INSTITUTO SUPERIOR DE ENGENHARIA DO PORTO
MESTRADO EM ENGENHARIA INFORMÁTICA




# Gestão de Conhecimento de uma Instituição de Educação

**JOSÉ EDUARDO BARREIRA CABEDA**
Outubro de 2018

POLITÉCNICO
DO PORTO


# Knowledge Management from an Educational Institution

**José Cabeda**

**Dissertation to obtain a master's Degree in software engineering, Specialization in Knowledge and Information systems**

**Advisor: Carlos Ferreira**

**Co-advisor: Ana Almeida**

Porto, October 2018

# Abstract

With the emergence of the Internet, bigger computation capacity and the fall of costs in storing information came a flood of new data and the capacity to analyze it. Many areas have developed and progressed due to this, such as fintech and online advertising, but others have only now started to develop.

One of these is the educational data mining and learning analytics which has enormous potential to empower the teachers and students to be more successful.

To this end, the present work analyzes data relating the students of Software Engineer Department (DEI) of Instituto Superior de Engenharia do Porto (ISEP) with the goal of improving students' success rate, by facilitating the comprehension of the information, detect patterns in the data and predict future events directly related to the students' behavior. This work has proposed and implemented an architecture which presents the results to the user through a data portal. This portal has been divided into two components. The first agglomerates the analysis of the data while the second presents models built with Random forests, Decision Trees and Naive Bayes to predict the students' behavior.

**Keywords**: Machine Learning, Business Intelligence, Educational Data Mining, Learning Analytics, R Language

# Resumo

Com o aparecimento da Internet, o aumento da capacidade de computação e a redução dos custos de armazenamento veio uma nova onda de dados acompanhada da capacidade para os analisar. Existem já múltiplas áreas que se desenvolveram graças a estes fatores, tais como a fintech e a publicidade *online*, mas existem outras áreas que apenas agora começaram a utilizar as oportunidades trazidas por este desenvolvimento tecnológico.

Uma destas áreas é a de *data mining* na área educacional, a qual apresenta um enorme potencial para desenvolver o sucesso académico quer de alunos quer de professores. Assim, neste trabalho é proposta e implementada uma análise e extração de conhecimento dos dados relativos ao Departamento de Engenharia de Informática do Instituto Superior de Engenharia do Porto (DEI).

O presente trabalho propõe uma arquitetura, a qual apresentará os resultados através de um portal de dados. Este será dividido em dois grandes componentes. O primeiro componente agrega as análises aos dados disponibilizados, enquanto que o segundo apresentará modelos construídos com Random Forest, Árvores de Decisão e Naive Bayes para prever o comportamento dos alunos.

**Keywords**: Machine Learning, Business Intelligence, Educational Data Mining, Learning Analytics, R Language

# Acknowledgments

To my father, mother, and brother for the special support they gave me

To Joana for the innumerous words that not only made my day but also made me better

To my family, for all the days I disappeared, and they understood my needs without a word

To my friends that put up with me even when I was childish and presumptuous

To my advisors, Carlos Ferreira and Ana Almeida for their precious help in pointing me in the right direction

# Index


**1 Introduction ..... 15**

1.1 Context ..... 15

1.2 Problem ..... 15

1.3 Objectives ..... 16

1.4 Expected Results ..... 16

1.5 Value's Analysis ..... 16

1.6 Work Methodology ..... 17

1.7 Thesis Structure ..... 18

**2 Context ..... 19**

2.1 Purpose ..... 19

2.2 Satisfaction of the client's requirements ..... 19

2.3 Assumptions ..... 20

2.4 Concepts ..... 21

2.4.1 Technologies and theories ..... 21

2.5 Critical reflection ..... 21

2.6 Implications ..... 21

2.7 Solution's lifecycle ..... 22

2.7.1 Implications of solution's failure ..... 22

2.7.2 Components tests ..... 23

2.7.3 Study, previous solutions and problems ..... 24

2.8 Value Evaluation ..... 24

2.8.1 Concept of Value ..... 24

2.8.2 Solution's Value ..... 25

2.8.3 Canvas' Model ..... 26

2.8.3.1 Key Activities ..... 27

2.8.3.2 Customer Relationships ..... 27

2.8.3.3 Customer Segments ..... 27

2.8.3.4 Key Resources ..... 27

2.8.3.5 Cost Structure ..... 27

2.8.3.6 Revenue Streams ..... 27

2.8.3.7 Channels ..... 28

2.8.3.8 Value Propositions ..... 28

**3 State of art ..... 29**

3.1 Business Intelligence Tools .....29
3.1.1 Power BI .....29
3.1.2 Tableau .....31

3.2 Existing Data Mining Technologies .....32
3.2.1 Data mining: Concepts and Techniques .....32
3.2.1.1 Clustering ..... 32
3.2.1.2 Classification ..... 33
3.2.1.3 Association Rules..... 34
3.2.2 Metrics of algorithm evaluation .....35
3.2.2.1 Confusion Matrix..... 35
3.2.3 Data Mining Languages .....37
3.2.3.1 Python ..... 37
3.2.3.2 R ..... 37
3.2.4 Data mining Frameworks and Libraries.....38
3.2.4.1 Pandas ..... 38
3.2.4.2 Numpy/SciPy ..... 38
3.2.4.3 Scikit-Learn ..... 39
3.2.4.4 TensorFlow ..... 39
3.2.4.5 Keras..... 40
3.2.4.6 Weka ..... 40
3.2.4.7 R special case ..... 41
3.2.4.8 Tidyverse ..... 41

3.3 Related Work.....43
3.3.1 Educational Data Mining .....43
3.3.2 Learning Analytics.....44

**4 Evaluation of solutions ..... 45**

4.1 BI Tools.....45
4.1.1 Conclusion .....47

4.2 Data Mining Technologies .....47
4.2.1 Data Mining Language .....47
4.2.2 Data Mining Frameworks and Libraries .....48
4.2.3 Conclusion .....48

4.3 Deployment.....49
4.3.1 Cloud .....49
4.3.2 On-Premises .....49
4.3.3 Conclusion .....50

**5 Design ..... 51**

5.1 Architecture ..... 51

5.2 Engineering requirements ..... 52
5.2.1 UC1: Data analysis ..... 52
5.2.2 UC2: Prediction of students' subject failure ..... 54
5.2.3 UC3: Recompute the models ..... 55

5.3 Non-functional Requirement ..... 57
5.3.1 Functionality ..... 57
5.3.2 Usability ..... 57
5.3.3 Reliability ..... 57
5.3.4 Performance ..... 57
5.3.5 Supportability ..... 57
5.3.6 + ..... 57

**6 Implementation of the solution ..... 59**

6.1 Data Analysis ..... 59
6.1.1 Data Source ..... 59
6.1.1.1 Seasons ..... 60
6.1.1.2 Grades ..... 61
6.1.1.3 Subjects ..... 62
6.1.2 Data Model ..... 63

6.2 Prediction Models ..... 64
6.2.1 Subjects' outcome ..... 64
6.2.1.1 Models with no transformation of data ..... 64
6.2.1.2 Models with the transformation of data ..... 66
6.2.1.3 Model with precedents ..... 69
6.2.1.4 Comparison of models ..... 72
6.2.2 Students' Main Path ..... 73
6.2.3 Access to model management through an API ..... 76

6.3 Development of the Dashboards ..... 78
6.3.1 Structure of the dashboard's data ..... 78
6.3.1.1 Tables ..... 78
6.3.1.2 Columns ..... 79
6.3.1.3 Measures ..... 82
6.3.2 Dashboards for students' grades analysis ..... 83
6.3.2.1 Grades ..... 83
6.3.2.2 Final Results ..... 84
6.3.2.3 Subjects analysis ..... 85
6.3.3 Prediction's Dashboards ..... 86
6.3.3.1 Prediction of students' outcomes ..... 86

6.3.3.2 Students' most common paths ..... 88

**7 Analysis of the Case Study (LEI's Grades) ..... 89**

7.1 Grades ..... 89

7.2 Final Results ..... 91

7.3 Subject Analysis ..... 93

7.4 Prediction Analysis ..... 95

7.5 Students' most frequent paths ..... 96

7.6 Data Analysis Conclusion ..... 97

**8 Conclusion ..... 99**

8.1 Limitations and Future Work ..... 99

8.1.1 Analysis of data ..... 99

8.1.2 Evaluation of the dashboard ..... 100

8.1.3 Model Predictions ..... 101

**9 References ..... 103**

**10 Appendix ..... 107**

# List of Figures

Figure 1 - CRISP-DM cycle ..... 17
Figure 2 - Example of Power BI dashboard ..... 30
Figure 3 - Example of KNN algorithm ..... 32
Figure 4 - Bayes theorem ..... 33
Figure 5 - Decision Tree to predict a subject's grades ..... 34
Figure 6 - Confusion matrix [22] ..... 35
Figure 7 - ROC Curve ..... 36
Figure 8 - R code snippet ..... 37
Figure 9 - Pandas data frame snippet ..... 38
Figure 10 - Numpy array snippet ..... 39
Figure 11 - Scikit-Learn SVM snippet ..... 39
Figure 12 - TensorFlow Snippet ..... 39
Figure 13- arXiv mentions [29] ..... 40
Figure 14 - Interface of Weka to develop models ..... 41
Figure 15 - Main steps in a data science project [31] ..... 42
Figure 16 - Gartner's comparison of BI tools [38] ..... 45
Figure 17 - Google trends 2004 to February 2017 (Blue - Python, Red - R) ..... 47
Figure 18 - Architecture ..... 51
Figure 19 - Use cases ..... 52
Figure 20 - Sample of the data ..... 60
Figure 21 - Data Model ..... 63
Figure 22 - Set seed function ..... 64
Figure 23 - Extraction of the dataset ..... 64
Figure 24 - Conversion of numerical grades to the ECTS system ..... 65
Figure 25 - Holdout function ..... 65
Figure 26 - Caret's Decision Tree function ..... 65
Figure 27 - Decision Tree model without transformation of the data ..... 66
Figure 28 - ROC curve of the models ..... 66
Figure 29 - Matrix of students' data ..... 67
Figure 30 - Pipeline used to get students' grades matrix ..... 67
Figure 31 - SGRAI's decision tree ..... 69
Figure 32 - Subjects' CSV file ..... 69
Figure 33 - ROC curve of PPROG ..... 71
Figure 34 - PPROG's Decision tree ..... 71
Figure 35 - Conversion of the grades to transactions ..... 73
Figure 36 - Conversion of students' grades to transactions format ..... 74
Figure 37 - Cspade algorithm ..... 74
Figure 38 - Import of data to a database ..... 75
Figure 39 - Functions exposed by plumber package ..... 77
Figure 40 - API running Swagger ..... 78

Figure 41 - Student's Analysis table formula ..... 79
Figure 42 - Subject's table formula ..... 79
Figure 43 - Number of failures formula ..... 79
Figure 44 - “With Failures” formula ..... 80
Figure 45 - Switch statement for “Ano Descrição” column ..... 80
Figure 46 - Function in R to generate new columns ..... 82
Figure 47 - Formula of the average of approved exam grades ..... 83
Figure 48 - Dashboard of the grades ..... 83
Figure 49 - Second dashboard of students' grades ..... 84
Figure 50 - Subject analysis ..... 85
Figure 51 - Students' Prediction ..... 86
Figure 52 - Models' performance ..... 87
Figure 53 - Bar plot of the models' performance ..... 87
Figure 54 - API's options ..... 87
Figure 55 - Most common paths page ..... 88
Figure 56 - Students by year ..... 89
Figure 57 - Range of categorical grades ..... 90
Figure 58 - Number of years in the program ..... 90
Figure 59 - Distribution of positive grades ..... 91
Figure 60 - Average of grades by year and type of evaluation ..... 91
Figure 61 - Average of grades by year and type of evaluation considering only ML grades ..... 92
Figure 62 - Grades by phase ..... 92
Figure 63 - Number of failures for each student ..... 93
Figure 64 - Range of grades by subject ..... 93
Figure 65 - Top 5 subjects by a number of failures ..... 94
Figure 66 - Total of students by subject and final result ..... 94
Figure 67 - Students predictions ..... 95
Figure 68 - Students' most common paths ..... 96
Figure 69 - Architecture without a data warehouse ..... 100
Figure 70 - Decision Tree ALGAV ..... 107
Figure 71 - Decision Tree PESTI ..... 108
Figure 72 - Decision Tree INFOR ..... 109
Figure 73 - Decision Tree CORGA ..... 110
Figure 74 - Decision Tree ANADI ..... 111
Figure 75 - Decision Tree SGRAI ..... 112
Figure 76 - Decision Tree LAPR5 ..... 113
Figure 77 - Decision Tree GESTA ..... 114
Figure 78 - Decision Tree ASIST ..... 115
Figure 79 - Decision Tree ARQSI ..... 116
Figure 80 - Decision Tree SCOMP ..... 117
Figure 81 - Decision Tree RCOMP ..... 118
Figure 82 - Decision Tree LPROG ..... 119
Figure 83 - Decision Tree LAPR4 ..... 120

Figure 84 - Decision Tree EAPLI ..... 121
Figure 85 - Decision Tree LAPR3 ..... 122
Figure 86 - Decision Tree FSIAP ..... 123
Figure 87 – ESINF ..... 124
Figure 88 - Decision Tree BDDAD ..... 125
Figure 89 - Decision Tree ARQCP ..... 126
Figure 90 - Decision Tree PPROG ..... 127
Figure 91 - Decision Tree MDISC ..... 128
Figure 92 - Decision Tree MATCP ..... 129
Figure 93 - Decision Tree LAPR2 ..... 130

# Table’s Index

Table 1 - Advantages of tableau and Power BI ..... 46
Table 2 - Phases ..... 60
Table 3 - Mapping of the old phase system to the new ..... 61
Table 4 - Subjects ..... 62
Table 5 - Evaluation of models for each subject ..... 68
Table 6 - Evaluation of models for each subject with precedents ..... 70
Table 7 - Evaluation of each model ..... 72
Table 8 - Top 25 most common paths (from top to bottom and left to right) ..... 75
Table 9 - Most common paths with subjects ..... 76
Table 10 - Students’ grades (part 1) ..... 80
Table 11- Students’ grades (part 2) ..... 81
Table 12 - List of measures ..... 82

# Equation's Index

$Accuracy = TP + TNTP + TN + FP + FN1$ ..... 35
$Precision = TPTP + FP2$ ..... 35
$Recall = TPTP + TN3$ ..... 36
$(F1 = 2 * (precision \times recall)(precision + recall))$ 4 ..... 36

# Acronyms and Symbols

## List of acronyms

| | |
|---|---|
| **ML** | *Machine Learning* |
| **BI** | *Business Intelligence* |
| **ISEP** | Instituto Superior de Engenharia do Porto |
| **LEI** | Licenciatura em Engenharia Informática (Computer Science's Degree) |
| **DEI** | Departamento de Engenharia Informática (Computer Science's Department) |
| **IPP** | Instituto Politécnico do Porto |
| **AI** | *Artificial Intelligence* |
| **R** | Programming language mainly used for statistics |
| **ETL** | Extract, Transform, Load |
| **MOOC** | Massive Open Online Course |
| **ECTS** | European Credit Transfer and Accumulation System |
| **API** | Application programming interface |
| **AUC** | Area Under the Curve |

# 1 Introduction

The present chapter presents a synthesis of the context of this thesis (in section 1.1), the problem that it's set to resolve (section 1.2), the main objectives (section 1.3), the expected results from it (section 1.4) and the value it will bring (section 1.5). It finishes with the recommended approach to the problem (section 1.6) and how the thesis is organized (section 1.7).

## 1.1 Context

The activities at educational institutes generate a big quantity of data directly related to the students' grades, their subjects or even how many times they miss classes. This kind of data has an immense potential to improve the quality of the education but most of the time isn't adequately explored and, consecutively, the opportunity to identify problems and help improve the process in these types of institutions is lost [1].

With the continuous improvement of hardware and the prices for storing data getting lower than ever, it is becoming harder to not be overwhelmed by the enormous amount of information available. This tendency brought a new field called Big Data. This, together with the field of Machine Learning and Statistics, created an opportunity to recognize patterns and predict outcomes that can be used to understand and, more importantly, improve education.

But, although data is considered one of the most precious resources, it requires a great depth of knowledge to extract something that can be useful.

## 1.2 Problem

The situations presented in the previous section, which is common in all educational institutions, motivated the present work on *Instituto Superior de Engenharia do Porto* (ISEP), particularly in the Departamento de Engenharia Informática (DEI). As the process of manual extraction of knowledge is extensive and elaborate, the DEI's direction hopes to improve their program's quality with the development of an automatic workflow that is able to help in their decision-making process and evaluate if it's in line with their needs. Given a dataset with the students' grades and overall performance it will be analyzed and presented through dashboards to facilitate the detection of patterns and extract knowledge to answer more specific problems like the identification of students that might drop out.

## 1.3 Objectives

This work aims to:

1. Extract information from the case study (LEI-ISEP);
2. Extract knowledge that could be used to improve programs;
3. Set an architecture to automatically extract information that could be applied to any educational institute;
4. Develop a platform to share the information and knowledge acquired.

## 1.4 Expected Results

This work proposes an architecture and its respective development to automatically extract knowledge from the database of an institution.

At the end of this thesis it's expected a:

- Dashboards with information gathered from the initial dataset;
- Trained models from the initial dataset;
- Data portal presenting the information and knowledge acquired with the initial dataset.

## 1.5 Value's Analysis

This thesis value can be synthesized as "the facilitation of the data analysis related to the academic path of the students and prediction of their success or failure to improve the decision-making process of DEI's Direction".

To this end, an architecture was proposed to solve it with its implementation to be developed in a later iteration. The architecture was divided into two big segments. The first presents the analysis of the data through dashboards while the second will generate the models to predict the students' behavior. The accessibility to analyzed data and the prediction of the students' outcome will enable gains in the quality of education as well as improvements in the management of the program.

