removal, Smote applied and no RFE applied.
- One with no outlier removal, Smote applied and RFE applied.
- One with outlier removal, no Smote applied and no RFE applied.
- One with outlier removal, no Smote applied and RFE applied.
- One with outlier removal, Smote applied and no RFE applied.
- One with outlier removal, Smote applied and RFE applied.

All datasets follow the feature structure represented in table 11 below:

Field Name	Description	Value Example
ID	Identifier of the student and subject.	9383807
Age	Age of the student.	22
# of forum messages read	Number of forum messages read.	10
# of forum messages posted	Number of forum messages posted.	2
# of pages accessed	Number of pages accessed.	130
# of files accessed	Number of files accessed.	22
# of total clicks	Number of total clicks.	400

Field Name	Description	Value Example
# of submissions	Number of submissions.	2
Gender_M	Dummy variable for the male gender.	1 or 0
Marital_Status_Divorced	Dummy variable for the divorced marital status.	1 or 0
Marital_Status_Single	Dummy variable for the single marital status.	1 or 0
Marital_Status_Consensual_Union	Dummy variable for the consensual union marital status.	1 or 0
Marital_Status_Widower	Dummy variable for the widower marital status.	1 or 0
PT_Nationality	Variable that identifies if the student is Portuguese.	1 or 0
Grade_Label	Transformed grade for the student in the subject.	0, 1, 2 or 3

Table 12 – Final Dataset structure

Mind that ID is only an identifier column and will not be used for modelling.

3.1.4. Model

The next chapter of this thesis report focuses on the algorithms used for modelling and their respective hyperparameter tuning. Keep in mind that the objective of this study is to build a predictive model capable of classifying new instances, in terms of their grade, given known variables as input. All the algorithms used in this study were implemented using the Scikit-Learn library. Also note that all 8 datasets were tested with the below algorithms, this means that a total of 32 model configurations were assessed to reach the study objective.

3.1.4.1. Logistic Regression

Logistic regression was first proposed in 1958 by D. R. Cox. Since that time, its use has increased massively in social sciences and education (Peng, Lee & Ingersoll, 2002). It is a development of the well-known linear regression, which is mainly used for regression problems (Cox, 1958). Logistic regression is useful for testing relationships between "a categorical outcome variable and one or more categorical or continuous predictor variables".

The main mathematical concept behind the algorithm is the logit function (Peng, Lee & Ingersoll, 2002).

The algorithm works by associating weights to each feature in the feature set using maximum likelihood estimation. A positive value means that a feature is associated with a class and a negative value means that it is not. Since the relation between the independent variable and the feature set is not linear, the logit function is used to form the relationship.

Figure 51 below shows the representation of the Logistic regression algorithm using the Logit function.

$$\text{logit}(Y) = \ln\left(\frac{\pi}{1-\pi}\right) = \alpha + \beta_1 X_1 + \beta_2 X_2.$$

Figure 51 – Logistic regression formula

Using the logit function and given a set of features, the probability of Y belonging to a class is computed. If the result is bigger than the defined threshold, generally 0.5, then Y belongs to class 1, if not, Y belongs to class 0.

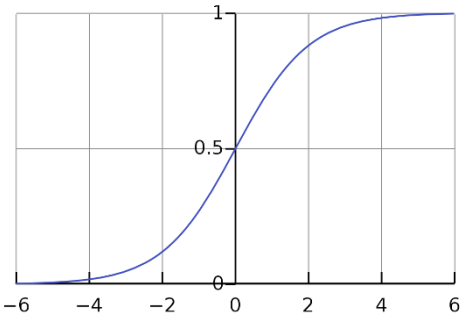

Figure 52 – Logistic Function graphical representation

3.1.4.2. K-Neighbours Classifier

K-Neighbours classification algorithm is a very simple and widely used machine learning technique. It works by classifying new instances of data based on the class of their nearest neighbours (Cunningham & Delany, 2007):

1. Computes the distance to the other datapoints in the dataset;
2. Identifies the instance K-Nearest Neighbours;
3. Uses the class labels of nearest neighbours to determine the class label of the new instance.

In this study, the distance metric used was the Minkowski distance.

$$D(X,Y) = \left(\sum_{i=1}^{n} |x_i - y_i|^p\right)^{\frac{1}{p}}$$

Figure 53 – Minkowsky Distance

To determine the class label the algorithm uses majority voting, the most straightforward approach, or the weighted distance. The latter works by assigning the nearer neighbours a bigger influence in the decision of the new instance label (Cunningham et. al, 2007).

Since the standard algorithm computes the distance of the new instance to all other in the dataset, this algorithm can be computational inefficient in some cases. To address this a variety of tree-based data structures have been implemented. In general, these structures reduce the number of calculations needed to find the K-nearest neighbours. In this study, ball tree and kd tree data structures from Scikit-Learn were tested (Scikit-Learn, 2020 f.).

3.1.4.3. Random Forest Classifier

Random Forest Classifier was proposed by Ho in 1995. This ensemble of classifiers is built using multiple decision trees that are generated using randomly selected subspaces of the feature space. Decision trees are a common machine learning method used for classification, where a tree shaped model grows progressively splitting into branches. The leaves represent class labels and branches represent rules that lead to those class labels. Studies have shown that the use of random forest can improve the predictive power of the model and control over-fitting (Ho, 1995) (Scikit-Learn, 2020 g.).