## 1.6 Work Methodology

The extraction of knowledge from DEI's database requires many steps of research and development. In this work, the main problems will be specific to data science, so we propose the CRISP-DM methodology. As the most crucial problems to solve are related to data we will start with CRISP-DM.

As can be seen in Figure 1, there are six steps [2]. In the first one, it's required a gain of some business understanding. In this case, the scope is on the students' path where we look to identify the path of most success.

In the next step, data understanding, the datasets provided by the advisor will be analyzed to create a basic understanding by applying statistics with the help of tools like Power BI.

On the third step, after having set some objectives of what metrics to be improved, the datasets must be cleaned, normalized and transformed into a format easier to be used by the models.

The modeling encompasses the applying of data mining techniques followed by the evaluation step where the created models are compared.

Finally, the best ones are selected and deployed to the desired system. Although it isn't the focus of this thesis, a web application will be developed to present the results interactively integrating with the deployment step of the methodology.

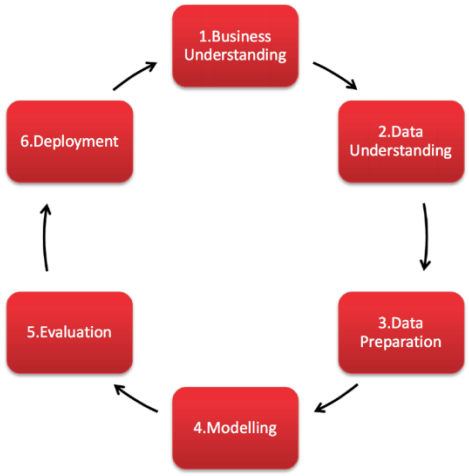



Figure 1 - CRISP-DM cycle

Each iteration will be centered on either the development of a data mining model to be deployed or the improvement of one of these steps. After each one of these, it's expected that the users of the final product, DEI's direction, will review the outputs of the models so that improvements can be implemented in a later date.

## 1.7 Thesis Structure

The present thesis has been divided into 7 chapters. The first chapter presents a structure of the context of this thesis and how it was structured.

In the second chapter, the context of the problem and the value brought by the solution is explained in more depth following the structure of “Engineering Reasoning” [3].

The third chapter presents the state of art focused on the analysis of data in the education area and reviews the main projects and tools used.

The following chapter focuses on evaluating which solutions are the best examples of the area and which techniques and tools should be used to develop the proposed solution.

After this evaluation of the state of art solutions, an architecture to solve the problem is proposed and the main use cases are set.

Finally, to evaluate if the implementation of the architecture met the necessary requirements, the metrics and methodologies were set in two parts. The first focused on the dashboard and data analysis, while the second focused on evaluating the predictive models. In the end, the conclusions were summarized, and future works were presented

# 2 Context

The presentation of the existing problem has followed the guidelines presented in the book "Engineering Reasoning" [3]. In this way, section 2.1 introduces the purpose of the present work followed by the questions to the problem at hand (section 2.2). Then, the assumptions taken in this work were presented in section 2.3 followed by the explanation of the main concepts applied in section 2.4 such as Business Intelligence, Machine Learning, and Data Mining. A critical reflection of the solutions needed is set in section 2.5 and the implications of the project are described in section 2.6. Finally, the presentation of the solution's lifecycle (section 2.7) and an evaluation of the value brought to the identified client is made (section 2.8).

## 2.1 Purpose

The main purpose of this thesis is to develop a streamlined process to analyze the data related to DEI's students and from there extract information and knowledge to be presented in a dashboard. In this way, the data is of easy access, easy to understand and always updated.

The project was developed for the DEI's direction to monitor the success rate of the students of Computer Science in ISEP. The final product should be used to assist in the decision-making process presenting the information in a clear and straightforward way.

## 2.2 Satisfaction of the client's requirements

With the introduction of this project, the client is looking for a way to improve the decision-making process and, as such, improve the teaching process and students' academic success. This can be done in two steps: 1) The analysis of the data to be presented in dashboards that synthesize and explore the data and; 2) the creation of models to predict how successful a student will be.

The first one should be able to facilitate the comprehension of the data and even detect patterns like the increase or decrease of a subject's approval rate.

The second module should not only make predictions but also be able to explain them to a certain extent so that the DEI's direction can make strategic and tactical decisions based on these.

Both modules should be presented in an obvious way, and if required, be able to present the origin of the data.

The system can be considered to have fulfilled the requirements of the client when it becomes the main tool to access this kind of data and the decisions taken are data-driven (justified by the results presented in both modules).

This project, by facilitating the process of making data-driven decisions [4], will be able to bring value to the client because, as explained in the previous section, the client hopes to improve the program and the students' academic success and one of the steps to do this is by making good decisions based on historical data.

## 2.3 Assumptions

For the first iteration of the project, the data needed for this work wasn't accessed directly from the ISEP's database but was given in an excel file and with an initial formatting and cleaning by the DEI's direction. In this step, we will assume that the information has not been tampered and is as close as possible to the original data.

In projects that require the use of datasets, there can be cases where bias needs to be taken into account as it could generate biased results (i.e., religion or ethnicity). For the current project no bias was considered, and it's assumed that the data is unbiased.

All the technologies used were as secured as possible so that only the end users can access the solution.

All the assumptions taken on this project were considered as advantageous and acceptable as they won't make the end results obsolete or less useful.

The product must be accurate and with low downtime. To serve as measures of the optimal solution being ready, the final solution must have an accuracy of 90% (see definition on section 3.2.2) and the uptime must be of >99%.

The application must use continuous delivery so that it can integrate seamlessly with existing systems and, automatically validate if the newest modules have any errors. In these cases, it should abort the deployment and alert to this event.

There should be at least three distinct environments: 1) the development; 2) the quality and; 3) the production. The development is used to create and test, the quality should be used for the client to make functional tests and, in the case of approval, deploy to the production environment. The production environment is the only one that must meet all the requirements set by the client.

The solution must be as simple as possible so that all the users, which we assume have at least a bachelor's degree and understand how the program is organized, can use the solution and immediately know how to access everything.

## 2.4 Concepts

### 2.4.1 Technologies and theories

The main technologies required are related to web development, Business Intelligence, and data mining and are reliable, secure and its theory is well documented.

First, the Business Intelligence requires some knowledge of data visualization, SQL (a language to query databases) and ETL (Extract, Transform, Load) to clean the data.

The data mining requires some knowledge of machine learning which includes the theory behind the algorithms used and the best way to format data to be used on each. Some knowledge of Data analysis and data mining programming language is also required.

Finally, the web development is a requirement as it's the main way to present the information to the user, and it needs to be fast, robust and secure to allow only the end users to access the information.

## 2.5 Critical reflection

A minimum viable solution must be able to at least present the information that is currently displayed with the help of excel in a website and access it through the automatic loading of the data. This solution can create the required minimum value to the client by automating the process of accessing and displaying the information in the most practical way (i.e., graphs and tables).

For the solution to be considered viable it must bring more value than there is (which are currently reports made on excel). The cases where the solution is rejected are:

- The analysis isn't clear to the client;
- The data on the dashboard has errors;
- The prediction models have low accuracy.

## 2.6 Implications

The data obtained for analysis and extraction of knowledge, in an initial phase, is filtered by DEI's direction making the first step of transformation and cleaning of data simpler but implies that the data hasn't suffered losses that could deteriorate its quality.

The solution will make use of matured technologies removing the uncertainty that's inherent in the development of new and immature technologies.

It is also built in a way that, as new data is stored in the database it should update the analytical and predictive modules. The rate of refresh will be set accordingly to the frequency of the decisions that will make use of the knowledge provided by this work.

Using the CRISP-DM methodology (section 1.6), the system will perform many iterations to add new models and to improve the ones already deployed.

In the future, the system could be built with data from other programs or even institutions of IPP which will improve the models and its usefulness with the increase of data.

## 2.7 Solution's lifecycle

### 2.7.1 Implications of solution's failure

The failure of the solution during the decision making can range from the progression without any hiccup to the delay of the decision-making until the system is restored. This could happen in two situations. The system on the premises is down or the models were unable to update with the newest data.

In the first situation, as long as the system is restored the solution should work unless the system isn't able to scale due to high demand (unlikely as the number of users is small).

The second situation may require more time (hours to weeks) to restore the system as there can be a major problem in the data or in the deployment process.

Although it isn't the main concern of this work, in the event of a security problem, it should be taken into account and if a crucial bug is found, the solution should be taken down until it's fixed.

The system is designed so that changes to one module don't affect in a fundamental way the other modules. However, the essential features are:

- The data sources which affect the ETL module;
- The web portal technologies and framework which, if changed, require the restructuration of the module;
- The change of BI technology which requires the development from scratch of the dashboards.

The final solution presented in this thesis doesn't have as end users the students or other external entities to the DEI's direction. As such, questions like the privacy of the data and high availability are not considered primary problems.

During the development of the modules, it is important to consider missing data that could be important to understand the models. An example of this is missing data like class attendance ratio (as stated by Researchers at SUNY Albany [6]) and extracurricular activities of the students which could be highly correlated to how successful the student is on LEI.

Although some missing data can be identified, the acquisition of it should be delegated to future work.

### 2.7.2 Components tests

During the development and deployment of the solution, it's important to test each of the modules so that we can be sure that the client is using a solution that brings value. The main tests for the components will be:

- The data rate of update;
- Tolerance to missing data;
- Tolerance to errors in data;
- Predictive model's accuracy.

For each component, some mechanisms need to be set in order to test if each iteration brings more value than the previous one. These should serve as a good measure of how the system is succeeding in implementing each requirement. Some of these tests are:

- Tolerance to the failure of the data portal;
- The accuracy of the predictive models;
- Comparison between the data mining algorithms.

In case one of the tests shows the system isn't viable, it's important to start a new iteration to detect the reason and improve. In the case a new requirement is set by the client or some of the existing requirements change, the experiments should be created or updated accordingly.

### 2.7.3 Study, previous solutions and problems

The current project has multiple similarities to other data mining solutions, especially in the educational data mining research domain (section **Error! Reference source not found.**). It is therefore important to study previous work in the area of data mining, focusing on methodologies and frameworks to analyze rectangular data (data represented by rows and columns).

Since the technologies to be used are well documented and with good support and the area of the domain is well defined, the information required to develop the solution is sufficient. The evaluation by the client is crucial as the solution has to fulfill the present needs of the client and to identify elements missing in each iteration.

The data source is the most important element of this work. If the data is incorrect, incomplete or completely unavailable, the solution may work only partially with outdated data or not work at all (not to be confused with resistance to failure).

In the first iteration, the source of data will come from excel, which was made available by the client. As one of the objectives of the final solution is to automate the process of transforming the data into the desired format, it is important, in a future iteration, to be allowed access to the relevant data from the ISEP's database as a substitute of the excel files.

As proved in the state of art chapter, the present solution is viable and can be developed.

## 2.8 Value Evaluation

### 2.8.1 Concept of Value

As stated in the paper Conceptualizing Value for the customer: An Attributional, Structural and Dispositional Analysis [7]," the term customer value is used within the marketing literature to portray both what is derived by the customer from the supplier, and also what is derived by the supplier from the customer".

Any type of work, either commercial or academic, must have at least one goal which must state who's the client, the problem and how it will bring value. This presentation of the solution to achieve this goal is denominated as a value proposition. The value proposition can be divided in three processes [8] 1) analyzing customer groups by the attributes that customers consider of value, 2) assessing opportunities in each segment to deliver superior value and, 3) explicitly choosing the value proposition that optimizes these opportunities.

### 2.8.2 Solution's Value

The client is DEI's direction represented by some professors. All of these have extensive experience in the area of education and have some sensitivity to the needs of the program. However, the manual analysis and extraction of knowledge from the data require lots of time which could lead to the decision-making process being made without the data, not due to lack of skills but due to the data being raw.

Some work in this area has already been made by the client but an automation and improvement to the existing work would remove some overhead and facilitate a change from gut-based decisions to data-driven decisions [4]. This type of decisions are not only clear, due to its (data based) nature, but also easier to track as it is represented through dashboards or other types of representations.

This thesis proposes an architecture to solve the above problems. It is divided into two big segments. The first will present the analysis of the data through dashboards in an equivalent way to the solution currently used. The second segment will answer more specific questions asked by the client through data mining.

The automation of this data can be done by setting a pipeline that accesses the original data source and applies the operations needed to format the data as it is required.

### 2.8.3 Canvas' Model

<table>
<tr>
<td rowspan="2">Key Partners<br>DEI's direction</td>
<td>Key Activities<br>Development of dashboards for the analysis of the data<br>Development of data mining models to predict student' behavior</td>
<td rowspan="2">Value Propositions<br>Automate the extraction of data<br>Analysis of the data related to the students<br>Predictions of student' behavior</td>
<td>Customer relationships<br>Development of models and dashboards to fulfill the needs of the client</td>
<td rowspan="2">Customer Segments<br>Educational institutions</td>
</tr>
<tr>
<td>Key Resources<br>Development team</td>
<td>Channels<br>Data Portal<br>ISEP's portal</td>
</tr>
<tr>
<td colspan="3">Cost Structure<br>ETL cost<br>Machine learning server<br>Data portal hosting</td>
<td colspan="2">Revenue Streams<br>Reduction in the students' dropout rate</td>
</tr>
</table>

#### 2.8.3.1 Key Activities

**Development of dashboards for data analysis**

The product will be presented as a web-based application.

**Development of data mining models to predict student behavior**

The extraction of knowledge can be done in two ways. Manually, by analyzing the information already presented to the users through graphs and dashboards in general, or through predictive models which can help answer specific questions like students about to drop out or what types of students exist.

#### 2.8.3.2 Customer Relationships

**Development of models and dashboards to fulfill the needs of the client**

The present work proposed approach is to, by multiple iterations, present small increments and, accordingly to the necessities of the client, improve the solution to optimize it for their needs.

#### 2.8.3.3 Customer Segments

**Educational institutions**

With the successful deployment of the product and its use to help the decision-making process, the product could be applied to more educational institutions or departments to bring the same value to these.

#### 2.8.3.4 Key Resources

**Development team**

During development, there will be at least one element developing the solution. After the thesis, alternatives should be considered for the maintenance and further development of the product.

#### 2.8.3.5 Cost Structure

There are three distinct fixed costs of development and maintenance of the product: the ETL, the machine learning server and the host of the data portal. Although the dataset is on premises it's expected to be the one to require more computation time to create each model.

#### 2.8.3.6 Revenue Streams

**Reduction in the students' dropout rate**

A direct consequence of decisions made by using this thesis solution would be the higher success rate of the students. This can be reflected on an increased institution rank and an increased revenue as the students that would otherwise drop the program successfully complete it.

#### 2.8.3.7 Channels

The project is intended only for internal use by the DEI's direction, not requiring any channel to promote it. However, the main channel of communication of the analysis could be a data portal built specifically for this or the integration in the existing portal of ISEP.

#### 2.8.3.8 Value Propositions

The value proposition is approached in more detail in section 1.5.

# 3 State of art

The chapter of State of art is divided into three sections: 1) Business Intelligence Tools; 2) Existing Data Mining Technologies and; 3) Related Work.

In the first section, it focuses on presenting the two main competitors in Business Intelligence, an area specialized in the preparation of the data and subsequent analysis. This is presented through the use of data visualization.

The second section describes the main languages and algorithms used for the development of models in data mining. Following the state of art of each language, some metrics to evaluate the developed models are shown, such as accuracy. The section ends with the main frameworks and libraries used to implement the models in R, Python or even Weka.

Finally, the last section provides an exploration of the main solutions that are driving the Education's data analysis. Two techniques used with more frequency are the Educational Data Mining (3.3) and the Learning Analytics (3.3.2) which have shown great results detecting new patterns of students' learning process and the reasons behind them.

The techniques used on both subjects have already been applied in other domains. One of the reasons for the late development of these in the educational field is that schools often used a paper form to store their data and have taken some time to change their process to one with relational data [9]. Another difficulty is that, after having the information stored digitally, each department and institution has its information stored in different formats.

## 3.1 Business Intelligence Tools

### 3.1.1 Power BI

Power BI [10] is a software developed by Microsoft integrated into a suite of analytical tools (office 365) and first made public in September of 2013. It was built to be used as self BI which means that it doesn't require much technical knowledge to start. It's able to connect to multiple data connections from CSV files to SQL databases or even services like Twitter. This is enabled by the use of a language called M which was specifically built to connect to different data sources.

Using their powerful query editor (powered by the DAX language and by power query) it can transform the data into new tables, columns and even measures being able to change to different structures like the star schema proposed by Ralph Kimball [11]. A powerful feature enabled by Power BI is the option to drill down or drill up. A common example of these is seen with dates. In the case, a user is analyzing a bar plot with the sales by the date the drill up lets

the user go from a granular level of days to month and the drill down does the exact opposite [12].

In terms of accessibility, as this is a Cloud-based Software as a service (SaaS) it can be accessed from the web, mobile or even be integrated into an app.

Its main features are [13]: 1) Hybrid deployment support; 2) Quick insights; 3) Cortana integration; 4) Customization and; 5) API's for integration. An example of a dashboard of Power BI can be seen in Figure 2.

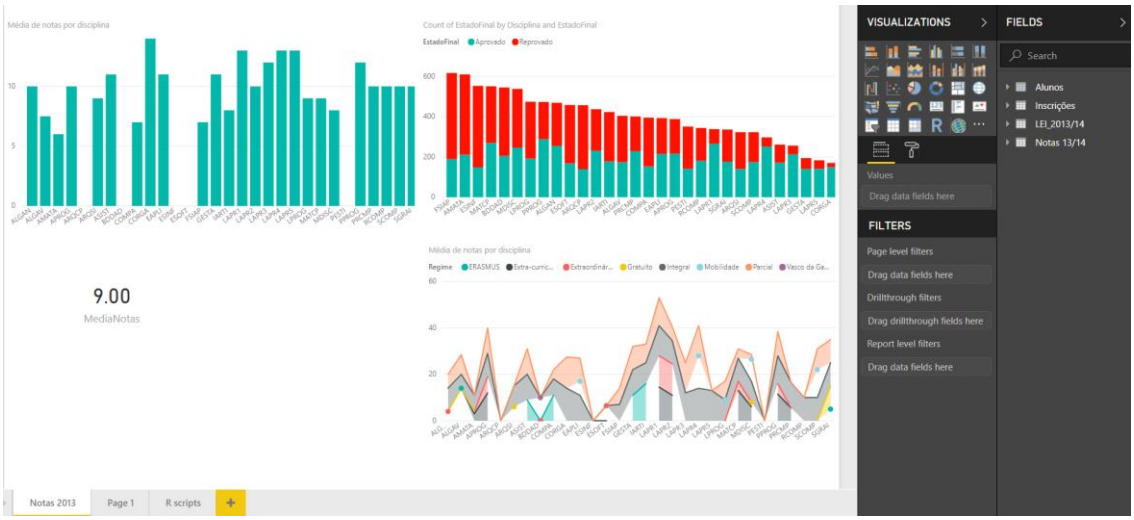



Figure 2 - Example of Power BI dashboard

### 3.1.2 Tableau

Developed by the company with the same name, the tableau is a visualization tool used with the mission of understanding data. It is divided into the following components [14]:

- Tableau Desktop;
- Tableau Reader;
- Tableau Public;
- Tableau Server;
- Tableau Online.

First, the tableau desktop, as the name implies, is a desktop software product for business analytics and data visualization. It can connect directly to a data warehouse to update data in real time or to multiple other data sources like excel files or API's.

While a user can develop the dashboards on the desktop, the tableau server can be seen as an enterprise product to publish the dashboards and share them throughout an organization with web-based Tableau server through their fast databases.

Tableau online is a version of Tableau Server hosted by the same company, functioning as a SaaS (Software as a Service), which helps to make Business Intelligence faster and easier than before. You can publish Tableau dashboards with Tableau Desktop and share them with colleagues.

The Reader is a free desktop application that enables you to open and see visualizations that are built in Tableau Desktop. You can filter and drill down data, but you cannot edit or perform any kind of interactions.

Finally, Tableau Public is a free Tableau software which you can use to make visualizations, but you need to save your workbook or worksheets in the Tableau Server which can be seen by anyone.

## 3.2 Existing Data Mining Technologies

### 3.2.1 Data mining: Concepts and Techniques

Extraction of knowledge through algorithms depends on the data, the questions asked and critically depends on algorithms used. The data used on this thesis is presented in a format known as rectangular data (data represented by rows and columns) and has a mixture of numerical and categorical attributes. Because of this, it creates important constraints to the data mining algorithms in use, as the logic behind a prediction should be as clear as possible.

To this end, this section introduces some of the most commonly used algorithms which were divided into three sections: 1) clustering; 2) classification and; 3) recommendation.

#### 3.2.1.1 Clustering

The clustering, or cluster analysis, "encompasses a number of different algorithms and methods for grouping objects of similar kind into respective categories" [15]. One of these algorithms is the K Nearest Neighbors (KNN).

The KNN algorithm is very simple and very effective, using the entire training dataset to represent the model (it can be considered that it is memorizing the entire training data set).

Predictions are made for a new data point by searching through the entire training set for the K most similar instances (the neighbors) and summarizing the output variable for those K instances. For regression problems, this might be the mean output variable and for classification problems, this might be the mode (or most common) class value.

The techniques applied to decide the nearest data point may vary but one of the simplest ones is to measure the Euclidean distance, a number that can be calculated with only the coordinates of the data points (see Figure 3).

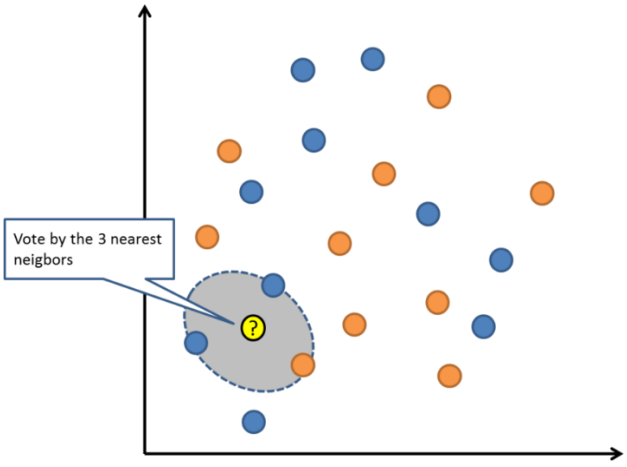



Figure 3 - Example of KNN algorithm

One of the problems of KNN is that it can require a lot of memory or space to store all the data. However, it only performs a calculation (or learn) when a prediction is needed.

In conclusion, in instances where there are very high dimensions (lots of input variables), it isn't suited as the best model as it can negatively affect the performance of the algorithm on the problem.

#### 3.2.1.2 Classification

The classification is a learning function that divides (or classifies) the data according to a predefined number of classes with the goal to organize and distribute data in different classes. This section introduces two classification algorithms, Naive Bayes and Decision trees, applied in the development of the solution.

Naive Bayes is a probabilistic algorithm which is called naive because it assumes that each input variable is independent (Figure 4). Although it's an unrealistic assumption for real data, the technique has proven to be very effective on a large range of complex problems.

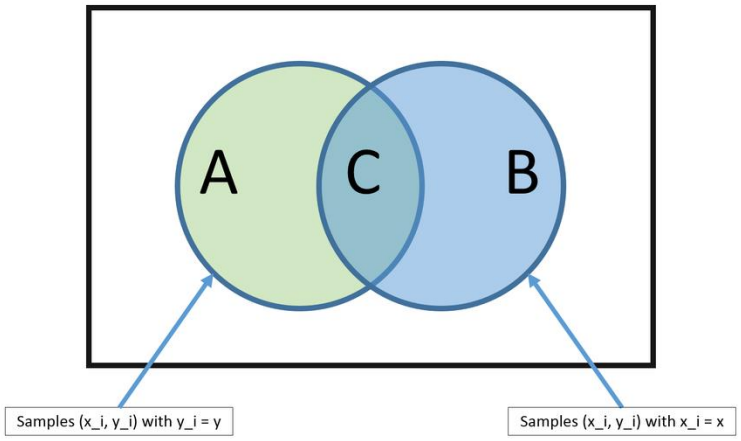



Figure 4 - Bayes theorem

The model is comprised of two types of probabilities that can be calculated directly from the training data: 1) The probability of each class; and 2) The conditional probability for each class given each x value. The formula is as below:

$$P(c|x) = \frac{P(x|c)P(c)}{P(x)}$$

Once calculated, the model can be used to make predictions for new data using the Bayes Theorem. The most used R package to implement the Naive Bayes is the e1071.

The second type of algorithm, decision trees, are one of the most widely used algorithms [17]. The representation of the decision tree model is a binary tree. As can be seen in Figure 5, each node represents a single input variable (x) and a split point on that variable (assuming the

variable is numeric). The leaf nodes of the tree contain an output variable (y) which is used to make a prediction. Predictions are made by walking the splits of the tree until arriving at a leaf node and output the class value at that leaf node.

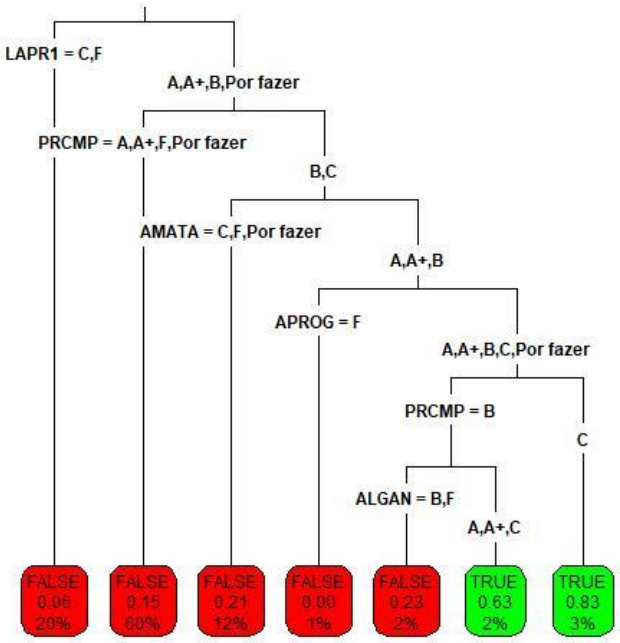



Figure 5 - Decision Tree to predict a subject's grades

The trees are known for being fast in making predictions and being accurate in a broad range of problems without requiring any special preparation for the data. One of the most used R packages to implement this type of algorithm is the caret [18].

#### 3.2.1.3 Association Rules

The association rules are often the first and most useful method for analyzing data that describe transactions, lists of items, unique phrases (in text mining), etc. Its objective is to, given a set of transactions of multiple items, find rules that will predict the occurrence of an item based on the occurrences of other items in the transaction.

To evaluate how good an association is there are three types of parameters: 1) Support; 2) Confidence and; 3) Lift. The support can be explained as the number of times in total that a given set occurs in the total of the transactions ($Support = \frac{frq(X,Y)}{N}$). The Confidence is the number of times the set occurs in the total of times the antecedent occurs ($Confidence = \frac{frq(X,Y)}{frq(X)}$). Finally, the lift is represented by the support of the set divided by the support of each operator ($Lift = \frac{Support}{Supp(X) \times Supp(Y)}$). The result of this calculation is one of the most important in detecting if the association is valid as it needs to be bigger than one. In case it's one, it means that the antecedent and the consequent are independent of each other.

One specific implementation of the association rules is the Apriori Principle which dictates that "Any subset of a frequent itemset must be frequent". By applying this, it's possible to invert the meaning and, when a superset is detected as infrequent, all of its subsets can be removed.

As the term baskets infer, one of the first use cases for using this algorithm was in associating the supermarket products that a consumer was more likely to buy together. In the R language, one of the packages used to implement this algorithm is the arules [20].

Another implementation in R of the association rules is the cspade from the arulessequences package [21] built to discover frequent sequential patterns (in the case of this work to discover frequent sequences of students through each semester).

### 3.2.2 Metrics of algorithm evaluation

#### 3.2.2.1 Confusion Matrix

The confusion matrix (Figure 6) divides the results into four types of results. The true positive and true negative is the results which were classified correctly as of a certain class or not. The false positive and false negative are the ones incorrectly classified as of a certain class or not.

| | | PREDICTED CLASS | |
|---|---|---|---|
| | | Class=Yes | Class=No |
| ACTUAL CLASS | Class=Yes | a | b |
| | Class=No | c | d |

a: TP (true positive)
b: FN (false negative)
c: FP (false positive)
d: TN (true negative)

Figure 6 - Confusion matrix [22]

The accuracy, precision, recall, and $F_1$ are four measures based on this matrix. The accuracy (formula below) measures how well the model can correctly predict if it's of a certain class or not. It's usually represented as a percentage.

$$\left(Accuracy = \frac{TP + TN}{TP + TN + FP + FN}\right) 1$$

The precision of a model refers to how concentrated the results and the more the results are close to each other the higher the precision. It is calculated by dividing the true positive results by the total of positive results (formula below).

$$\left(Precision = \frac{TP}{TP + FP}\right) 2$$

The recall is a metric used to measure the sensitivity of the model. It is the fraction of relevant instances that have been retrieved over the total amount of relevant instances (formula below).

$$\left(Recall = \frac{TP}{TP + TN}\right) 3$$

Precision and recall can be collapsed to a single performance measure known as the F1 measure which offers a useful alternative to the simpler misclassification rate [23]. The F1 measure is the harmonic mean of precision and recall and is defined as the formula in the formula below.

$$(F_1 = 2 * \frac{(precision \times recall)}{(precision + recall)}) 4$$

Finally, to compare multiple models and detect which ones are the best at predicting the desired outcome, we can use the ROC curve. Developed in the 1950s for signal theory as a means to analyze the noisy signal, it characterizes the trade-off between positive hits and false alarms [24]. The ROC curve is represented with true positives (TP) in the y-axis against the false positives (FP) on the x-axis.

To compare two or more models, the area under the curve is measured and the greater it is the better. If it is below 0,5 (see Figure 7) it means it predicts more times incorrectly than correctly. Accordingly, the closer it is to 1 the better, as it means the model predicts more cases successfully.

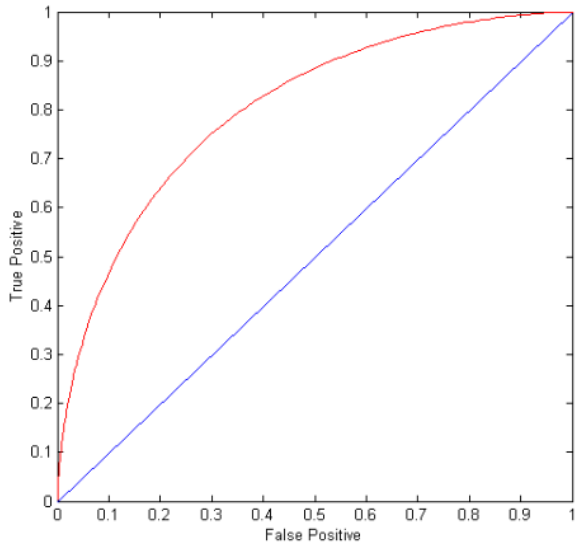



Figure 7 - ROC Curve

### 3.2.3 Data Mining Languages

When it comes to data mining there are two main languages to work with: Python and R. These languages, although with different backgrounds, have grown to become the two main options for data science. In this section, we present for each of these languages some background, the main features, libraries, and frameworks.

#### 3.2.3.1 Python

Conceived in the late 1980's by Van Rossum, Python is an open source object-oriented programming language with dynamic types. The dynamic types make the language cleaner and easier to learn and, grouped with automatic memory management, remove the need for manually allocating memory.

It has grown to the last distribution of Python 3, released in 2008. This last version, as a result of the many years of development, has many redundant features, which has produced the emergence of a division in the community between versions two and three. Nonetheless, the main libraries have decided to end support to Python 3 in 2018 and the scientific community should merge again into a single version.

The languages ease of use together with the existence of strong data science libraries (like numpy or pandas who were built around Python) have made the language a hub for the development of statistical and data mining related work.

#### 3.2.3.2 R

Based on the S language (which in turn was developed by the Bell Labs), the R language made its first appearance in 1993 as a creation of Ross Ihaka and Robert Gentleman. Its first stable implementation was released in 2000. The main purpose of R is to introduce an open source language (GPL license) to help in the area of statistics and graphics. The easiness of use (Figure 8) and great extensibility with packages (collection of functions, data, and documentation made available with the help of a repository called CRAN) makes the language a great fit for scientists and data engineers to implement their statistical and modeling projects.

```
list <- c(2,3,4)
list
```

Figure 8 - R code snippet

### 3.2.4 Data mining Frameworks and Libraries

With an extensive number of algorithms in machine learning, a number of frameworks were developed to simplify the process. In this section, we present the most popular ones and give a brief explanation of the main features and philosophy of each one. This section not only presents the main frameworks used for data mining but also refers to the libraries (and packages in R) used to create models or transform the data.

#### 3.2.4.1 Pandas

Pandas is an open source Python library under the BSD license built to provide data structures and data analysis tools, which have become the de facto standard tool for this type of work. It introduces statistical concepts like data frames which help to analyze tabular data (see Figure 9).

```
In [6]: dates = pd.date_range('20130101', periods=6)

In [7]: dates
Out[7]:
DatetimeIndex(['2013-01-01', '2013-01-02', '2013-01-03', '2013-01-04',
               '2013-01-05', '2013-01-06'],
              dtype='datetime64[ns]', freq='D')

In [8]: df = pd.DataFrame(np.random.randn(6,4), index=dates, columns=list('ABCD'))

In [9]: df
Out[9]:
                   A         B         C         D
2013-01-01  0.469112 -0.282863 -1.509059 -1.135632
2013-01-02  1.212112 -0.173215  0.119209 -1.044236
2013-01-03 -0.861849 -2.104569 -0.494929  1.071804
2013-01-04  0.721555 -0.706771 -1.039575  0.271860
2013-01-05 -0.424972  0.567020  0.276232 -1.087401
2013-01-06 -0.673690  0.113648 -1.478427  0.524988
```

Figure 9 - Pandas data frame snippet

#### 3.2.4.2 Numpy/SciPy

Numpy and SciPy is an alternative to pandas. It's an open source extension library that provides efficient ways of running operations on arrays of homogeneous data as seen in Figure 10. It is able to provide a faster alternative to native Python by running time-consuming tasks in C and provides multiple algorithms using the SciPy, which works in conjunction with the structures from numpy.

```
>>> x = np.array([2,3,1,0])
>>> x = np.array([2, 3, 1, 0])
>>> x = np.array([[1,2.0],[0,0],(1+1j,3.)]) # note mix of tuple and lists,
    and types
>>> x = np.array([[ 1.+0.j, 2.+0.j], [ 0.+0.j, 0.+0.j], [ 1.+1.j, 3.+0.j]])
```

Figure 10 - Numpy array snippet

#### 3.2.4.3 Scikit-Learn

This framework is an open source library built around Python that originated with David Cournapeau. It is a generic one featuring algorithms like SVM (Figure 11), decision trees, random forests, and k-means. This library was built around the notion of being an extension to the SciPy library and was first released in 2010.

```
>>> from sklearn import svm
>>> clf = svm.SVC(gamma=0.001, C=100.)
```

Figure 11 - Scikit-Learn SVM snippet

It introduces a common interface to run all algorithms being responsible for the holdout, creating the models and making predictions and, with the help of SciPy, it is also able to wrangle the data into a structure suitable for analysis and modeling.

#### 3.2.4.4 TensorFlow

TensorFlow is a low level, open source framework built for numerical computation using data flow graphs. It was developed initially by Google's Brain Team to help on deep neural networks research but has grown to accommodate other domains as well. It works natively with Python (Figure 12) but there's also a wrapper to use it with the R language [25].