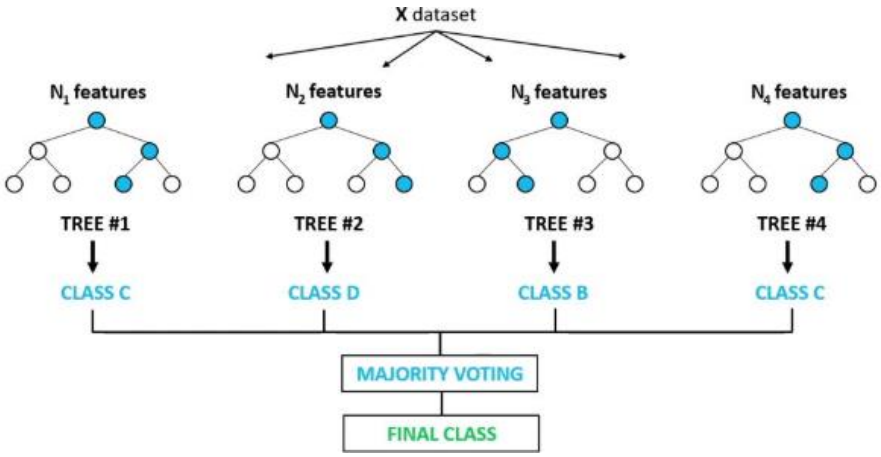

Figure 54 – Random Forest Classifier illustration

3.1.4.4. Multi-Layer Perceptron

Multi-Layer Perceptron is a class of feedforward artificial neural network. They can learn any mapping function and have been proven to be a universal approximation algorithm (Machine Learning Mastery, 2020 a.). It consists of an arrangement of layers. Each layer is composed of nodes, which one with a nonlinear activation function, that connect between each subsequent layer, with weights attributed to which connection. Moreover, some nodes receive additional information through what is called a bias. To train such algorithm, an optimization method like the gradient descent is used. In its minimal state, the algorithm is composed by 3 layers, the input, hidden and output layer, as seen on figure 55 below (Mitchell, 1997) (Nazzal, El-Emary & Najim, 2008).

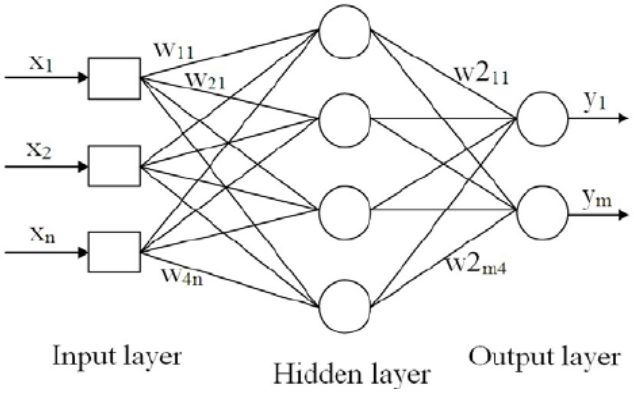

Figure 55 - Multi-Layer Perceptron architecture

3.1.4.5. HyperParameter Tuning

"Hyperparameters are values that affect the learning dynamics of a model but are not directly optimized by the training procedure" (Misra, Liam, Dunlap, Bhardwaj, Kandasamy, Gonzalez, Stoica & Tumanov, 2021). In other words, these are values that cannot be optimized from data, as opposed to model parameters, but control the learning process of a model. These must be tuned in order to reach an optimal performance.

With this in mind, Hyperparameter tuning is an essential task when building a model. Its objective is to obtain the best set of hyperparameters of an algorithm by repeatedly training it with different value combinations and selecting the one with the best metric score (Misra et al., 2021).

To achieve this, a Grid Search with stratified 10-Fold cross validation was implemented using the Scikit-Learn library.

The tables below show the different hyperparameters values tested in each algorithm.

Parameter	Scikit-Learn Description	Tested Values
Penalty	Used to specify the norm used in the penalization.	'1', '2', 'lasticnet', 'none'
Class Weight	Weights associated with classes.	'balanced', 'None'
Solver	Algorithm to use in the optimization problem.	'newton-cg', 'lbfgs', 'liblinear', 'sag', 'saga'
Multi Class	Strategy for multiclass modelling.	'auto', 'ovr', 'multinomial'

Table 13 - Hyperparameter Tuning for Logistic Regression

Parameter	Scikit-Learn Description	Tested Values
Neighbours	Number of neighbours to use.	3, 5, 10, 12, 15, 20, 25, 30
Weights	Weight function used in prediction.	'uniform', 'distance'
Algorithm	Algorithm used to compute the nearest neighbours.	'auto', 'ball_tree', 'kd_tree', 'brute'
Leaf size	Leaf size passed to BallTree or KDTree.	5, 10, 15, 20, 25, 30, 35, 40, 45, 50

Table 14 - Hyperparameter Tuning for K-Neighbours Classifier

Parameter	Scikit-Learn Description	Tested Values
Estimators	The number of trees in the forest.	1, 2, 3, 4, 5, 10, 15, 20, 50, 100
Criterion	The function to measure the quality of a split.	'entropy', 'gini'
Max Depth	The maximum depth of the tree.	None, 5, 10, 15, 20, 25, 30, 35, 40, 45, 50, 55
Max Features	The number of features to consider when looking for the best split.	'auto', 'sqrt', 'log2'
Class Weight	Weights associated with classes.	'balanced', 'balanced_subsample'

Table 15 - Hyperparameter Tuning for Random Forest Classifier

Parameter	Scikit-Learn Description	Tested Values
Hidden Layer Sizes	The ith element represents the number of neurons in the ith hidden layer.	(10), (20), (50), (100), (50, 50)
Activation	Activation function for the hidden layer.	'identity', 'relu', 'tahn', 'logistic'
Solver	The solver for weight optimization.	'adam', 'sgd', 'lbfgs'
Learning Rate	Learning rate schedule for weight updates.	'adaptive', 'constant', 'invscaling'
Batch Size	Size of minibatches for stochastic optimizers.	'auto', 20, 50, 100

Table 16 - Hyperparameter Tuning for Multi-Layer Perceptron Classifier

For each of the 32 different model configurations the best set of hyperparameters was obtained.

3.1.5. Assess

The final step in the SEMMA methodology is the assess phase. The objective of this phase is to understand and evaluate the several produced models' performance through the use of metrics. Given the different models tested, a metric that enabled the comparison between balanced and unbalanced datasets needed to be used. With this in mind, F1-Score and the ROC AUC curve were the metrics taken into consideration.

3.1.5.1. K-Fold Cross-Validation

K-Fold Cross-Validation was used to train the different models. The output metrics were the assessment values taken into consideration when selecting the best model out of the 32 configurations tested.

Cross validation is a well-known method "to assess the generalization ability of predictive models and to prevent overfitting". It works by dividing the dataset into K different folds of approximately equal size with stratified random sampling, this means that class proportions in the individual subsets reflect the proportions in the learning set. The model is then trained with K-1 subsets and assessed with the remaining one. This is an interactive process and ends when all the K subsets have served as test sets. The average of the K iterations performances is the final performance of the model (Berrar, 2018).