It's one of the most popular open source projects on GitHub (4$^{th}$ place in February 2018 [26]). It's able to run on multiple CPU's and GPU's speeding the creation of the models and has recently made available a new module, TensorFlow Lite, which has the ability to run models inside devices with low computation power like smartphones and Arduino's [27].

```
>>> import tensorflow as tf
>>> hello = tf.constant('Hello, TensorFlow!')
>>> sess = tf.Session()
>>> sess.run(hello)
'Hello, TensorFlow!'
>>> a = tf.constant(10)
>>> b = tf.constant(32)
>>> sess.run(a + b)
42
>>> sess.close()
```

Figure 12 - TensorFlow Snippet

#### 3.2.4.5 Keras

Originally built by François Chollet in 2015, Keras is a high-level neural network API, written in Python that can be run on top of TensorFlow or CNTK. Its main objective is to enable the fast iteration with deep neural networks by creating an abstraction to the other low-level frameworks.

Its guiding principles are [28]:

- User Friendliness;
- Modularity;
- Easy Extensibility;
- Work with Python.

As an open source project, Keras isn't as widely used as TensorFlow but, as of September 2017, Keras was the second most referred deep learning framework in arXiv (Figure 13), indicating that it is widely used in the academic area.

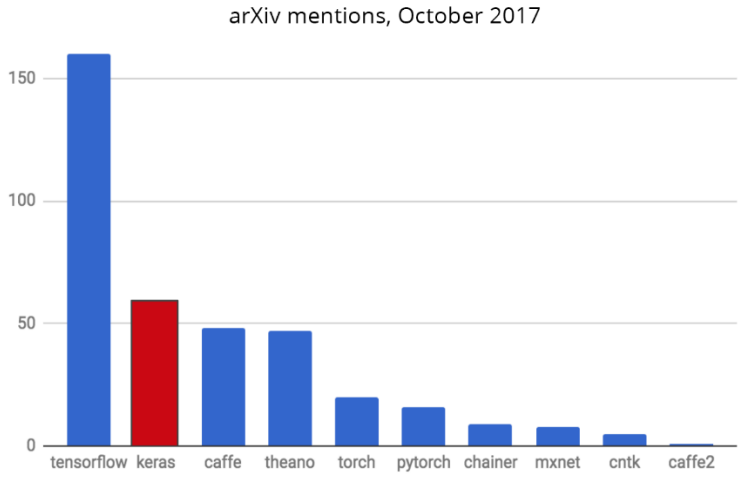



Figure 13- arXiv mentions [29]

#### 3.2.4.6 Weka

Weka is a free, open-source suite of software machine learning built in Java by the University of Waikato in New Zealand and first released in 1993.

As stated in their homepage [30], their main objectives are to "make ML techniques generally available" and "contribute to a theoretical framework for this field (machine learning)". With its easiness of use and focus on accessibility, Weka is applied not only by ML Researchers to test new algorithms but also by students to learn each concept of the field.

As presented in Figure 14, Weka was built to help the user on each step of the development of new models (following a pattern similar to CRISP-DM).

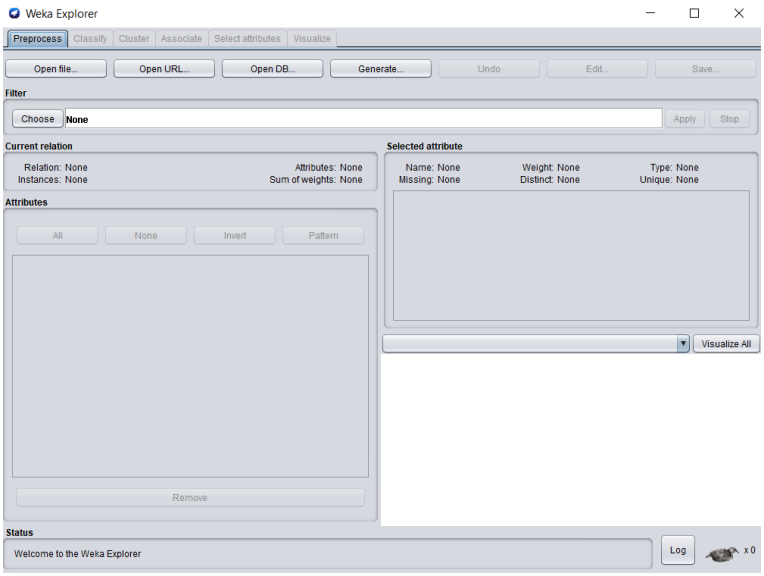



Figure 14 - Interface of Weka to develop models

#### 3.2.4.7 R special case

The above frameworks and packages all have been developed first with Python in mind, showing how strong the Python community is. However, there are stable implementations of frameworks like Keras and many of the algorithms implemented in these frameworks are available as separate packages.

#### 3.2.4.8 Tidyverse

A collection of R packages designed for data science (ggplot2, tibble, tidyr, readr, purr, and dplyr) is a standard way of preparing data for analysis thanks to the common data representations and the API design.

The purpose of each package is:

- readr, for data import;
- tidyr, for data tidying;
- dplyr, for data manipulation;
- ggplot2, for data visualization;
- purrr, for functional programming;
- tibble, re-imagination of data frames.

These packages help in the process of analyzing the data following the flow described in Figure 15.

First, the data needs to be imported into the R environment. It may come from a single source or multiple, from a CSV file or from a web API but the readr is responsible for importing it so that we can work with it.

As the data can be represented in diverse ways, it's important to tidy it up in a streamlined way. In general, the data can be represented in a rectangular way, with a column for each variable and a row for each observation. After going through this step, using the tidyr, the data should be in a way that the data scientist can focus on maximizing the analyses possible for the given data.

The transformation using the dplyr package step starts after tiding the data and refers to all operations necessary to make the data suitable to be worked upon. This usually refers to the creation of new variables narrowing on information of interest (examples like median and counts). The tibble and purrr are both packages utilized to improve the coding experience in the import and transformation steps.

The visualization done with the ggplot2 is fundamental so that the final user is able to interpret the data and detect patterns. These phases occur in parallel with the modeling which, when the question is precise enough, can be represented as a model which scales better when compared to the visualization.

Finally, the communication step is used to explain the result of the data analysis.

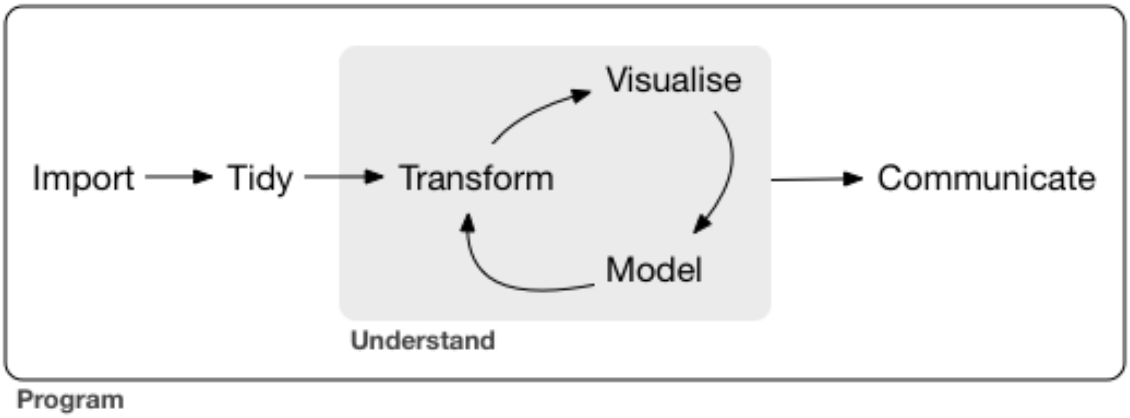



Figure 15 - Main steps in a data science project [31]

## 3.3 Related Work

### 3.3.1 Educational Data Mining

The educational data mining (EDM) is an area of study with its first conference in 2008, which looks into studying educational data generated by students and instructors by applying techniques like statistics, machine learning and data mining [9].

It has huge potential as a methodology to discover how people learn by predicting their learning and understanding their behavior. Although it's a recent field of research, it has seen considerable progress. Some state of art models is able to infer when a student is able to learn a new concept leading to the discovery of the factors that instigated it.

There are other types of models called automated detectors which, based on students' data (and metadata) can analyze the impact of decisions. An example given by Ryan S. Baker [9] is that the models can detect what decisions more likely lead to retaining a student in an academic program [32].

One big booster of this field has been the open source release of the Pittsburgh Science of Learning Center Datashop. It's the biggest dataset related to the iterations between students and educational software and has been used on a large number of papers related to EDM.

The applications of EDM can be divided into 4 groups [33]:

- Student modeling;
- Modeling of the knowledge structure of the domain;
- Pedagogical support;
- Scientific research.

These applications should be fast and simple to use so that the patterns are clear and it's easy to reason with the data. Some implementations that apply the basic analysis of this type of data exist like with Moodle, blackboard or even google analytics.

Educational Institutions have three types of teaching that they can consider nowadays. The traditional learning is done onsite, usually with students and teachers living closer geographically. The e-learning, with examples like Coursera (funded by Stanford teachers) and EDX (funded by MIT and Harvard University) which are provided exclusively through the Internet. Finally, there's the b-learning (blended learning) which mixes the first two types of learning.

While the presential learning has physical limitations, the e-learning has seen a high dropout rate [34]. The blended learning proposes the use of the e-learning platform to reach students with Internet access while keeping some steps under supervised teachings such as evaluation and some classes.

### 3.3.2 Learning Analytics

The Learning Analytics (LA) can be defined as the measurement, collection, analysis, and reporting of data about learners and their contexts, for the purposes of understanding and optimize learning and the environments in which it occurs [34].

According to Larusson & White [35], learning analytics has the following use cases:

- How to enhance student and faculty performance;
- How to improve student understanding of the program's material;
- How to assess and attend to the needs of struggling learners;
- How to improve accuracy on grading;
- How to allow instructors to assess and develop their own strengths;
- How to encourage the more efficient use of resources at the institutional level.

This area works around answering hypothesis. It tests if each hypothesis proposed by the ones interested (i.e., instructors, direction, students) is valid and, in either case, presents the information supporting the conclusion.

With this type of data, which is made available to a low number of people or that have only a small portion accessible, the biggest improvements have come from MOOC's (Massive Open Online Course) platforms, more precisely from adaptive MOOC's. One example comes from a paper [36] which proposes a framework for MOOC's platforms that includes the use of software to evaluate the efficiency of these platforms. Another example of the application of LA uses data from Coursera, one of the biggest MOOC's platforms. It proposes the analysis of the students' data to gain insight into the learning process by applying mining techniques. In the end, the authors find that successful students watch the program's material in the proper way without skipping and progress in batches while the opposite is true for unsuccessful ones. It additionally found a correlation between the viewing behavior of the program's material and the final grades. Their results were all supported by statistical techniques like Pearson's and Spearman's correlation coefficients [37].

# 4 Evaluation of solutions

After the presentation of the main frameworks, solutions, and techniques that are considered as the state of art, it's important to evaluate each one, compare them and choose the best to achieve the intended solution. This chapter is divided into three parts. The first compares and chooses the best frameworks and language to extract knowledge from the data, the second compares the BI tools to choose the one that is more useful to create dashboards for the presentation and analysis of the data and, finally, the third presents the best options to deploy this work making the case for either a cloud option or an on-premises one.

## 4.1 BI Tools

Before this thesis work started, the author had a preference for Power BI due to some pre-existent successful experiences. Nonetheless, a comparison between the BI tools is needed to decide if it is the best tool for this thesis. To gather insights from the data there are nowadays many tools available. The website "software advice" made a concise list of the top 200 and in Figure 16 we can see a comparison between some of them, separated by quadrants.

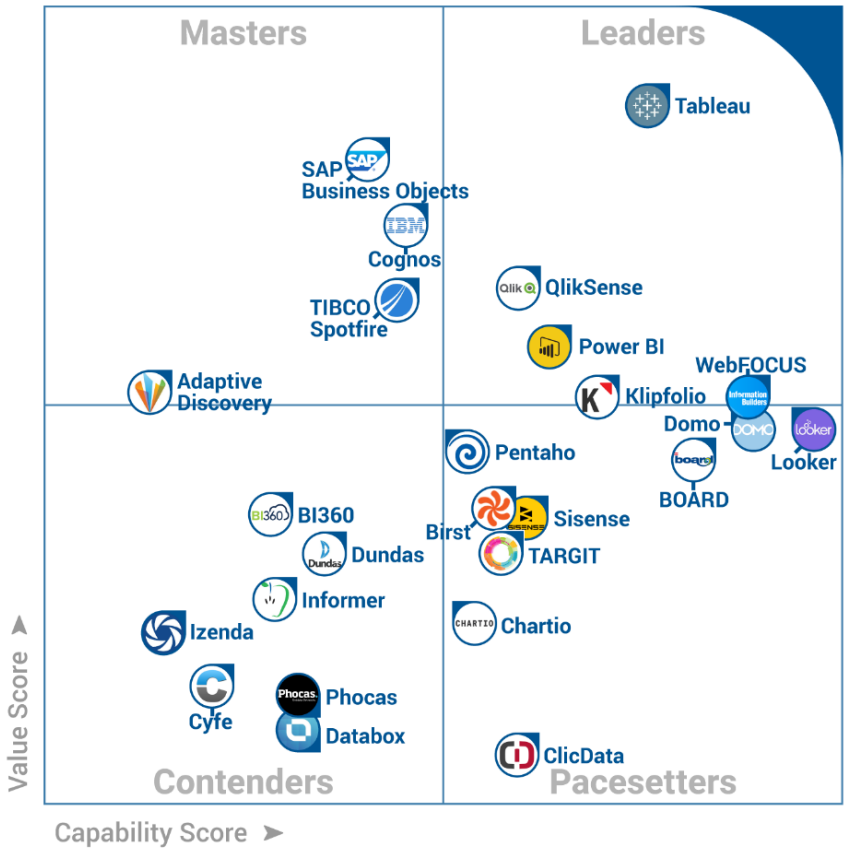



Figure 16 - Gartner's comparison of BI tools [38]

The quadrants are:

- Contenders: have a specialized set of features at a high point price;
- Pacesetters: have multiple great features but have a high price;
- Masters: focus heavily on key features;
- Leaders: all around strong products.

All the tools in display had at least a 3.0 score (out of a maximum of 10) by users of the software advice so that they can be considered as useful for at least some companies. As this work doesn't have any critical specifies necessary we will concentrate on Tableau and Power BI.
According to a paper by Grant Duncan [39], the categories to look out when choosing a BI tool are: 1) Overall capabilities and alignment to business goals; 2) Data management capabilities; 3) Capabilities for building dashboards; 4) Visuals; 5) Formatting & Functionality; 6) Capabilities for consuming dashboards; 7) Advanced analytics; 8) Technical architecture; 9) Exporting and integrations; 10) Mobility; 11) Cost structure and; 12 Support and maintenance
From these, we present in Table 1 the main advantages of each system.

Table 1 - Advantages of tableau and Power BI

| Tableau advantages | Power BI advantages |
|---|---|
| Aesthetic Abilities | Drag and drop interface |
| Filter and Drill through to other dashboards /report page | Data Extraction, Transformation, and Load |
| Customizable parameters for what-if analysis by the user | Custom Heat Maps |
| Built-in map layers for demographic info | Detail Line and Multi-bar char |
| Custom shapes as visuals | Natural Query language |
| Unlimited point limitations on scatter plots | Pin visuals to custom dashboards |
| Customizable formatting | Cortana integration |
| Containerized dashboard layouts | Ability to create an Excel Pivot directly from the cloud-based data model |
| Floating legends | |
| Hyperlink on images | |
| Customizable mouse-over tooltips | |
| Storyboarding capability | |
| Built-in statistical capabilities | |
| Python and MATLAB model integration<br>User Community Forum Support | |

### 4.1.1 Conclusion

After the consideration of both Tableau and Power BI, both solutions are considered sufficient to the realization of the analysis of this work as both are able to connect to this work data sources (SQL database and excel file present transform the data to the required format and display the visuals. The Tableau offers better tools to format the dashboards, but the Power BI offers stronger support for the integration with Microsoft products. In the end, the choice was to use Power BI due to the author's previous knowledge using the platform and due to the history of ISEP using Microsoft products, which should make it easier for the application to be integrated with other services in a later iteration.

## 4.2 Data Mining Technologies

### 4.2.1 Data Mining Language

Both Python and R have a strong scientific community backing them up and developing an environment around these. The focus of this section will be on comparing the multiple features of both languages to choose the one most adjusted to the work in this thesis.

The evaluation of which language is the most used by data science communities can be a good metric to measure the easiness of developing with it and its community support. The more developers there are, the more probable it is that someone has already experienced the same problems we will encounter.

In terms of pure usage, both languages have seen a tremendous growth since the start of the machine learning popularization. Figure 17 shows that since 2013 both languages have experienced an increase in use, but Python is starting to distance itself.


Figure 17 - Google trends 2004 to February 2017 (Blue - Python, Red - R)

Similarly, Knuggets [40] has shown an increase in data science jobs that require Python and/or R. By using this metric, Python seems to have a bigger percentage of jobs than R.

In conclusion, although for many years, both languages have had similar success attracting developers. In recent years, Python has become the clear frontrunner in terms of usage.

However, R has wrappers to many of Python frameworks and has equally good (and sometimes better) packages for handling the data and statistics. In the end, R was selected due to the authors having previous knowledge with the language.

### 4.2.2 Data Mining Frameworks and Libraries

While Python has libraries as Scikit-Learn for general machine learning algorithms, R as multiple separate packages for each one.

As the dataset to be analyzed in the present thesis was small, the deep learning algorithms haven't been considered for the first iterations as they require substantial amounts of data to be accurate. In later iterations, if the number of data increases, frameworks like Keras could be considered for prototyping and others like TensorFlow could be used for later optimization.

With the R language as the primary language for this work, the extraction of knowledge used package tidyverse to prepare the data while other packages as Caret were applied to create the models.

### 4.2.3 Conclusion

In this section, with the consideration of R as the main language and with requirements as a clear representation of how the models reach a conclusion, many tools weren't considered, especially the deep learning frameworks and Python-specific libraries. However, the implemented work is exploratory which means that, although the main tools were chosen, the comparison of the best algorithms must be done in the implementation stage.

## 4.3 Deployment

### 4.3.1 Cloud

The three biggest cloud providers are AWS (Amazon), GCP (Google) and Azure (Microsoft) [41]. Each one of these, offer similar products and, with the option to use docker, the same product can be run on any one of these environments without making changes.

All of them are able to run R models and the big advantages of using the cloud instead of locally are:

- **RAM**: R is limited by the accessible RAM. While the local environment has limited capacity, the cloud is able to scale to the needs;
- **Big Data:** The network speed inside a cloud is much faster than using a common HTTP protocol, which is usually implemented on premises;
- **Services:** The cloud providers offer multiple services to facilitate the implementation and deployment of models like the ones used on this project;
- **Scalability:** With the passage of time the needs of a system usually grow and require an upgraded server on premises. The cloud providers are able to automatically or easily grow with the needs of the system (dataset storage, computation power).

In terms of security, as the R is an executable language, there's the risk of the server being compromised by a malware inserted into the code. However, there are alternatives like Microsoft R server which can be deployed to Azure that provides protection against unauthorized access and even encrypts the virtual machine where the models are deployed [42].

### 4.3.2 On-Premises

The deployment to a server on premises is useful when it's important to keep the data as close to the source as possible and the needs of the system are easily managed without the need for constant upgrades. This usually happens when the datasets aren't big enough to enter the domain of Big Data.

Two of the biggest solutions for the deployment of R models are RStudio Server and Machine Learning Server. Both support the development locally (RStudio and visual studio respectively) with the later deployment to their servers which are optimized to run with bigger datasets and use the resources that are available.

### 4.3.3 Conclusion

The deployment of the product to the cloud presents many advantages like the reduction of maintenance, high availability, and protection of the data. However, as one of the requirements for this project is to keep the data inside the ISEP premises, it was decided to keep the solution on premises.

Inside the on-premises solutions it's important to consider factors as 1) cost; 2) easiness to deploy; 3) scalability and; 4) facility to migrate to another provider.

The RStudio offers a free version but, if there's a need to scale to more than one server, it's mandatory to pay for each new server 9,999$/year.

The Machine Learning server is integrated with the SQL server 2017 at no additional costs. As ISEP has already access to this instance of SQL; its use doesn't bring any additional costs. Another advantage of using the Machine learning server is that it's built upon open source technologies that can easily be scaled or migrated to the cloud. In the end, the Microsoft alternative presents itself as a better alternative for the deployment of the model.

It's important to refer that, as it can be assumed that data in the cloud is as secure, if not more [42], as the data on premises, in the event that the requirement to keep the data on ISEP's server is no longer true, the cloud solution should be considered as the better option.

# 5 Design

With the evaluation of the best solutions and tools, the next step for the development of the data product is to define an architecture that is able to fulfill the requirements of the users which are to analyze the DEI's dataset and present the data in a way that improves the decision-making of the users and that is able to extract knowledge from the same data. To this end, the design chapter has been divided into Architecture (section 5.1), Engineering requirements where the use cases are listed (section 5.2) and Non-functional Requirement (section 5.3).

## 5.1 Architecture

To present the information of the datasets in a timely manner and enable the user to not only to visualize the predictions but also recompute them, this work was divided into three modules as can be seen in Figure 18. On the server side, there will be a Power BI server, a Rest API and a database while on the client side there will be only the browser to access the Power BI server.

The Power BI instance is responsible for 1) presenting the analysis of the data; 2) present the model prediction and 3) interact with these models through the API. This component was built so that the information is refreshed in a scheduled way. In this way, all the process time is done as preemptively as possible reducing the time needed to load the dashboard.

The Rest API, built in R, will be responsible for creating the models, hosting them and updating them depending on the required frequency (i.e., after each semester) or if requested by the user. Finally, the database built-in SQL Server will host the outputs of the REST API such as model performances and student predictions. In this work, all the data related to the students' grades were stored in a single Excel File.

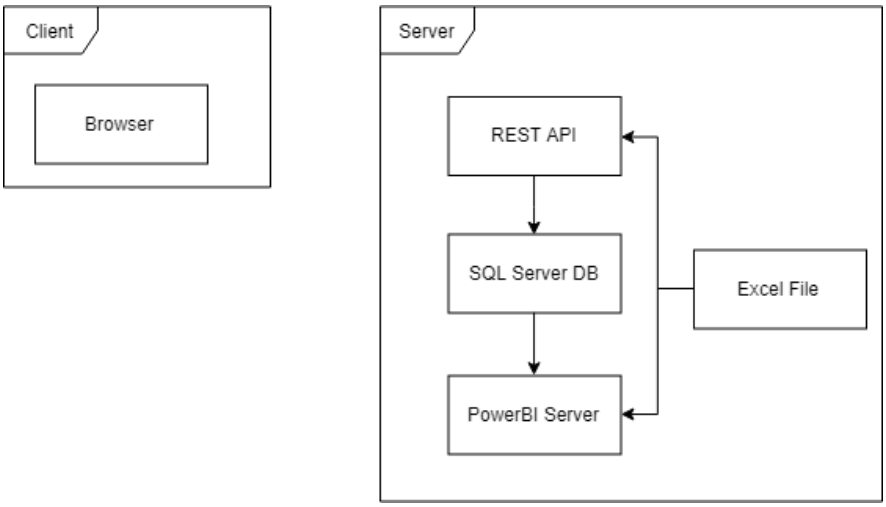



Figure 18 - Architecture

## 5.2 Engineering requirements

This section presents the main use cases of this work. As can be seen in Figure 19, these are: 1) Data Analysis; 2) Prediction of student's subject failure and; 3) Recompute the models.

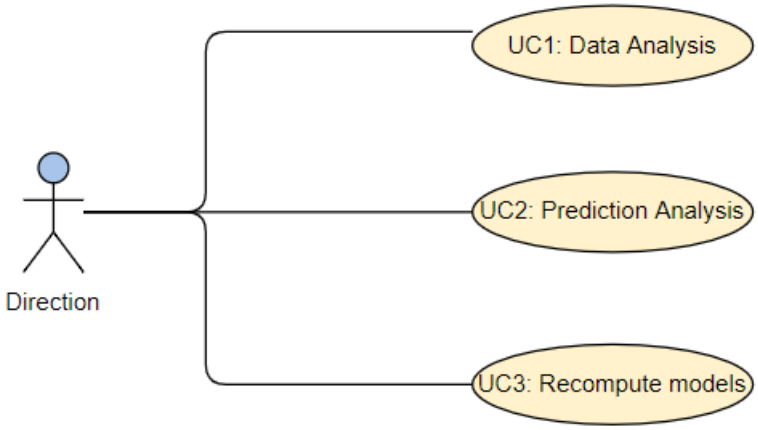



Figure 19 - Use cases

### 5.2.1 UC1: Data analysis

The data analysis is done mainly through plots (i.e., bar plots, scatter plots, etc.). The data portal will organize this by domain of interest. The user should be able to choose the one it's interested in, and the system should present it immediately. The system should conclude with success when the user closes the dashboard.

**Main Actors**

DEI's direction

**Interested parts**

DEI's direction

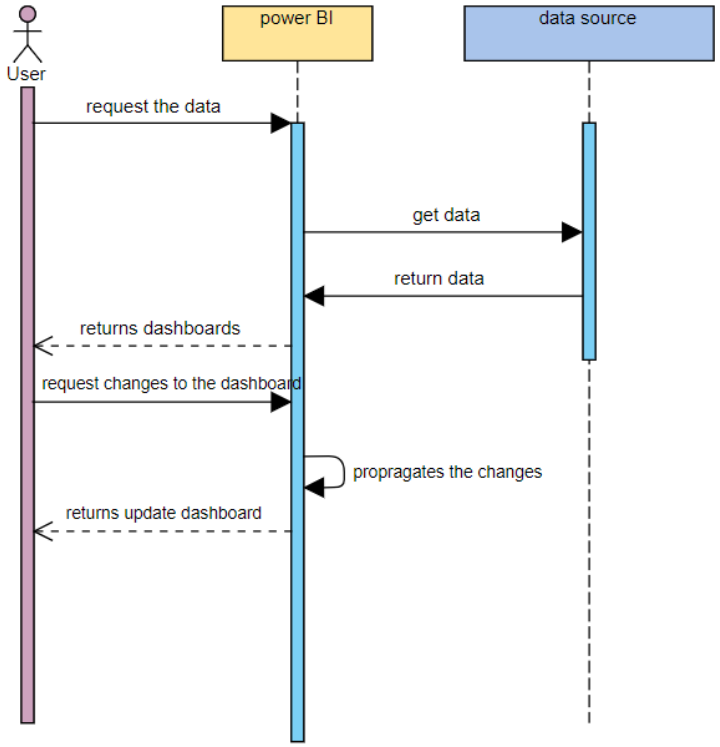
**Pre-condition**

-

**Main Scenario**

1. The user requests the data;
2. The system presents the requested data on multiple formats (i.e., pivot tables, plots, standard deviation, etc.);
3. The user iterates with the dashboard;
4. The system propagates the data throughout the dashboard;
5. The user closes the dashboard;
6. The system closes with success.

**Complete Path**




**Special Requisites**

- Presentation of the data through dashboards and reports.

**Technology and variation of data**

-

**Occurrence frequency**

-

**Open questions**

- Must there be a formal way to request a new analysis of the data?

### 5.2.2 UC2: Prediction of students' subject failure

The user must choose which students it wants to predict if are at risk of failing a subject. The system runs the model with the given students and presents the results with its accuracy. After visualizing the results, the system terminates with success.

**Main Actors**

DEI's direction

**Interested parts**

DEI's direction

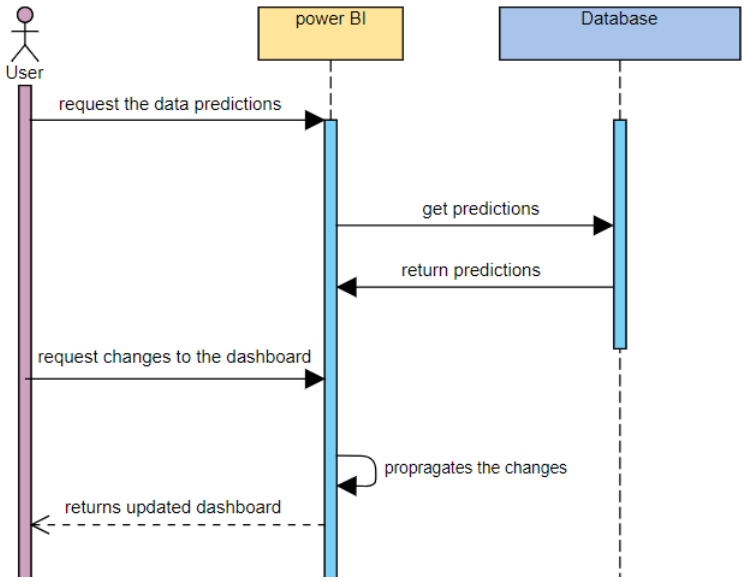
**Pre-condition**

-

**Main Scenario**

1. The user requests the student predictions;
2. The system presents the requested data on multiple formats;
3. The user iterates with the dashboard;
4. The system propagates the data throughout the dashboard;
5. The user closes the dashboard;
6. The system closes with success.

**Complete Path**



**Special Requisites**

-

**Technology and variation of data**

-

**Occurrence frequency**

-

**Open questions**

- What are the best features to identify a student at risk of failing a subject?
- Should there be a way to filter through the identified students? Or an explanation of why they are at risk of failing?

### 5.2.3 UC3: Recompute the models

The user must choose which years it considers important to generate the model. After confirming the inputs, the system runs the model with the given students and saves the new models. In the end, it returns a message and terminates with success.

**Main Actors**

DEI's direction

**Interested parts**

DEI's direction

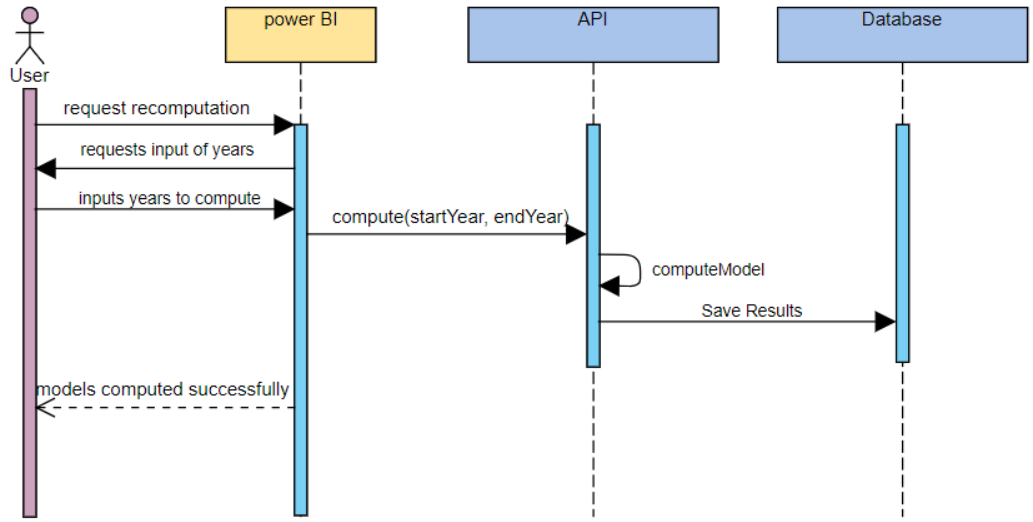
**Pre-condition**

-

**Main Scenario**

1. The system request input;
2. The user selects the student to predict;
3. The system returns the result with respective accuracy;
4. The system terminates with success.

**Complete Path**




**Special Requisites**

-

**Technology and variation of data**

-

**Occurrence frequency**

-

**Open questions**

- What are the best features to identify a student at risk of failing a subject?
- Should there be a way to filter through the identified students? Or an explanation of why they are at risk of failing?

## 5.3 Non-functional Requirement

### 5.3.1 Functionality

- Show usable statistics;
- Easy addition of new statistical metrics in the LEI's context;
- Traceability of the data origin;
- High accuracy (>60%).

### 5.3.2 Usability

- The statistics (i.e., graphs, tables, etc.) must be easily readable to facilitate its comprehension;
- The predictions must be clear, direct and concise;
- The predictions must always present how confident the models are.

### 5.3.3 Reliability

- High availability of the portal (>99%).

### 5.3.4 Performance

- The data mining models' build time doesn't need to be fast as they must be built sporadically (once or twice a year).

### 5.3.5 Supportability

- The solution must be clearly divided into the frontend, statistics, and data mining so that they aren't dependent on one another;
- The portal, statistics, and prediction must be in Portuguese;
- The development of the solution will be in English;

### 5.3.6 +

- As the information is private all the systems must run on-premises (ISEP's server).

# 6 Implementation of the solution

Following the project architecture definition, it is now presented the development of this solution which includes the software (dashboard and API) and the models to predict the students' behavior from their historical data. This includes: 1) the definition of the data source model (6.1); 2) the development of the models (6.2), including the API to automatically recompute them (6.2.3) and; 3) development of the dashboards to present in an interactive and user friendly way the data and the predictions generated (6.3).

## 6.1 Data Analysis

### 6.1.1 Data Source

This data is divided into 9 columns: Subject, Season, Student Id, Subject Grade, Date of Grade, Subject Exam Grade, Date of Exam, Subject's Final Result and Date of Final Result. The reason for this format is as follows.

Each year the student has two semesters with 5 subjects. For each one of the subjects, the method applied is to give theoretical and practical classes and do some evaluations during this period. The average of these grades, using a unique formula for each subject, is the frequency grade which is the first given to the student. The following phase is the exam given at a later date. The result of the exam and the frequency combined to return the final result of the student. There are other cases like the student doing an exam which substitutes the frequency grade, but these will be explained in more detail in section 6.1.1.1.

The analyzed dataset (Figure 20) includes grades starting in 2014/2015 until 2016/2017 with a total of 1974 students and 34 subjects. The present work was built around a data file in a CSV format.

| 1 | Disciplina | Época | Num | Freq. | Data Freq. | Exame | Data Exame | Resultado | Data Resultado |
|---|---|---|---|---|---|---|---|---|---|
| 2 | BDDAD | NM | 127289 | NF | 1/24/2017 | | | NF | 1/24/2017 |
| 3 | ARQSI | NM | 127289 | NF | 1/24/2017 | | | NF | 2/1/2017 |
| 4 | BDDAD | NM | 718720 | 13 | 1/24/2017 | 6.4 | 1/24/2017 | SM | 1/24/2017 |
| 5 | ALGAV | NM | 989462 | 12.3 | 1/3/2017 | 2.5 | 1/28/2017 | 8 | 1/28/2017 |
| 6 | ALGAV | NM | 963488 | | | 2.9 | 1/28/2017 | SM | 1/28/2017 |
| 7 | RCOMP | NM | 973478 | 19 | 6/23/2017 | 15 | 7/4/2017 | 17 | 7/4/2017 |
| 8 | RCOMP | NM | 959492 | 19.5 | 6/23/2017 | 12.9 | 7/4/2017 | 16 | 7/4/2017 |
| 9 | ALGAV | NM | 967484 | 12.1 | 1/3/2017 | 1 | 1/28/2017 | 8 | 1/28/2017 |
| 10 | ARQSI | NM | 267569 | SM | 1/24/2017 | | | SM | 2/1/2017 |
| 11 | ARQSI | NM | 326687 | 13.6 | 1/24/2017 | 10.3 | 2/1/2017 | 12 | 2/1/2017 |
| 12 | ANADI | NM | 326687 | 13 | 6/23/2017 | | | 13 | 6/23/2017 |

Figure 20 - Sample of the data

#### 6.1.1.1 Seasons

There are 7 evaluation seasons in which a student can pass a subject. The case explained in the previous sections, where the student passes the exam by doing the frequency and the final exam is the “normal” season (NM). However, in case the student fails or wants to improve his grades, he has the option of doing a second exam (RE). The other seasons are variations of this, depending on the status of the student. In Table 2 the existing phases are presented:

Table 2 - Phases

| Phases | Description |
|---|---|
| **C2** | Special phase for working students |
| **DZ** | Special phase for finalists (students with less than 4 subjects to finish the program) |
| **EE** | Special season |
| **ML** | Second season. This season is equal to RE, but the student has already passed the subject and is looking into improving his grade |
| **NM** | Normal season |
| **PO** | Oral season |
| **RE** | Second season |

#### 6.1.1.2 Grades

For a student to pass a subject he/she must have a final grade equal to or above 10 (in a scale of 0-20). Each subject has its own formula where some have frequency and/or exams with a minimum grade to pass (i.e., a student with 20 at frequency will fail with 0 in an exam even if the final result is above 10).
While there is only one way to pass, there are multiple forms of failing a class. In Table 3 the possible results are presented with a mapping to the old system used.