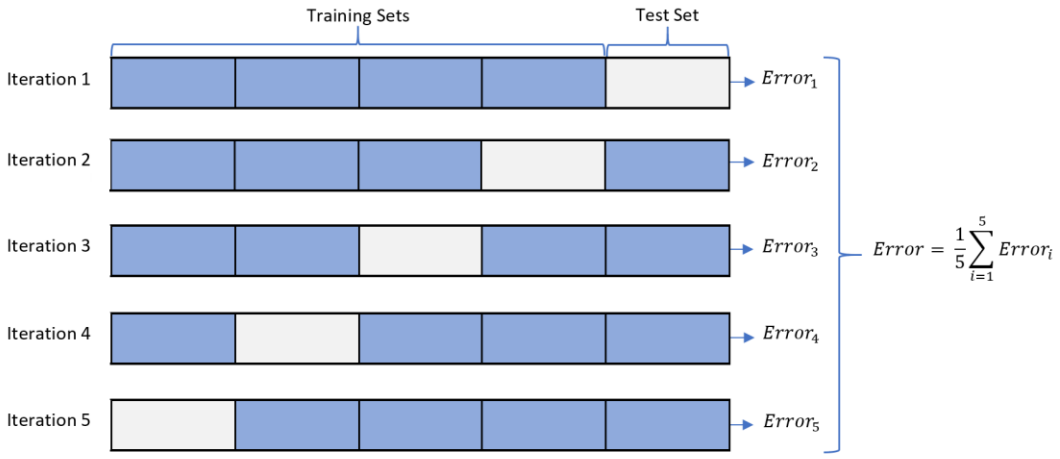

Figure 56 - K-Fold Cross-Validation

Figure 56 above, represents the K-Fold Cross-Validation process. In this study, 10 folds were used, this is the number of folds recommended by Kohavi for real world datasets (Kohavi, 1995).

3.1.5.2. F1-Score

The F1-Score is the "weighted average of the precision and recall". This metric ranges from 0 to 1, with the latter being a classifier that correctly predicts all samples of the dataset (Scikit-Learn, 2020 h.).

Figure 57 below shows the F1-Score formula.

$$F1 = 2 \times \frac{Precision * Recall}{Precision + Recall}$$

Figure 57 - F1-Score Formula

3.1.5.3. ROC AUC Curve

The Receiver Operating Characteristic curve displays the relation between sensitivity, the relative frequency of correctly classified positives, and specificity, the relative frequency of correctly classified negatives. This metric refers to the Area Under the ROC curve where a score equal to 1.0 usually means that the model has a good measure of separability, classifying all examples correctly (Kononenko & Kukar, 2007).

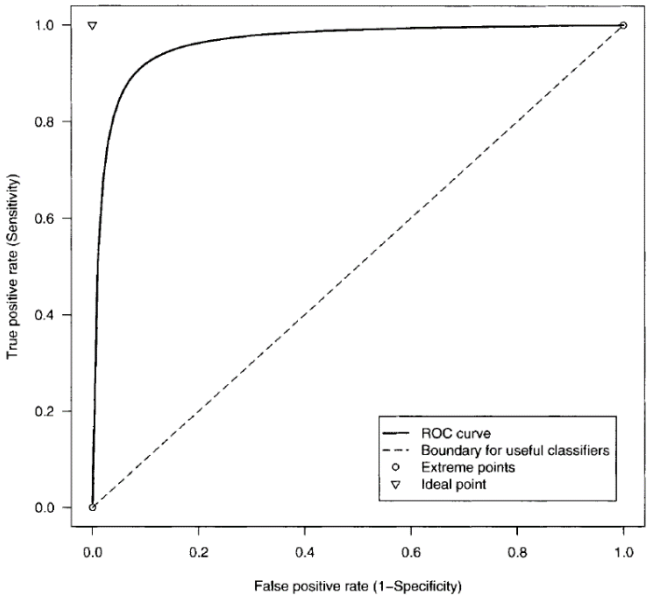

Figure 58 - ROC AUC Curve
Source: Kononenko & Kukar, 2007.

4. RESULTS AND DISCUSSION

The objective of this paper was to develop a machine learning model capable of classifying students' grades based on their activity with the Moodle platform and sociodemographic traits present in the Netpa system. For this purpose, a dataset was built from scratch joining the information from these two systems. Looking at the Moodle logs, the interactions between the students and the platform were counted and transformed into variables. Furthermore, the Netpa system information was transformed to obtain usable variables to be fed to the model. An in-depth exploratory data analysis was conducted to the variables in the dataset to reach insights about the students of Nova IMS in 2019.

Figure 59 below represent all the steps taken from the data collection phase to the final model created.