After an analysis of the data, some cases were detected where the student had as final grade NC, NF or SM. These grades don't make sense in the context of the final grade and a possible explanation is an error during the input of data.

Table 3 - Mapping of the old phase system to the new

| **Type** | **Old system** | **New System** | **Description** |
|---|---|---|---|
| *Frequency* | SMNF, SMS, SMR | SM | Without minimum required to pass |
| | NF | NF | No attendance |
| | NC | NC | Not classified |
| | 0-20 | 0-20 | Frequency grade. It must be higher than the minimum required |
| *Exam* | DT | FT | Missed the exam |
| | AN | AN | Annulled due to fraud |
| | 0-20 | 0-20 | Exam grade. It must be higher than the minimum required |
| *Final Grade* | AN | AN | Annulled due to fraud |
| | 0-20 | 0-20 | Exam grade. It must be higher than the minimum required |

#### 6.1.1.3 Subjects

The LEI's program is currently divided into 3 years, each with two semesters (Table 4). For each semester there are 5 subjects. For the present work, they were manually introduced into the system to allow some analysis, as to ask if a student has already graduated.

Table 4 - Subjects

| Year | Semester | Subject |
|---|---|---|
| **1** | 1 | LAPR1 |
| | | AMATA |
| | | ALGAN |
| | | PRCMP |
| | | APROG |
| | 2 | PPROG |
| | | LAPR2 |
| | | MDISC |
| | | ESOFT |
| | | MATCP |
| **2** | 1 | ARQCP |
| | | BDDAD |
| | | ESINF |
| | | FSIAP |
| | | LAPR3 |
| | | P/EST-ERASMUS 10 |
| | 2 | EAPLI |
| | | LAPR4 |
| | | LPROG |
| | | RCOMP |
| | | SCOMP |
| | | P/EST-ERASMUS 20 |
| **3** | 1 | IARTI |
| | | COMPA |
| | | ASIST |
| | | ALGAV |
| | | ARQSI |
| | | GESTA |
| | | LAPR5 |
| | | SGRAI |
| | 2 | CORGA |
| | | INFOR |
| | | ANADI |
| | | PESTI |

### 6.1.2 Data Model

The data model is closely related to the data to be analyzed. The model presented in Figure 21 was focused on the analysis of the students' behavior and was based on the dataset to be used later in the experimentation phase. The identified entities were 1) Student (Aluno); 2) Subjects (Unidades Curriculares); and 3) Grades (Notas). The student is identified by a unique number. Each student can have multiple registrations and has multiple grades at the end of each semester. The other entities are all related to the model created to predict the students' grades. These are: 1) Predictions (Previsões); 2) Model Performance (Performance Modelos); 3) Models (Modelos); and 4) Main Paths (Caminhos Frequentes).

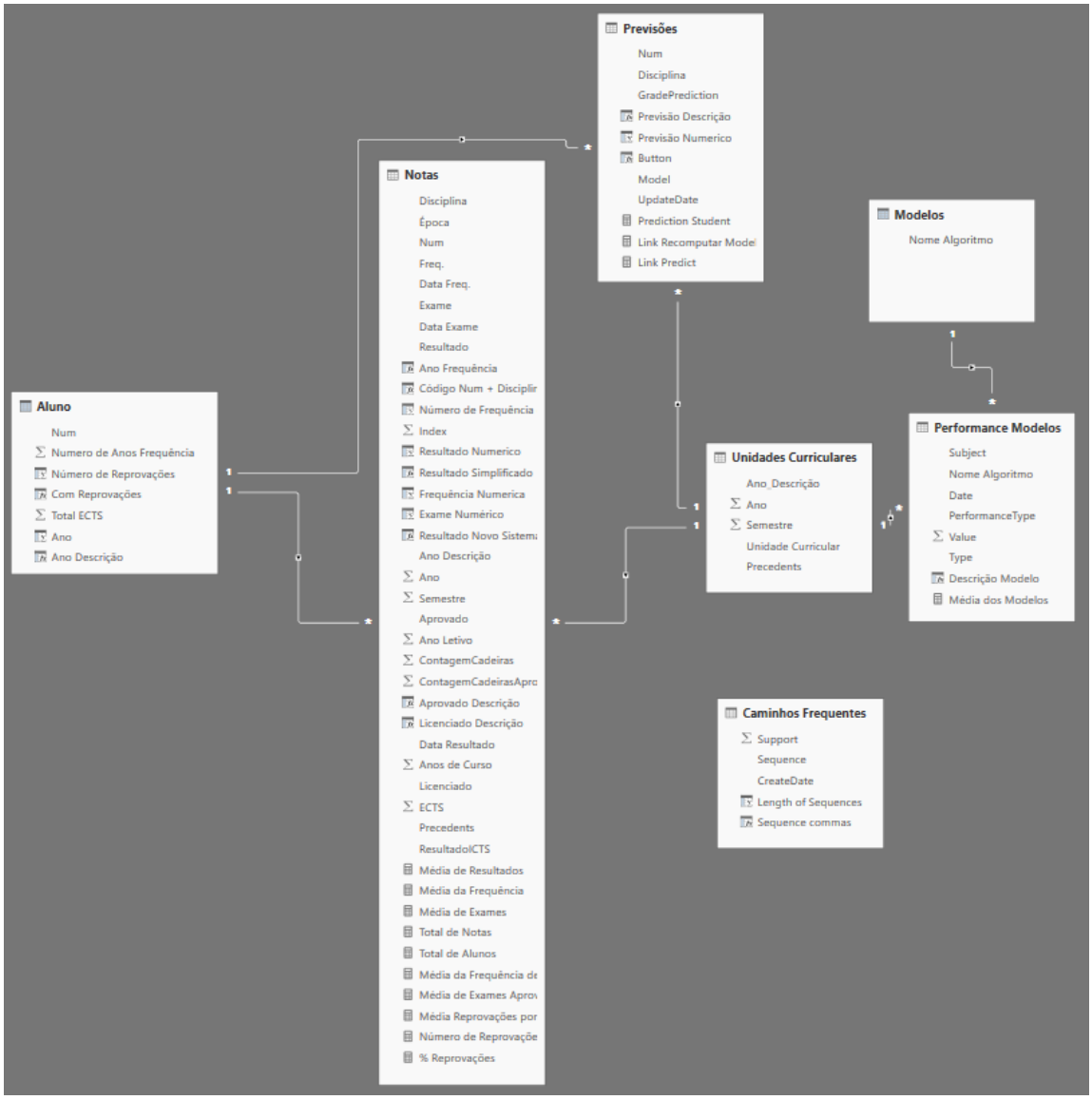



Figure 21 - Data Model

## 6.2 Prediction Models

In the present work, we aimed at developing two different prediction models. The first one aimed at predicting, from historical data of each student the likelihood of succeeding in each subject (6.2.1). The second model aimed at deriving the most common paths of the student during the program (6.2.2).

In order to enable the user to redefine the models with the evolution of the available data, an API was developed to automate the process and is described in section 6.2.3.

### 6.2.1 Subjects' outcome

The main idea of this section is to create a model capable of predicting if a student will get a positive grade on a specific subject. As the dataset is formed by the grades of each student from the period of 2013 to 2016 the focus of the constructed model is in using previous grades to predict the outcome.

#### 6.2.1.1 Models with no transformation of data

As explained in section 4.2.3, the models were developed with the R programming language. For the development of the models, a seed was set to remove randomization from the analysis (Figure 22).

```
set.seed(123)
```

Figure 22 - Set seed function

The next step was to get the grades data from the excel files presented in section 6.1.1. As the data was divided into a sheet for each year, the map_df was used to iterate through each sheet and bind to a single data.frame. Finally, a file with all the subjects of LEI was imported (Figure 23).

```
filePath <- '../Data/Notas_CGF_Anonimizadas.xlsx'

grades_sheets <- excel_sheets(filePath)

grades <- map_df(grades_sheets, ~ read_excel(filePath, sheet = .x))

disciplinas <- read_xlsx('../Data/Disciplinas.xlsx')
```

Figure 23 - Extraction of the dataset

Afterward, some transformations were applied to the students' grades to generate columns like the binary outcome of the student (column "Aprovado") which were already explained in

section 6.1.1. Another transformation that was implemented was of the grades to the ICTS system. This European system divides the notes in letters ranging from F to A+ where F is the only grade that implicates a student failed, as can be seen in Figure 24. With this conversion to categorical values, we hope to improve the models generated.

```
ResultadoICTS = case_when(
  is.na(as.numeric(Resultado)) ~ "F",
  as.numeric(Resultado) < 10 ~ "F",
  as.numeric(Resultado) < 14 ~ "C",
  as.numeric(Resultado) < 16 ~ "B",
  as.numeric(Resultado) < 18 ~ "A",
  as.numeric(Resultado) <= 20 ~ "A+"
```

Figure 24 - Conversion of numerical grades to the ECTS system

To train and test the models, the holdout method was applied with the help of caret's createDataPartition. This method consists of holding some of the data from using in the training phase so that it's possible to test the models with data it hasn't seen and, as such, facilitates the performance of the model with new data (the main objective of developing the predictive models). In this case, the data was divided into 2/3 for the training dataset and 1/3 for the testing. In Figure 25 the outcome column is referenced for the division of the dataset.

```
  in_train <- caret::createDataPartition(studentGrades$Aprovado, p = 2/3, list = FALSE)

set_train <- studentGrades[ in_train,]
set_test  <- studentGrades[-in_train,]
```

Figure 25 - Holdout function

In this work the models used were the Decision Tree, Random Forest, and Naive Bayes. All of these are supervised classification algorithms which need a label for them to be able to classify. In this case, the variable used was the "Aprovado" column which validates if the student passed a class or not. In Figure 26 it's presented the caret's decision tree function which applies the formula "Aprovado ~ Disciplina + Época + ResultadoICTS" b". The same formulas were applied to the other two models. After the creation of the model, the predict function was applied to try to predict the students' outcomes in the test dataset.

```
DTModel <- rpart( AprovadoPPROG ~ LAPR1 + AMATA + ALGAN + PRCMP + APROG, data = set_train, method =
"class")

rpart.plot(DTModel, type = 3, box.palette = c("red", "green"), fallen.leaves = TRUE)


set_test$DTpred <- predict(DTModel, select(set_test, -Aprovado), type = "class", probability = TRUE)
```

Figure 26 - Caret's Decision Tree function

In a first iteration, the models were created with the data in its current format (subject - year - grade). However, the models predicted (Figure 27) would easily overfit without having acquired knowledge (if the grade was negative the student would pass).

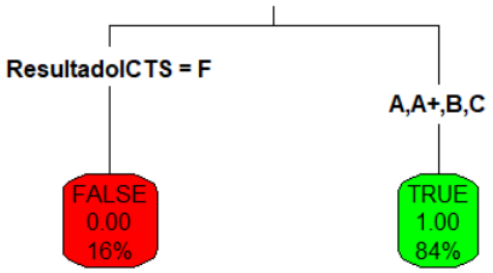



Figure 27 - Decision Tree model without transformation of the data

The evaluation of the models was presented as the mean of cases predicted correctly, accuracy and area under the curve (section 3.2.2). Figure 28 shows the roc curve of the models where the prediction is perfect. But, as we have seen, this is due to overfitting.

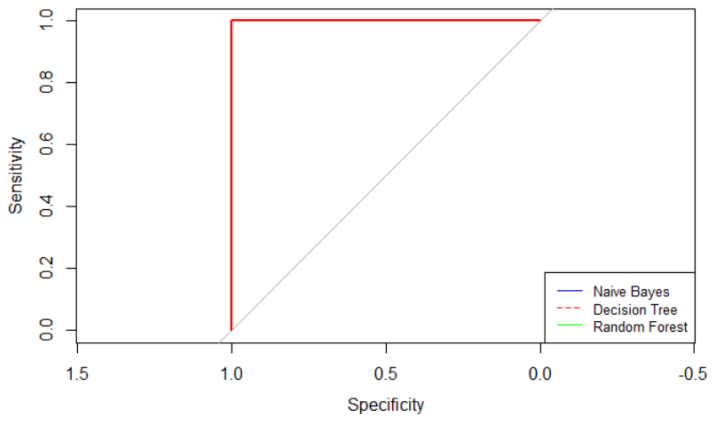



Figure 28 - ROC curve of the models

### 6.2.1.2 Models with the transformation of data

As the model generated using the data in its original format weren't very useful, we tested another alternative by changing its format to one that better represents the problem we are trying to solve (predict subjects' outcome for each student). As such, instead of creating a model to predict the outcome in all subjects, we opted for creating a model for each subject. To this end, the grades of each student were aggregated one row per student (Figure 29).

| | ALGAN | ALGAV | AMATA | ANADI | APROG | ARQCP | ARQSI | ASIST | BDDAD | COMPA | CORGA | EAPLI |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | Por fazer | Por fazer | Por fazer | Por fazer | Por fazer | Por fazer | Por fazer | Por fazer | Por fazer | Por fazer | Por fazer | Por fazer |
| 2 | Por fazer | Por fazer | Por fazer | Por fazer | Por fazer | F | Por fazer | Por fazer | Por fazer | Por fazer | Por fazer | C |
| 3 | Por fazer | F | Por fazer | Por fazer | Por fazer | Por fazer | Por fazer | Por fazer | Por fazer | Por fazer | Por fazer | Por fazer |
| 4 | Por fazer | B | Por fazer | A+ | B | A | A | A+ | B | Por fazer | A+ | B |
| 5 | Por fazer | Por fazer | Por fazer | Por fazer | Por fazer | Por fazer | Por fazer | C | Por fazer | Por fazer | Por fazer | Por fazer |
| 6 | Por fazer | Por fazer | C | Por fazer | B | Por fazer | Por fazer | Por fazer | Por fazer | Por fazer | Por fazer | Por fazer |
| 7 | Por fazer | Por fazer | Por fazer | Por fazer | A | Por fazer | Por fazer | Por fazer | Por fazer | Por fazer | Por fazer | Por fazer |
| 8 | Por fazer | Por fazer | Por fazer | Por fazer | Por fazer | Por fazer | Por fazer | Por fazer | Por fazer | Por fazer | Por fazer | Por fazer |
| 9 | Por fazer | Por fazer | Por fazer | Por fazer | Por fazer | Por fazer | C | C | Por fazer | Por fazer | Por fazer | Por fazer |
| 10 | Por fazer | Por fazer | Por fazer | Por fazer | Por fazer | Por fazer | Por fazer | Por fazer | F | Por fazer | Por fazer | Por fazer |
| 11 | Por fazer | Por fazer | Por fazer | Por fazer | Por fazer | Por fazer | Por fazer | Por fazer | Por fazer | Por fazer | B | Por fazer |
| 12 | Por fazer | Por fazer | Por fazer | Por fazer | Por fazer | Por fazer | Por fazer | Por fazer | Por fazer | Por fazer | Por fazer | Por fazer |
| 13 | Por fazer | Por fazer | Por fazer | Por fazer | Por fazer | Por fazer | Por fazer | Por fazer | Por fazer | Por fazer | Por fazer | Por fazer |
| 14 | Por fazer | F | Por fazer | F | Por fazer | Por fazer | F | F | Por fazer | Por fazer | F | Por fazer |
| 15 | Por fazer | Por fazer | F | F | Por fazer | Por fazer | F | Por fazer | F | F | Por fazer | Por fazer |
| 16 | Por fazer | Por fazer | Por fazer | Por fazer | Por fazer | Por fazer | Por fazer | Por fazer | Por fazer | C | Por fazer | Por fazer |
| 17 | Por fazer | A | Por fazer | Por fazer | Por fazer | Por fazer | Por fazer | C | Por fazer | Por fazer | Por fazer | Por fazer |

Figure 29 - Matrix of students' data

To transform the data, a pipeline was created using R in conjunction with tidyverse packages. As can be seen in Figure 30, the grades were grouped by students' number and subject and only the last result in a subject was kept. In this way, all the historical data was deleted. Afterward, using the tidyr package, the subject column ("Disciplina") is spread, which means that each distinct value is separated into a new column. Finally, a column for each subject was created to indicate if the student passed ("F" and "Por fazer" indicate the student has failed).

```r
#' Transform all grades to a format of a row for each student with all of his grades
#' @param studentGrades Grades of the student
#' @param removeNum Remove the num column at the end
#' @return matrix with all the student's grades
getStudentGradesMatrix<- function(studentGrades, removeNum = TRUE) {

  studentGradesMatrix <- studentGrades %>%
    group_by(Num, Disciplina) %>%
    arrange( Disciplina, DataResultado, .by_group = TRUE) %>%
    summarize(
      ResultadoICTS = last(ResultadoICTS) # Get the last results of each students for the subject
    ) %>%
    group_by(Num, Disciplina) %>%
    mutate(id = row_number()) %>%
    select(-Num) %>%
    spread(Disciplina, ResultadoICTS, fill = "Por fazer") %>%
    select(-id) %>%
    ungroup(Num) %>%
    mutate(
      AprovadoALGAV = ifelse(ALGAV == "F" | ALGAV == "Por fazer",FALSE, TRUE),
```

Figure 30 - Pipeline used to get students' grades matrix

With the data in the desired format, a function name "getModelEvaluations" was created to accommodate the need of generating more than 30 models instead of one. This function returns the same type of evaluations generated on the first model. The models were all evaluated with the techniques explained in section 3.2.2. The diminutives represent the

following. NB is Naive Bayes, DT is Decision trees and RF is Random Forest. Table 5 presents the results of the function saved in a CSV file. The models were all evaluated with the techniques explained in section 3.2.2. The diminutives represent the following. NB is Naive Bayes, DT is Decision trees and RF is Random Forest.