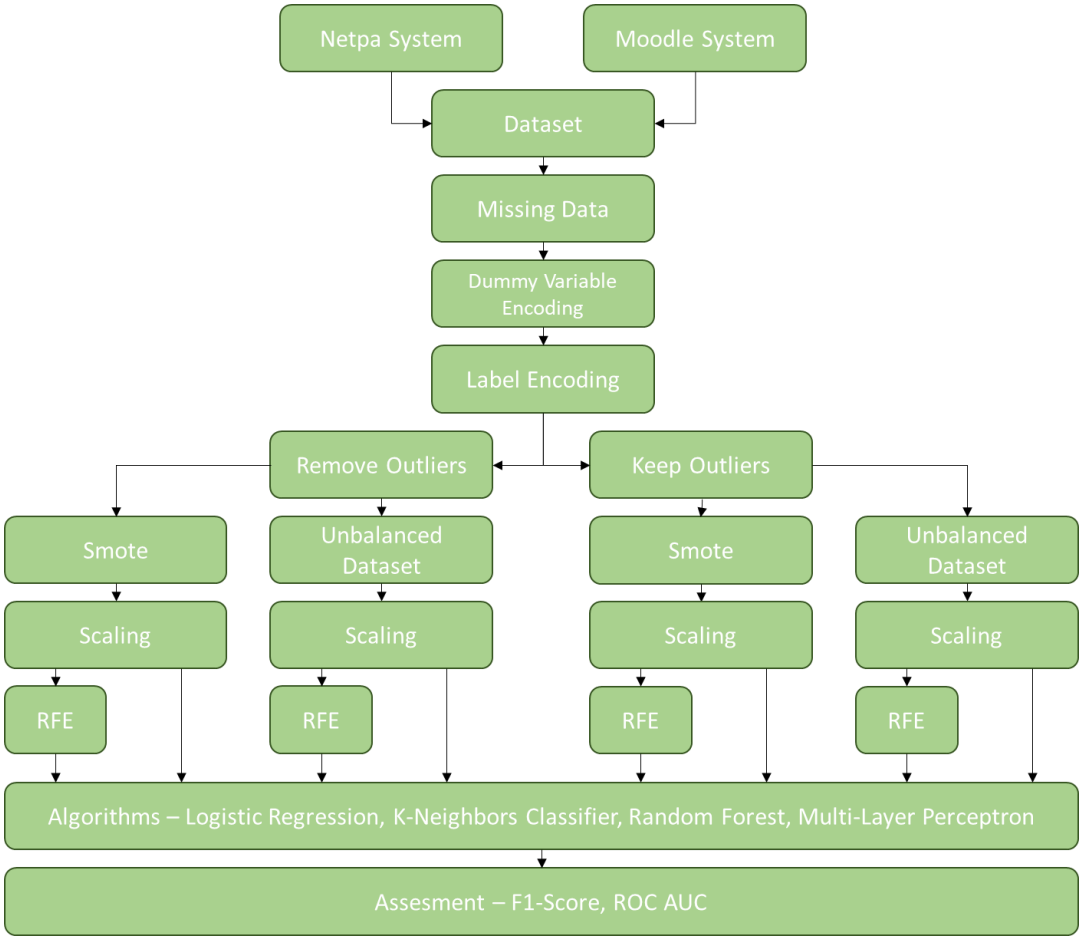

Figure 59 - Top-Down Perspective of the tasks performed

While developing the 32 different model configurations some decisions were taken. Stating off with the number of classes the label variable would assume. Most of the papers in the literature review only discriminated 2 performance classes, Pass and Fail. While this brings a better model in terms of predictions and is suitable in some applications, the objective of this study was to be able to discriminate even the passing students' performance. Has seen before, the dataset created has a lot of students that were able to obtain a passing status grade, so to be able to discriminate different levels of passing grade performance 4 class labels were used. To represent students that failed the course only one class was created. This decision is due to the really low number of grades from 1 to

10. To represent students that passed the course 3 classes were created, representing medium, good and excellent performance.

Regarding data sampling techniques, no undersampling technique was considered since the dataset created for this study is small in terms of size, to be able to reach the number of samples in the minority class almost all the dataset needed to be deleted. Although the dataset is small, when compared to the ones used in most of the studies reviewed in the literature review section of this paper, we can observe that it is substantially bigger.

Annex 8.1 shows the results obtained of all 32 different configurations. Has explained earlier in this report, cross validation was used to assess the performance of the different models. First, a comparison of the different techniques and algorithms is done using the average performance of the models. This analysis helps to understand how the different steps taken affected the performance of the models. After that, an analysis on the best 8 model configurations is performed.

Observing figure 60 below, we can see that removing or not removing the outliers in the dataset has almost no effect on the models.

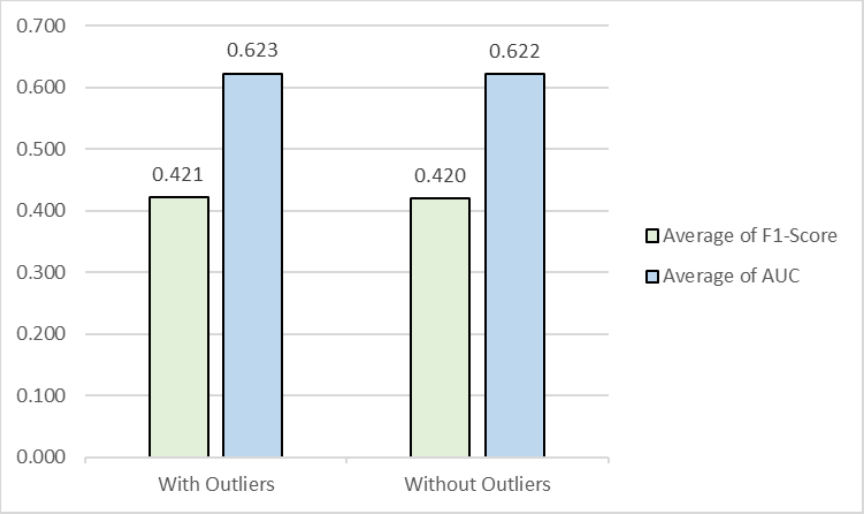

Figure 60 - Average model performance by Outlier technique

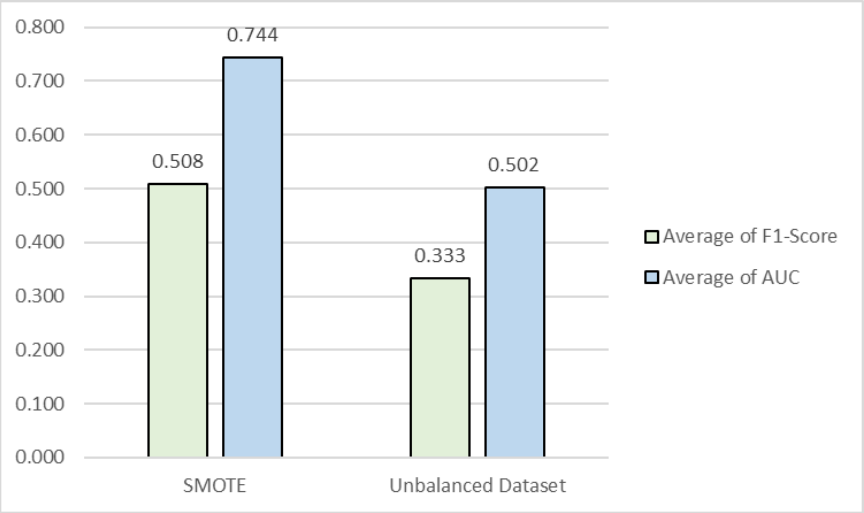

Figure 61 - Average model performance by Sampling technique

Figure 61 depicts the effect of using SMOTE sampling technique. The results show that the use of SMOTE to balance the dataset increased the F1-Score by 0.17 and the AUC by 0.242. This is an expected result as it is known that models trained in imbalanced datasets tend to have worst performance on the minority class than balanced datasets. In addition, Figure 63 represents the average effect of using RFE as a Feature Engineering technique. Almost no change is obtained when comparing the average results with and without RFE. One thing to note is that in datasets where no SMOTE was applied, the RFE algorithm picked only one variable to be used as input to the model, the # total of clicks. When SMOTE was applied but the outliers were not deleted, the algorithm picked every variable except for Marital_Status_Consensual_Union and Marital_Status_Widower, and when outliers were deleted, Marital_Status_Divorced wasn´t used as well.

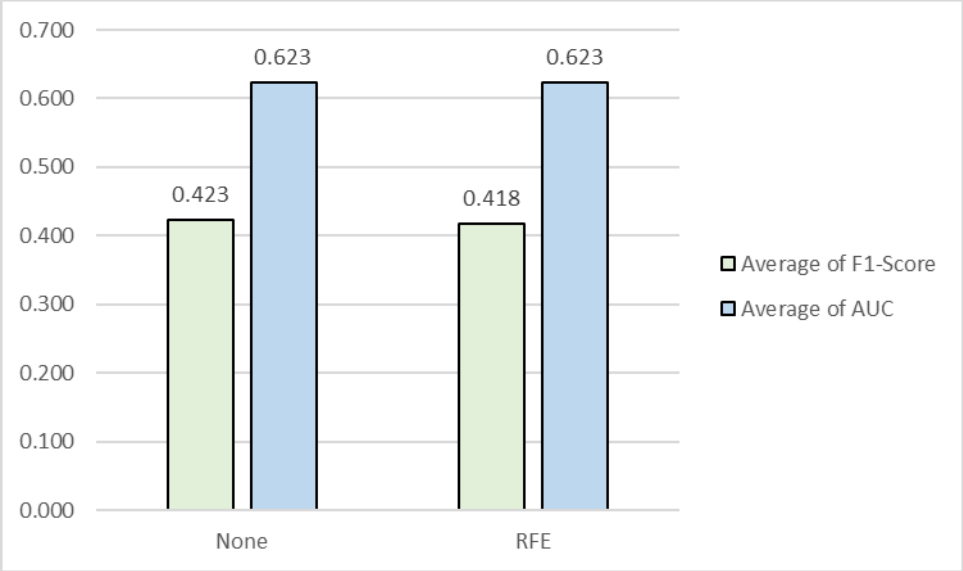

Figure 62 - Average model performance by feature engineering technique

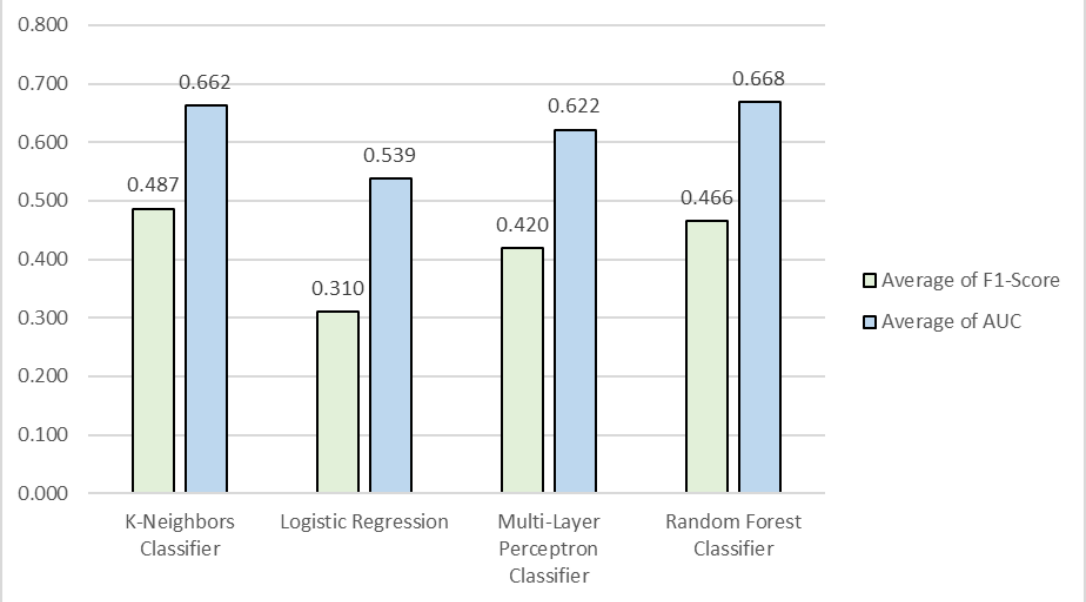

Figure 63 - Average model performance by algorithm

Figure 64 represents the average model performance by algorithm used. It is observable that K-Neighbours and Random Forest Classifier were the algorithms with the better results with Multi-Layer Perceptron just below. Logistic Regression was the algorithm with the worst performance out of the 4 tested.

Taken into consideration the assessment metrics chosen for this study, the best model configurations were selected. Table 17 shows the top 8 model configurations selected, all of them with F1-Score ranging from 0.60 to 0.62 and AUC ranging from 0.82 to 0.83.

Outliers	Sampling	Scaler	Variable Selection	Model	F1-Score	AUC
With Outliers	SMOTE	No scaler	None	K-Neighbors Classifier	0.62478	0.82873
With Outliers	SMOTE	No scaler	RFE	K-Neighbors Classifier	0.62002	0.82234
With Outliers	SMOTE	MinMax (-1,1)	RFE	Random Forest Classifier	0.61949	0.83182
With Outliers	SMOTE	MinMax (-1,1)	None	Random Forest Classifier	0.61466	0.83493
Without Outliers	SMOTE	No scaler	RFE	K-Neighbors Classifier	0.61317	0.81932
Without Outliers	SMOTE	No scaler	None	K-Neighbors Classifier	0.61315	0.81937
Without Outliers	SMOTE	MinMax (-1,1)	RFE	Random Forest Classifier	0.60801	0.83415
Without Outliers	SMOTE	MinMax (-1,1)	None	Random Forest Classifier	0.60622	0.83276

Table 17 - Top 8 Model configurations

Balancing the dataset using SMOTE was crucial to obtain good results. All the models trained with an unbalanced dataset were unusable, presenting low values for the assessment metrics chosen. Furthermore, the techniques for outlier removal and variable selection tested in this study did not produce a dominant strategy to follow, with all producing good results. In terms of scalers, for the K-Neighbours Classifier, not scaling the data produced the best results and for the Random Forest Classifier, Min-Max scaler with -1 to 1 range work the best. Regarding the algorithms used and as seen in the average performance metrics, K-Neighbours and Random Forest Classifier obtained the best results.