Table 5 - Evaluation of models for each subject

| | mean | | | Accuracy | | | AUC | | |
|---|---|---|---|---|---|---|---|---|---|
| **Name** | NB | DT | RF | NB | DT | RF | NB | DT | RF |
| **ESOFT** | 91% | 100% | 100% | 91% | 100% | 100% | 91% | 100% | 100% |
| **LAPR2** | 92% | 100% | 100% | 92% | 100% | 100% | 92% | 100% | 100% |
| **MATCP** | 92% | 100% | 100% | 92% | 100% | 100% | 92% | 100% | 100% |
| **MDISC** | 95% | 100% | 100% | 95% | 100% | 100% | 95% | 100% | 100% |
| **PPROG** | 94% | 100% | 100% | 94% | 100% | 100% | 94% | 100% | 100% |
| **ARQCP** | 92% | 100% | 100% | 92% | 100% | 100% | 92% | 100% | 100% |
| **BDDAD** | 92% | 100% | 100% | 92% | 100% | 100% | 91% | 100% | 100% |
| **ESINF** | 93% | 100% | 100% | 93% | 100% | 100% | 93% | 100% | 100% |
| **FSIAP** | 87% | 100% | 100% | 87% | 100% | 100% | 86% | 100% | 100% |
| **LAPR3** | 94% | 100% | 100% | 94% | 100% | 100% | 94% | 100% | 100% |
| **EAPLI** | 95% | 100% | 100% | 95% | 100% | 100% | 95% | 100% | 100% |
| **LAPR4** | 96% | 100% | 100% | 96% | 100% | 100% | 96% | 100% | 100% |
| **LPROG** | 93% | 100% | 100% | 93% | 100% | 100% | 93% | 100% | 100% |
| **RCOMP** | 93% | 100% | 100% | 93% | 100% | 100% | 93% | 100% | 100% |
| **SCOMP** | 95% | 100% | 100% | 95% | 100% | 100% | 95% | 100% | 100% |
| **ALGAV** | 91% | 100% | 100% | 91% | 100% | 100% | 92% | 100% | 100% |
| **ARQSI** | 97% | 100% | 100% | 97% | 100% | 100% | 97% | 100% | 100% |
| **ASIST** | 96% | 100% | 100% | 96% | 100% | 100% | 96% | 100% | 99% |
| **COMPA** | 94% | 100% | 100% | 94% | 100% | 100% | 97% | 100% | 100% |
| **GESTA** | 96% | 100% | 100% | 96% | 100% | 100% | 96% | 100% | 100% |
| **IARTI** | 93% | 100% | 100% | 93% | 100% | 100% | 96% | 100% | 100% |
| **LAPR5** | 97% | 100% | 100% | 97% | 100% | 100% | 97% | 100% | 99% |
| **SGRAI** | 95% | 100% | 100% | 95% | 100% | 100% | 96% | 100% | 100% |
| **ANADI** | 97% | 100% | 100% | 97% | 100% | 100% | 97% | 100% | 100% |
| **CORGA** | 97% | 100% | 100% | 97% | 100% | 100% | 97% | 100% | 99% |
| **INFOR** | 96% | 100% | 100% | 96% | 100% | 100% | 95% | 100% | 99% |
| **PESTI** | 90% | 100% | 100% | 90% | 100% | 100% | 90% | 100% | 100% |

Although the random forest and the naive Bayes presented more realistic values, the decision tree continues overfitting which might indicate a problem in the data. By looking into one of the decisions trees generated (see Figure 31) it's clear that the model is giving preference to subjects of the same semester or future semesters.

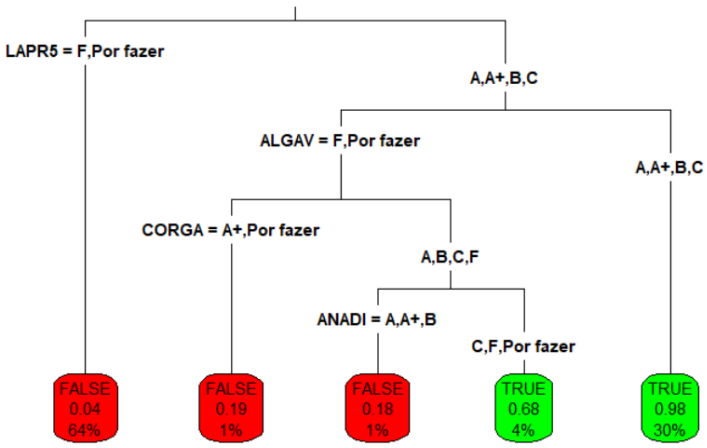



Figure 31 - SGRAI's decision tree

#### 6.2.1.3 Model with precedents

Although the models for each subject are an improvement when compared to a more general model, we noticed that they were too dependent on grades of the same semester or future subjects. This is a good case for repeating students but, as in most cases the student is having the subject for the first time, the model can be a bit lacking. To take this into account, the concept of precedents was introduced, which is a list of all subjects taken on previous semesters. A separate file (Figure 32) was created with the information regarding each subject and its precedents.

| Ano Descrição | Ano | Semest | Disciplina | Precedents |
|---|---|---|---|---|
| 1º Ano | 1 | 1 | ALGAN | |
| 1º Ano | 1 | 1 | AMATA | |
| 1º Ano | 1 | 1 | APROG | |
| 1º Ano | 1 | 1 | LAPR1 | |
| 1º Ano | 1 | 1 | P/EST-ERASMUS 10 | |
| 1º Ano | 1 | 1 | PRCMP | |
| 1º Ano | 1 | 2 | ESOFT | LAPR1, AMATA, ALGAN, PRCMP, APROG |
| 1º Ano | 1 | 2 | LAPR2 | LAPR1, AMATA, ALGAN, PRCMP, APROG |
| 1º Ano | 1 | 2 | MATCP | LAPR1, AMATA, ALGAN, PRCMP, APROG |
| 1º Ano | 1 | 2 | MDISC | LAPR1, AMATA, ALGAN, PRCMP, APROG |
| 1º Ano | 1 | 2 | PPROG | LAPR1, AMATA, ALGAN, PRCMP, APROG |
| 2º Ano | 2 | 1 | ARQCP | LAPR1, AMATA, ALGAN, PRCMP, APROG, ESOFT, LAPR2, MATCP, MDISC, PPROG |

Figure 32 - Subjects' CSV file

The subjects' information is merged with the grades' dataset adding the subjects' year, semester and precedents with the help of dplyr package.

In the previous model, all columns in the students' matrix were used but to implement the concept of precedents, a parameter was added to the function. An example of this is PPROG, in which R formula was previously "AprovadoPPROG ~." and now is "AprovadoPPROG ~ LAPR1 + AMATA +ALGAN + PRCMP + APROG". In Table 6 we can see the results of the table evaluations.

Table 6 - Evaluation of models for each subject with precedents

| | mean | | | Accuracy | | | AUC | | |
|---|---|---|---|---|---|---|---|---|---|
| Name | NB | DT | RF | NB | DT | RF | NB | DT | RF |
| ESOFT | 86% | 86% | 86% | 86% | 86% | 86% | 86% | 85% | 86% |
| LAPR2 | 89% | 90% | 89% | 89% | 90% | 89% | 89% | 90% | 88% |
| MATCP | 87% | 87% | 87% | 87% | 87% | 87% | 87% | 87% | 86% |
| MDISC | 90% | 91% | 90% | 90% | 91% | 90% | 89% | 91% | 89% |
| PPROG | 88% | 88% | 89% | 88% | 88% | 89% | 88% | 88% | 88% |
| ARQCP | 68% | 69% | 73% | 68% | 69% | 73% | 66% | 66% | 71% |
| BDDAD | 68% | 74% | 76% | 68% | 74% | 76% | 66% | 70% | 73% |
| ESINF | 69% | 69% | 72% | 69% | 69% | 72% | 65% | 60% | 68% |
| FSIAP | 66% | 73% | 75% | 66% | 73% | 75% | 61% | 64% | 66% |
| LAPR3 | 72% | 70% | 74% | 72% | 70% | 74% | 70% | 65% | 71% |
| EAPLI | 93% | 93% | 94% | 93% | 93% | 94% | 93% | 93% | 93% |
| LAPR4 | 92% | 91% | 93% | 92% | 91% | 93% | 92% | 91% | 93% |
| LPROG | 93% | 91% | 92% | 93% | 91% | 92% | 93% | 90% | 91% |
| RCOMP | 93% | 90% | 93% | 93% | 90% | 93% | 92% | 90% | 92% |
| SCOMP | 93% | 93% | 94% | 93% | 93% | 94% | 92% | 93% | 93% |
| ALGAV | 80% | 82% | 84% | 80% | 82% | 84% | 84% | 75% | 83% |
| ARQSI | 79% | 82% | 82% | 79% | 82% | 82% | 82% | 78% | 83% |
| ASIST | 77% | 82% | 83% | 77% | 82% | 83% | 77% | 77% | 79% |
| GESTA | 77% | 82% | 84% | 77% | 82% | 84% | 77% | 78% | 80% |
| LAPR5 | 77% | 80% | 82% | 77% | 80% | 82% | 80% | 76% | 78% |
| SGRAI | 77% | 81% | 82% | 77% | 81% | 82% | 81% | 77% | 82% |
| ANADI | 96% | 96% | 97% | 96% | 96% | 97% | 95% | 94% | 95% |
| CORGA | 95% | 94% | 96% | 95% | 94% | 96% | 95% | 92% | 95% |
| INFOR | 93% | 96% | 97% | 93% | 96% | 97% | 93% | 95% | 96% |
| PESTI | 90% | 88% | 91% | 90% | 88% | 91% | 90% | 86% | 90% |

The current model, although with inferior accuracy and area under the curve, can generate models that take into account only previous subjects. In Figure 33 we can see the model with an area under the curve of 88% for each model.

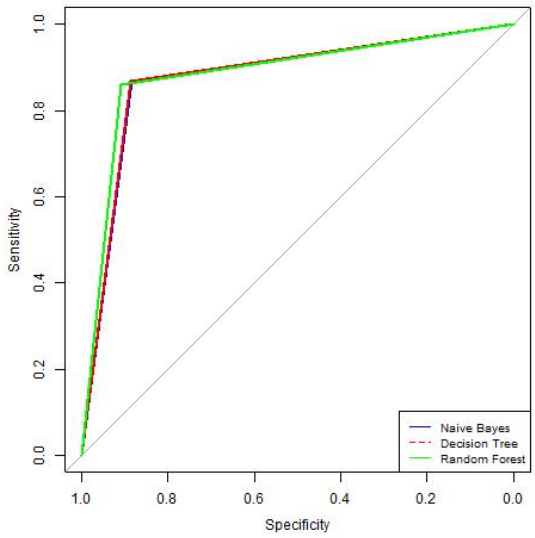



Figure 33 - ROC curve of PPROG

In Figure 34, we can detect that students that didn't complete LAPR1 have a higher risk of not passing PPROG. The visuals resulting from the other models are presented on the

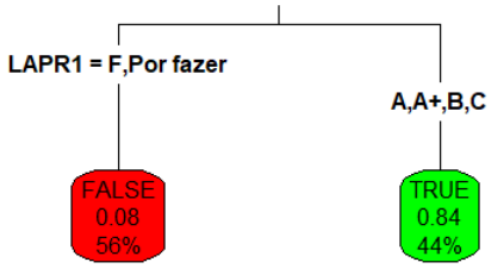



Figure 34 - PPROG's Decision tree

#### 6.2.1.4 Comparison of models

This work has developed three iterations. In the first one, the original structure of data was used. In the second iteration the subjects aggregated to each student and, in the third one, for each subject, the respective model considered only the previous subjects. In Table 7 the mean, accuracy, and area under the curve of each iteration were presented. As the second and third iteration had multiple models, only the average of each metric was considered.

Table 7 - Evaluation of each model

| **Type** | **mean** | | | **accuracy** | | | **area under the curve** | | |
|---|---|---|---|---|---|---|---|---|---|
| | **NB** | **DT** | **RF** | **NB** | **DT** | **RF** | **NB** | **DT** | **RF** |
| Original Data | 100% | 100% | | 100% | 100% | | 100% | 100% | |
| No Precedents | 94% | 100% | 100% | 94% | 100% | 100% | 94% | 100% | 100% |
| Precedents | 83% | 85% | 86% | 83% | 85% | 86% | 83% | 80% | 82% |

a) Shown values are average of all models derived using each type of data

The results show that the first and second iteration reached a consistently of 100% which indicates that the models overfitted. In the first case, this was due to the structure of the data as the algorithm detected that, in an ECTS scale, the student would always fail if it had a score of “F”.

In the second iteration, although the subject grade was removed to avoid the previous problem, the models overfitted again, as it always deduced that if the student hadn’t done the other subjects in the same semesters that he also wouldn’t have done the subject in question. Although useful in cases where the student is repeating a chair which he has already failed, the models weren’t as good in detecting the students’ outcome when doing a subject for the first time.

Finally, in the third iteration, the model’s performance worsened but kept a score above 80%. The models created were able to correctly predict based on previous subjects which were the main purpose of this work, although it proved better at subjects in the second semesters than in the first semester. Another limitation of this model, as it needs previous subjects is that it is unable to predict for the subjects in the first semester of the first year.

### 6.2.2 Students' Main Path

To generate the most common paths taken by the students, the cspade algorithm was applied (section 3.2.1.3) to the students' grades dataset (section 6.1.1). The implementation of the solution was done in a Rmarkdown file named "Main Path.Rmd" and was divided into 3 sections: 1) Conversion of the grades to a transaction format; 2) Computation of the cspade algorithm and; 3) Conversion of the output to a CSV.

In the first step the grades should be a data frame with the following columns: 1) Subject; 2) Approved; 3) Students' Number; 4) Year and; 5) Semester. The output returned after running the convertToTransactionFormat as seen in Figure 35 should be in the following structure: 1) Num; 2) EventID; 3) Size and; 4) Sequence.

```
studentPaths <- convertToTransactionFormat2(grades = studentGrades)

# Exports the result to a file to be read later by the cspade algorithm
write.table(studentPaths,"../Data/grades_ARules.txt",sep="\t",row.names=FALSE, col.names = FALSE,
quote = FALSE)
```

Figure 35 - Conversion of the grades to transactions

In Figure 36 it's presented the function responsible for the conversion. It's divided into multiples steps using the purrr package. The transformations are the following: 1) grouping of the grades by the student for each year and subject; 2) the results are merged by the highest grade of each student in each subject; 3) addition of an index column called EventID which represents the order of the sequences.

At the end of the function, a data frame is returned with the required columns to run the cspade algorithm (Num, EventID, Size, and Sequence).

```r
#' Function to convert the initial data to a basket format
#' @param  grades Student's grades
#' @return Student's grades in a transaction format
convertToTransactionFormat2 <- function(studentGrades) {

  studentPaths <- studentGrades %>%
    group_by(Num, AnoFreq, Disciplina) %>%
    summarise(
      Disciplina2 = last(Disciplina),
      ResultadoICTS = min(ResultadoICTS)
    ) %>%
    tidyr::unite(DisciplinaResultado, Disciplina2, ResultadoICTS) %>%
    ungroup() %>%
    group_by(Num, AnoFreq) %>%
    mutate(
      SequenceID = 1,
      Size = n_distinct(Disciplina),
      Disciplinas = paste0(DisciplinaResultado, collapse = "\t")
    ) %>%
    summarize(
      SequenceID = last(SequenceID),
      Size = last(Size),
      Disciplinas = last(Disciplinas)
    ) %>%
    rowid_to_column("EventId") %>%
    ungroup() %>%
    select(Num, EventId, Size, Disciplinas)

  return (studentPaths)
}
```

Figure 36 - Conversion of students' grades to transactions format

With the grades in the pretended format (it was saved into a text file), as shown in Figure 37, it's read by the read_baskets function which transforms the file into a basket format which is then read by the cspade function in the arulesSequences package. A minimum support of 0.05 was used. The sequences were saved into a CSV file. It's important to refer that although the code presented is in the Rmarkdown file that there is a bug in the package that returns an error running the code. The problem was solved by running the code in Figure 37 in the R console (version 3.5.0).

```r
trans <- arulesSequences::read_baskets(con = "../Data/grades_ARules.txt", info =
c("sequenceID","eventID","SIZE"))


s1 <- arulesSequences::cspade(trans, parameter = list(support = 0.05), control = list(verbose = TRUE, tidLists =
TRUE))

cspade_output <- summary(s1)

capture.output(cspade_output, file = "cspade summary.txt")

write.csv(as(s1, "data.frame"), "../Data/cspade sequences.csv")
```

Figure 37 - Cspade algorithm

As a final step, with the results saved in a CSV file, these are imported to a database (Figure 38).

```
paths <- read_csv("cspade sequences.csv")

for (row in 1:nrow(paths)) {
  support <- as.numeric(paths[row, "support"])
  sequence <- as.character(paths[row, "sequence"])
  saveFrequentPaths(support, sequence)
}
```

Figure 38 - Import of data to a database

After storing the results in a database, the most common paths are presented in Table 8.