To summarize, the best model configuration obtained was the one with Outliers in the dataset, SMOTE sampling technique, not scaling the data, no variable selection technique and using a K-Neighbours Classifier algorithm. With this configuration the model obtained a F1-Score of 0.624 and AUC of 0.828.

Comparing the results to the papers in the literature review section is a complex task. This complication comes from the fact that different institutions follow different principles. For starters there are a lot of different computer based educational systems, with some being more advanced than others. With different systems, different variables can be used to predict performance. In addition to this, different geographies tend to grade students differently. Furthermore, a lot of the papers in the literature review described closed studies conducted by tutors and researchers, with the only objective of trying to predict final grade. These types of studies use a lot of variables that are not generally accessible through normal computer based educational systems. Other restraint on most of the papers analysed is that the study is usually only conducted in a course subject class of around 200 students, with some exception. Plus, different tutors have different ways of teaching and using support systems, this can alter the student interaction with the platform. These points are part of the reason why performance prediction is such a difficult task to generalize and build methodologies accessible to all researchers.

Despite these complications, I believe that the results shown by this study are good indicators that the variables extracted from the Nova IMS Moodle platform combined with sociodemographic variables from the Netpa System are decent predictors of the final grade obtained by a student in a certain course.

5. CONCLUSIONS

The objective of this study was to develop a model capable of classifying a student grade based on their activity on the Moodle platform and sociodemographic traits.

To achieve the objective, the following steps were taken:

- Created a dataset from the Moodle system logs and Netpa system information;
- Performed an exploratory data analysis on the dataset created;
- Created a machine learning algorithm, using known techniques, that predicts the students' performance, classifying it in 1 of 4 classes.

Moodle system logs dating from September 2019 to January 2020 were extracted and then imported to a jupyter notebook using Python and the pandas package. A counting phase proceeded to transform several student related raw logs to usable variables, representing a student engagement with the platform. Furthermore, Netpa variables extracted from the system were also imported and transformed. The variables obtained from these 2 systems were joined in one dataset with the granularity of one student per line.

Secondly, an exploratory data analysis was performed in order to give deep insights on the dataset created. Python Seaborn package helped building simple visualizations needed to study the data. Plus, several kernel density estimation plots were created, with the objective of understanding the underlaying density of the data and if different student traits would engage differently with the system.

Lastly, the modification phase occurred where several transformations were performed on the dataset and the variables, with the objective of building a more powerful and reliable prediction model.

From the 32 model configurations tested, some conclusions were drawn:

- Outlier removal did not have an effect on performance;
- Balancing the dataset with SMOTE improved the performance of the model by a substantial amount;
- The use of Recurse feature elimination to perform feature selection did not have an effect on performance;
- K-Neighbours Classifier and Random Forest Classifier were the algorithms that performed better.

Finally, the model configuration that provided the best F1-Score using a stratified 10-fold cross validation technique was the one with Outliers in the dataset, SMOTE sampling technique, not scaling the data, no variable selection technique and using a K-Neighbours Classifier algorithm. With this configuration the model obtained a F1-Score of 0.624 and AUC of 0.828.

6. LIMITATIONS AND RECOMMENDATIONS FOR FUTURE WORKS

The final chapter of this report explains the limitations of this study and recommendations for future works.

Starting off by the data collected. The dataset constructed only contained information about one semester of classes at NOVA Information Management School. It is known that the more data a model is trained on, better performance it has in real world applications. In addition, data from different academic years would be beneficial to build a more general model and to understand if students from previous years showed the same pattern of interaction with the Moodle system. Despite this, the process of transforming raw Moodle logs into usable variables took a long time to run, making it impossible to collect and process more data in time. Furthermore, the collection of more data allows for the construction of an independent test set for model assessment. Due to the low number of students in the dataset a K-Fold cross validation technique was used for the assessment of the different models.

In terms of methodology, there is room for improvement in terms of variety. A lot more methods can be tested, not only different algorithms but also different sampling techniques, feature engineering techniques and variables. Again, time was the biggest restriction to this project and not all available techniques were hypothesized.

Finally, it is proposed as future work that this paper serves as a building block for performance prediction researchers in hopes to build upon the study presented and build a new process capable of overcoming the restrictions and present a solution with a more powerful prediction capability.

7. BIBLIOGRAPHY

Agudo-Peregrina, Á. F., Iglesias-Pradas, S., Conde-González, M. Á., & Hernández-García, Á. (2014). Can we predict success from log data in VLEs? Classification of interactions for learning analytics and their relation with performance in VLE-supported F2F and online learning. Computers in Human Behavior, 31(1), 542–550. https://doi.org/10.1016/j.chb.2013.05.031

Allison, P. (2001). Missing Data. In Quantitative Applications in the Social Sciences, 72-89.

Alkhatlan, A., & Kalita, J. (2018). *Intelligent Tutoring Systems: A Comprehensive Historical Survey with Recent Developments*. http://arxiv.org/abs/1812.09628

Analytics Vidhya (2016). Feature Selection Methods. Retrieved from: https://www.analyticsvidhya.com/blog/2016/12/introduction-to-feature-selection-methods-with-an-example-or-how-to-select-the-right-variables

Baker, R. S., & Inventado, P. S. (2014). Educational data mining and learning analytics. In *Learning Analytics: From Research to Practice* (pp. 61–75). Springer New York. https://doi.org/10.1007/978-1-4614-3305-7_4

Berrar, D. (2018). Cross-validation. In *Encyclopedia of Bioinformatics and Computational Biology: ABC of Bioinformatics* (Vols. 1–3, pp. 542–545). Elsevier. https://doi.org/10.1016/B978-0-12-809633-8.20349-X

Chawla, N. v, Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: Synthetic Minority Over-sampling Technique. In *Journal of Artificial Intelligence Research* (Vol. 16).

Chen, X & Jeong J. (2007) Enhanced Recursive Feature Elimination. In Sixth International Conference on Machine Learning and Applications

Cruz-Benito, J., Therón, R., García-Peñalvo, F. J., & Pizarro Lucas, E. (2015). Discovering usage behaviors and engagement in an Educational Virtual World. *Computers in Human Behavior*, *47*, 18–25. https://doi.org/10.1016/j.chb.2014.11.028

Cunningham, P., & Delany, S. J. (2007). k-Nearest Neighbour Classifiers - A Tutorial. *ACM Computing Surveys*, *54*(6), 1–25. https://doi.org/10.1145/3459665

Daniel Larose, & Chantal Larose. (2015). *Data Mining and Predictive Analytics*

Delgado Calvo-Flores, M., Gibaja Galindo, E., Pegalajar Jiménez, M. C., & Pérez Piñeiro, O. (2006). *Predicting students' marks from Moodle logs using neural network models*.

Guyon, I., Weston, J., & Barnhill, S. (2002). *Gene Selection for Cancer Classification using Support Vector Machines* (Vol. 46).

Hung, J.-L., Hsu, Y.-C., & Rice, K. (2012). *Integrating Data Mining in Program Evaluation of K-12 Online Education*. https://www.researchgate.net/publication/264702287

Hwang, G. J., Chu, H. C., & Yin, C. (2017). Objectives, methodologies and research issues of learning analytics. *Interactive Learning Environments*, *25*(2), 143–146. https://doi.org/10.1080/10494820.2017.1287338

Kaidi, Z. (2000). *Data visualization*.

Liñán, L. C., & Pérez, Á. A. J. (2015). Education Data Mining and Learning Analytics: Differences, similarities and time evolution. *RUSC Universities and Knowledge Society Journal*, *12*(3), 98–112. *https://doi.org/10.7238/rusc.v12i3.2515*

Li, S. & Liu, T. (2021). Performance Prediction for Higher Education Students Using Deep Learning. *Complexity, vol. 2021, Article ID 9958203, 10 pages, 2021. https://doi.org/10.1155/2021/9958203*

Lu, O., Huang, A., Huang, J., Lin, A., Ogata, H., & Yang, S. (2018). Applying Learning Analytics for the Early Prediction of Students' Academic Performance in Blended Learning. *Source: Journal of Educational Technology & Society*, *21*(2), 220–232. https://doi.org/10.2307/26388400

Macfadyen, L. P., & Dawson, S. (2010). Mining LMS data to develop an "early warning system" for educators: A proof of concept. *Computers and Education*, *54*(2), 588–599. https://doi.org/10.1016/j.compedu.2009.09.008

Machine Learning Mastery (2020 a.). One Hot Encoding. Retrieved from: https://machinelearningmastery.com/why-one-hot-encode-data-in-machine-learning

Machine Learning Mastery (2020 b.). Multi-Layer Perceptron. Retrieved from: https://machinelearningmastery.com/neural-networks-crash-course

Mahmoud Nazzal, J., Nazzal, J. M., El-Emary, I. M., Najim, S. A., & Ahliyya, A. (2008). Multilayer Perceptron Neural Network (MLPs) For Analyzing the Properties of Jordan Oil Shale. *World Applied Sciences Journal*, *5*(5), 546–552. https://www.researchgate.net/publication/239580128

Mccuaig, J., & Baldwin, J. (2012). Identifying Successful Learners from Interaction Behaviour. International Conference on Educational Data Mining (EDM)

Mckimm, J., Jollie, C., & Cantillon, P. (2003). *ABC of learning and teaching Web based learning*. www.ltsn.ac.uk

Minaei-Bidgoli, B., Kashy, D. A., Kortemeyer, G., & Punch, W. F. (2003). Predicting student performance: An application of data mining methods with an educational web-based system. *Proceedings - Frontiers in Education Conference, FIE*, *1*, T2A13-T2A18. https://doi.org/10.1109/FIE.2003.1263284

Misra, U., Liaw, R., Dunlap, L., Bhardwaj, R., Kandasamy, K., Gonzalez, J. E., Stoica, I., & Tumanov, A. (2021). RubberBand: Cloud-based hyperparameter tuning. *EuroSys 2021 - Proceedings of the 16th European Conference on Computer Systems*, 327–342. https://doi.org/10.1145/3447786.3456245

Mitchell, T. M. (1997). *Machine Learning*.

NIST/SEMATECH e-Handbook of Statistical Methods, http://www.itl.nist.gov/div898/handbook/, 2003.

Pandas (2021). Software Library. Retrieved from: https://pandas.pydata.org/about/

Papamitsiou, Z., & Economides, A. A. (2014). Learning Analytics and Educational Data Mining in Practice: A Systematic Literature Review of Empirical Evidence. In *Educational Technology & Society* (Vol. 17, Issue 4).

Peña-Ayala, A. (2014). Educational data mining: A survey and a data mining-based analysis of recent works. In *Expert Systems with Applications* (Vol. 41, Issue 4 PART 1, pp. 1432–1462). https://doi.org/10.1016/j.eswa.2013.08.042

Peng, C. Y. J., Lee, K. L., & Ingersoll, G. M. (2002). An introduction to logistic regression analysis and reporting. *Journal of Educational Research*, *96*(1), 3–14. https://doi.org/10.1080/00220670209598786

Popescu, E., & Leon, F. (2018). Predicting Academic Performance Based on Learner Traces in a Social Learning Environment. *IEEE Access*, *6*, 72774–72785. https://doi.org/10.1109/ACCESS.2018.2882297

Project Jupyter (2021). Web application. Retrieved from: https://jupyter.org/index.html

R. Kohavi (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection. Proceedings of the 14th International Joint Conference on Artificial Intelligence - Volume 2, Morgan Kaufmann Publishers Inc., pp. 1137-1143.

RE Bellman (1957). Dynamic programming. Princeton University Press, Princeton, NJ, USA

Romero, C., Espejo, P. G., Zafra, A., Romero, J. R., & Ventura, S. (2013). Web usage mining for predicting final marks of students that use Moodle courses. *Computer Applications in Engineering Education*, *21*(1), 135–146. https://doi.org/10.1002/cae.20456

Romero, C., & Ventura, S. (2010). Educational data mining: A review of the state of the art. In *IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews* (Vol. 40, Issue 6, pp. 601–618). https://doi.org/10.1109/TSMCC.2010.2053532

Romero, C., & Ventura, S. (2020). Educational data mining and learning analytics: An updated survey. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, *10*(3). https://doi.org/10.1002/widm.1355

Romero, C., Ventura, S., & García, E. (2007). *Data mining in course management systems: Moodle case study and tutorial*.

Rusli, D., & Ibrahim, Z. (2007). *Predicting Students' Academic Performance: Comparing Artificial Neural Network, Decision Tree and Linear Regression*. https://www.researchgate.net/publication/228894873

Samuels, P. (2014). *Pearson Correlation*. https://www.researchgate.net/publication/274635640

Saqr, M., Fors, U., & Tedre, M. (2017). How learning analytics can early predict under-achieving students in a blended medical education course. *Medical Teacher*, *39*(7), 757–767. https://doi.org/10.1080/0142159X.2017.1309376

Scikit Learn (2020). Software Library. Retrieved from: https://scikit-learn.org/stable/index.html

Scikit Learn (2020 a.). Label Encoding. Retrieved from: https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.LabelEncoder.html

Scikit Learn (2020 b.). Scaling Methods. Retrieved from: https://scikit-learn.org/stable/modules/preprocessing.html

Scikit Learn (2020 c.). Min-Max Scaler. Retrieved from: https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.MinMaxScaler.html

Scikit Learn (2020 d.). Standard Scaler. Retrieved from: https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html

Scikit Learn (2020 e.). Robust Scaler. Retrieved from: https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.RobustScaler.html

Scikit Learn (2020 f.). K-Nearest Neighbours Classifier. Retrieved from: https://scikit-learn.org/stable/modules/neighbors.html

Scikit Learn (2020 g.). Random Forest Classifier. Retrieved from: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html

Scikit Learn (2020 h.). F1-Score. Retrieved from: https://scikit-learn.org/stable/modules/generated/sklearn.metrics.f1_score.html

Seaborn (2021). Software Library. Retrieved from: https://seaborn.pydata.org/

Shute, V. J., & Zapata-Rivera, D. (2012). Adaptive educational systems. In *Adaptive Technologies for Training and Education* (pp. 7–27). Cambridge University Press. https://doi.org/10.1017/CBO9781139049580.004

Ünal, F. (2020). Data Mining for Student Performance Prediction in Education.

Venkat, N., Year, F., & Student, U. (2018). *The Curse of Dimensionality: Inside Out*. https://github.com/nmakes/curse-of-dimensionality

Villagrá-Arnedo, C., Gallego-Duran, F. J., Compan-Rosique, P., Llorens-Largo, F., & Molina-Carmona, R. (2016). Predicting academic performance from Behavioural and learning data. *International Journal of Design and Nature and Ecodynamics*, *11*(3), 239–249. https://doi.org/10.2495/DNE-V11-N3-239-249

Zheng, W., & Jin, M. (2020). The Effects of Class Imbalance and Training Data Size on Classifier Learning: An Empirical Study. *SN Computer Science*, *1*(2). https://doi.org/10.1007/s42979-020-0074-0

8. ANNEXES

8.1. MODEL ASSESSMENT

Outliers	Sampling	Scaler	Variable Selection	Model	F1-Score	AUC
With Outliers	Unbalanced Dataset	Standard	None	Logistic Regression	0.29215	0.47675
With Outliers	Unbalanced Dataset	Standard	None	K-Neighbors Classifier	0.35028	0.51286
With Outliers	Unbalanced Dataset	MinMax (0,1)	None	Random Forest Classifier	0.33927	0.50853
With Outliers	Unbalanced Dataset	Standard	None	Multi-Layer Perceptron Classifier	0.37043	0.50652
With Outliers	Unbalanced Dataset	Standard	RFE	Logistic Regression	0.29635	0.51123
With Outliers	Unbalanced Dataset	Standard	RFE	K-Neighbors Classifier	0.36232	0.48910
With Outliers	Unbalanced Dataset	MinMax (0,1)	RFE	Random Forest Classifier	0.30066	0.49841
With Outliers	Unbalanced Dataset	Standard	RFE	Multi-Layer Perceptron Classifier	0.34082	0.50823
With Outliers	SMOTE	MinMax (0,1)	None	Logistic Regression	0.32914	0.58752
With Outliers	SMOTE	No scaler	None	K-Neighbors Classifier	0.62478	0.82873
With Outliers	SMOTE	MinMax (-1,1)	None	Random Forest Classifier	0.61466	0.83493
With Outliers	SMOTE	Robust	None	Multi-Layer Perceptron Classifier	0.47549	0.73097
With Outliers	SMOTE	MinMax (0,1)	RFE	Logistic Regression	0.32714	0.58759
With Outliers	SMOTE	No scaler	RFE	K-Neighbors Classifier	0.62002	0.82234
With Outliers	SMOTE	MinMax (-1,1)	RFE	Random Forest Classifier	0.61949	0.83182
With Outliers	SMOTE	Robust	RFE	Multi-Layer Perceptron Classifier	0.47913	0.73299
Without Outliers	Unbalanced Dataset	Robust	None	Logistic Regression	0.29088	0.47594
Without Outliers	Unbalanced Dataset	Standard	None	K-Neighbors Classifier	0.34978	0.51291
Without Outliers	Unbalanced Dataset	Robust	None	Random Forest Classifier	0.34510	0.51616
Without Outliers	Unbalanced Dataset	Standard	None	Multi-Layer Perceptron Classifier	0.36788	0.50131
Without Outliers	Unbalanced Dataset	Robust	RFE	Logistic Regression	0.29532	0.49738
Without Outliers	Unbalanced Dataset	Standard	RFE	K-Neighbors Classifier	0.36015	0.49534
Without Outliers	Unbalanced Dataset	Robust	RFE	Random Forest Classifier	0.29523	0.48919
Without Outliers	Unbalanced Dataset	Standard	RFE	Multi-Layer Perceptron Classifier	0.36714	0.52630
Without Outliers	SMOTE	MinMax (-1,1)	None	Logistic Regression	0.32520	0.58580
Without Outliers	SMOTE	No scaler	None	K-Neighbors Classifier	0.61315	0.81937
Without Outliers	SMOTE	MinMax (-1,1)	None	Random Forest Classifier	0.60622	0.83276
Without Outliers	SMOTE	Robust	None	Multi-Layer Perceptron Classifier	0.47922	0.73244
Without Outliers	SMOTE	MinMax (-1,1)	RFE	Logistic Regression	0.32418	0.58640
Without Outliers	SMOTE	No scaler	RFE	K-Neighbors Classifier	0.61317	0.81932
Without Outliers	SMOTE	MinMax (-1,1)	RFE	Random Forest Classifier	0.60801	0.83415
Without Outliers	SMOTE	Robust	RFE	Multi-Layer Perceptron Classifier	0.47637	0.73341