Table 8 - Top 25 most common paths (from top to bottom and left to right)

| sequence | support |
|---|---|
| <{C}> | 0.858663 |
| <{B}> | 0.719858 |
| <{C}, {C}> | 0.698075 |
| < {B, C}> | 0.632219 |
| <{C}, {B}> | 0.598784 |
| <{A}> | 0.562817 |
| <{B}, {C}> | 0.56079 |
| <{F}> | 0.54002 |
| <{C}, {C}, {C}> | 0.516717 |
| < {B, C}, {C}> | 0.516211 |
| <{B}, {B}> | 0.505066 |
| <{C}, {B, C}> | 0.502026 |
| < {A, C}> | 0.476697 |
| < {B, C}, {B}> | 0.468085 |
| <{C}, {A}> | 0.459473 |
| <{B}, {B, C}> | 0.418946 |
| <{C}, {B}, {C}> | 0.415907 |
| <{B}, {C}, {C}> | 0.413374 |
| < {A, B}> | 0.404762 |
| <{C}, {C}, {B}> | 0.404762 |
| <{B}, {A}> | 0.404762 |
| <{C}, {C}, {C}, {C}> | 0.402229 |
| <{A}, {C}> | 0.401722 |
| < {B, C}, {B, C}> | 0.39463 |
| < {B, C}, {C}, {C}> | 0.378926 |
| <{A}, {B}> | 0.378926 |
| <{C}, {B, C}, {C}> | 0.370821 |
| < {B, C}, {A}> | 0.370821 |
| <{C}, {F}> | 0.364742 |
| < {A, C}, {C}> | 0.35309 |
| <{C}, {A, C}> | 0.35309 |
| <{C}, {B}, {B}> | 0.35157 |
| <{C}, {C}, {B, C}> | 0.348024 |
| <{B}, {B}, {C}> | 0.343465 |
| <{B}, {C}, {B}> | 0.340426 |
| < {A, B, C}> | 0.337893 |
| < {C, F}> | 0.334853 |

Although this analysis is interesting to detect common paths in student's grades, another approach taken in this work was the addition of the subject to each grade. In Table 9 we can see the most common results. The sequences below are the ones that can be analyzed on the dashboards built in this work.

Table 9 - Most common paths with subjects

| sequence | support |
|---|---|
| <{ESOFT_C}> | 0.33 |
| <{BDDAD_C}> | 0.29 |
| <{FSIAP_C}> | 0.29 |
| <{SGRAI_C}> | 0.28 |
| <{RCOMP_C}> | 0.28 |
| <{ARQCP_C}> | 0.27 |
| <{LPROG_C}> | 0.26 |
| <{EAPLI_C}> | 0.26 |
| <{ESINF_C}> | 0.24 |
| <{FSIAP_F}> | 0.24 |
| <{ALGAV_C}> | 0.24 |
| <{LAPR2_C}> | 0.23 |
| <{PPROG_C}> | 0.23 |
| <{GESTA_C}> | 0.23 |
| <{AMATA_C}> | 0.22 |
| <{LAPR3_C}> | 0.22 |
| <{ASIST_C}> | 0.21 |
| <{MATCP_C}> | 0.21 |
| <{APROG_C}> | 0.21 |
| <{ALGAN_C}> | 0.20 |
| <{SCOMP_C}> | 0.20 |
| <{ESOFT_F}> | 0.20 |
| <{ARQSI_C}> | 0.19 |
| <{BDDAD_F}> | 0.19 |
| <{ESINF_F}> | 0.18 |
| <{MATCP_F}> | 0.18 |
| <{PPROG_F}> | 0.18 |
| <{ANADI_C}> | 0.17 |
| <{LAPR2_F}> | 0.16 |
| <{ARQCP_F}> | 0.16 |
| < {LPROG_C, RCOMP_C}> | 0.15 |
| <{ALGAV_F}> | 0.15 |
| <{INFOR_C}> | 0.15 |
| <{LAPR1_C}> | 0.15 |
| < {SGRAI_C, ALGAV_C}> | 0.15 |
| < {RCOMP_C, EAPLI_C}> | 0.15 |
| < {PPROG_C, ESOFT_C}> | 0.14 |
| < {ESOFT_C, LAPR2_C}> | 0.14 |
| <{LAPR4_B}> | 0.14 |
| <{ALGAN_F}> | 0.14 |
| <{LAPR5_C}> | 0.14 |
| <{MDISC_C}> | 0.14 |
| <{APROG_F}> | 0.14 |
| <{MDISC_A}> | 0.13 |
| <{CORGA_B}> | 0.13 |
| < {ASIST_C, GESTA_C}> | 0.13 |
| < {PPROG_F, ESOFT_F}> | 0.13 |
| < {LPROG_C, EAPLI_C}> | 0.13 |

### 6.2.3 Access to model management through an API

As described in section 5.2, one of the features is the computation of the models presented on this section. In this work we opted for the creation of an API. This type of structure enables the abstraction from the user interface. As such, although the API in this work is called in the dashboard, it isn't mandatory.

The development of the API was done with the help of an R package called Plumber [43]. This package exposes a list of functions as a web service through the use of comments in each function. This work created two functions: 1) Compute and; 2) Predict.

As can be seen in Figure 39, these functions use tags to define how each endpoint should work. In this case, the "@param" is used to define user inputs and the "@get" to set the endpoint as Get request.

```
source("models.R")

#* Recompute the models
#* @param anoInicio The beginning year
#* @param anoFim The ending year
#* @param precedents Compute with precedents
#* @get /compute
function(anoInicio = NULL, anoFim = NULL, precedents= TRUE) {

  anoInicio <- as.numeric(anoInicio) -1
  anoFim <- as.numeric(anoFim) + 1
  computeModel(startYear = anoInicio, endYear = anoFim, withPrecedents = TRUE)

  print(paste0("Model computed with success for: (",anoInicio, "-", anoFim, ") with precedents = ",
precedents))
}

#* Predict student's grades
#* @param anoInicio The beginning year
#* @param anoFim The ending year
#* @get /predict
function(anoInicio, anoFim, algoritmo = "NB") {

  anoInicio <- as.numeric(anoInicio) -1
  anoFim <- as.numeric(anoFim) + 1

  grades <- getGrades(anoInicio, anoFim);

  predictStudents(grades, algorithm = "NB")

  list(msg = paste0("The students grades were correctly predicted between ", anoInicio, "and ", anoFim))
}
```

Figure 39 - Functions exposed by plumber package

The function has as parameters the starting year, end year and precedents. The first two arguments determine the range of data that should be used to compute the models. The last argument, "precedents", is used to determine if only subjects of previous years should be used or if all should be taken into account (the precedent concept is explained in section 6.2.1).

In the second endpoint, the user is able to choose the start and end years as well and, through the "algoritmo" argument, specify which algorithm to use. Currently, there are three types: 1) NB (Naive Bayes); 2) DT (Decision Tree) and; 3) RF (Random Forest).

The results of the model performance and the predictions are saved on an SQL Server database (the results stored in this database were presented and analyzed on section 6.3.2),

To interact with the API without the help of external tools, after the initialization of the API, it's possible to use Swagger which, as shown in Figure 40, presents each endpoint and enables the user to send a request to each one.

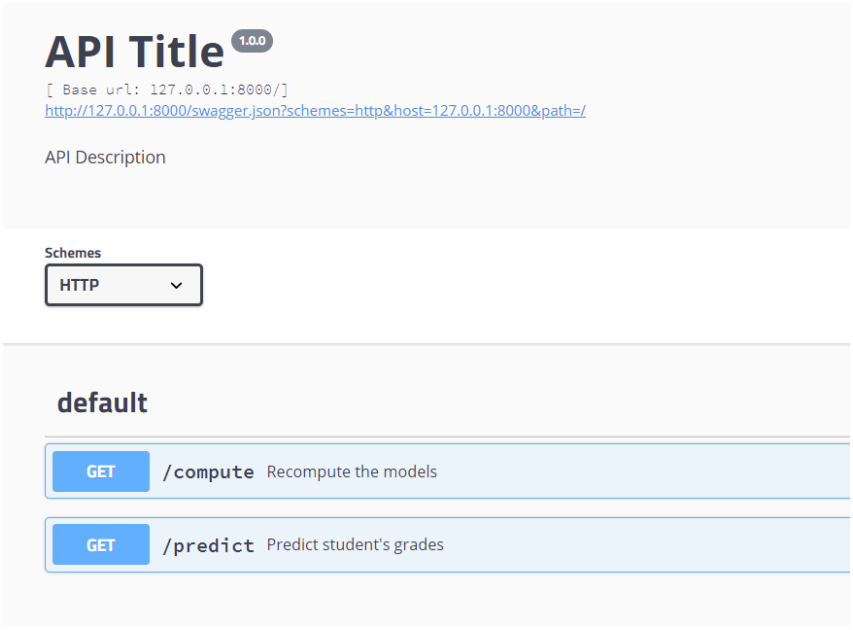



Figure 40 - API running Swagger

It's important to refer that, if the models were built with data that omits one of the subjects, it will throw an error when trying to predict the students' outcome as it will require the same features as the ones used in the modeling phase.

## 6.3 Development of the Dashboards

### 6.3.1 Structure of the dashboard's data

The Power BI services enable the possibility of creating relationships between multiple tables of data (i.e., the student table can be related to grades by its identification number). This type of relationship is very important as it enables the propagation of filters between multiple graphs and the division of the data in logical tables. In section 6.1.2 there's a representation of the model developed for the analysis of the grades which will guide in the presentation of the analysis.

Inside the table, there are other two entities: columns and measures. While the column represents a variable in the table, the measure is a single value which can be as simple as a dynamic title or the average of the grades. The names in the model are all in Portuguese as these dashboards are intended for LEI, a Portuguese institution.

#### 6.3.1.1 Tables

The model is divided into 4 tables which are 1) Aluno (table of student shown in Figure 41); 2) Unidades Curriculares (Subjects); 3) Notas (Grades); 4) Previsões (Predictions); 5) Performance Modelos (Model Performance); 6) Modelos (Models); and 7) Caminhos Frequentes (Main

Paths). The first table summarizes the data for each student by using the *summarizeColumns* command which aggregates all the data for each student number (*unique* for each student).

```
Aluno =
SUMMARIZECOLUMNS (
    'Notas'[Num],
    "Numero de Anos Frequência", DISTINCTCOUNT ( 'Notas'[Ano Frequência] )
)
```

Figure 41 - Student's Analysis table formula

The Subject's table is related to the info of each subject. The Subject table is created with the command Distinct which creates a list of distinct items for the given column (Figure 42).

```
Disciplinas = DISTINCT('Grades LEI'[Disciplina])
```

Figure 42 - Subject's table formula

The Prevision table holds all the predictions of the students' approval at the subjects, the Performance Models holds all models created and respective performance metrics (which, like the grades table is used to generate the Models table) and the Main Paths presents the most common paths taken by the students.

#### 6.3.1.2 Columns

Each table has at least one column. The student analysis has two columns that aggregate the data for each student. The first one is "Número de Reprovações" (Figure 43) which counts the number of subjects in which the final grade was below 9.5 (the column "Resultado Numérico" used for this formula substitutes text with 0).

```
Número de Reprovações = CALCULATE(
                            COUNTROWS('Grades LEI')
                            , 'Grades LEI'[Resultado Numerico] < 9.5,
                             FILTER('Grades LEI', 'Análise de Aluno'[Num] = 'Grades LEI'[Num])) + 0
```

Figure 43 - Number of failures formula

The second column is "Com Reprovações" (With failures), which is true if the student has had at least one failing subject (checks the first column). The formula uses an if clause as can be seen in Figure 44.

```
Com Reprovações =
IF ( [Número de Reprovações] = 0, "Sem Reprovações", "Com Reprovações" )
```

Figure 44 - “With Failures” formula

In the table of Subjects, there are three columns beside the subject description: 1) “Ano” (Year), “Ano Descrição” (Description Year shown in Figure 45) and “Semestre” (Semester). The three columns use a switch statement to complement the dataset with the year and semester in which the subject is lectured. This information is, as such, considered static.

```
Ano Descrição = SWITCH( Disciplinas[Ano]
                , 1, "1º Ano"
                , 2, "2º Ano"
                , 3, "3º Ano"
                , "Inválido")
```

Figure 45 - Switch statement for “Ano Descrição” column

The grades’ table is the main table used in the analysis. The initial ones were already described in section 6.1.1 but Table 10 lists all the columns.

Table 10 - Students’ grades (part 1)

| Column Name | Description |
|---|---|
| **Ano** | Year of the grade |
| **Ano Descrição** | Year as text |
| **Ano frequência** | Year in which the exam was taken |
| **AnoFreq** | The year in which the subject occurred |
| **Anos De Curso** | Counts the number of years the student has been doing the program |
| **Aprovado** | If the Student was approved at the end of the subject |
| **Aprovado Descrição** | Description if the student was approved or not approved. This is applied to the graphs to make the final situation of the students clear for the user |
| **Código Num + Disciplina** | Concatenation of the Number and Subject |
| **ContagemCadeiras** | Counts the total number of distinct subjects realized by the student |
| **ContagemCadeirasAprovadas** | Counts the total number of distinct subjects the student has been approved on |
| **Data Exame** | Exam’s date |
| **Data Freq.** | Frequency’s date |
| **Data Resultado** | Final Result Date |
| **Disciplina** | Subject name |
| **Época** | Season in which the student got the final grade |
| **Exame** | Exam grade |
| **Exame Numérico** | Substitutes failure status like “DT” and “FT” with 0 |

Table 11- Students' grades (part 2)

| Freq. | Frequency grade |
|---|---|
| **Frequência Numérica** | Substitutes failure status like "SM" and "NF" with 0 |
| **Licenciado,** | True if the student has finished the program |
| **Licenciado Descrição** | Describes if the student has finished the program or not |
| **Num** | Students' number |
| **Número de Frequência da Cadeira** | Using the column "Código Num + Disciplina" it counts the number of times the student has repeated the same subject |
| **Resultado Simplificado** | Approves a student if the final grade is superior or equal to 10 |
| **Resultado** | Final Result |
| **Resultado Novo Sistema** | Final Result using the new system |
| **Resultado Numérico** | Transforms all the final results' grades to numeric |
| **Semestre** | Semester |

Table 10 only describes the columns generated using DAX. These columns can be divided into the ones used to detect if a student finished the program and the one that presents the grades in a numerical format.

The columns "Aprovado", "AnoFreq", "ContagemCadeiras" and "ContagemCadeirasAprovadas" were developed in R and added to the source file read by Power BI. R was used in this case due to the easiness of development of more difficult queries like these.

For the functions presented in Figure 46 the following assumptions were made:

- A grade of at least 10 was required to pass;
- A curricular year starts in September and ends in July;
- The grades are given in January, February, and July;
- Grades given in September and December occur in a special season and correspond to the previous curricular year;
- For a student to be considered as having completed the program, he needs to have been approved in at least 30 subjects. (doesn't take into account students with equivalent subjects from other programs).

```
studentGrades <- grades %>%
  as_tibble() %>%
  left_join(disciplinas, by = c('Disciplina')) %>%
  mutate(
  Aprovado = ifelse(!is.na(as.numeric(Resultado)) &
  as.numeric(Resultado) >=
  10, TRUE, FALSE),
  AnoFreq = ifelse(
  lubridate::month(DataResultado) == 12,
  lubridate::year(DataResultado),
  lubridate::year(DataResultado) - 1
  ),
  Disciplina = replace(Disciplina, Disciplina == "P/EST-ERASMUS 20", "Erasmus20"),
  Disciplina = replace(Disciplina, Disciplina == "P/EST-ERASMUS 10", "Erasmus10"),
  ResultadoICTS = case_when(
  is.na(as.numeric(Resultado)) ~ "F",
  as.numeric(Resultado) < 10 ~ "F",
  as.numeric(Resultado) < 14 ~ "C",
  as.numeric(Resultado) < 16 ~ "B",
  as.numeric(Resultado) < 18 ~ "A",
  as.numeric(Resultado) < 20 ~ "A+"
  )
  ) %>%
  group_by(Num) %>%
  arrange(desc(DataResultado)) %>%
  distinct(Disciplina, .keep_all = TRUE) %>%
  mutate(
  ContagemCadeiras = n(),
  ContagemCadeirasAprovadas = ifelse(Aprovado == TRUE, n(), NA),
  AnosDeCurso = n_distinct(AnoFreq)
  ) %>%
  group_by(Num, AnoFreq) %>%
  mutate(Licenciado = ifelse(ContagemCadeirasAprovadas < 30, FALSE, TRUE))
```

Figure 46 - Function in R to generate new columns

#### 6.3.1.3 Measures

As the data model of this analysis is very simple, the measures are all directly associated with the grades' table. These can be seen in Table 12.

Table 12 - List of measures

| Measure Name | Description |
|---|---|
| **% Reprovações** | % of Failures |
| **Média da Frequência** | Average of frequency grades |
| **Média da Frequência de Aprovados** | Average of positive frequency grades (≥ 9.5) |
| **Média de Exames** | Average of exam grades |
| **Média de Exames Aprovados** | Average of positive exam grades |
| **Média de Reprovações por aluno** | Average of failures per student |
| **Média de Resultados** | Average of final results |
| **Número de Reprovações** | Total number of failures |
| **Total de Alunos** | Total number of students |
| **Total de Notas** | Total number of given grades |

These measures can be divided into relative calculations (percentage of failures), averages of grades and counters (students and grades). The first category divides the total number of failures by the number of students. The averages are all created using the structure presented in Figure 47. The calculate function is used in two steps, the aggregation function and the filters (in the case of Figure 47 it filters by all exam grades above 9.4). The third category is the total which is calculated either by counting the distinct values (different student numbers) or the total number of rows of the dataset (total number of grades).

```
Média de Exames Aprovados = CALCULATE(
                                AVERAGE('Notas'[Exame Numérico])
                                , 'Notas'[Exame Numérico] > 9.4)
```

Figure 47 - Formula of the average of approved exam grades

### 6.3.2 Dashboards for students' grades analysis

Dashboard used for the analysis of the data focused on 1) Analysis of the students' grades (6.3.2.1); 2) Analysis of students' final results (6.3.2.2) and; 3) Analysis of each subject (6.3.2.3).

#### 6.3.2.1 Grades

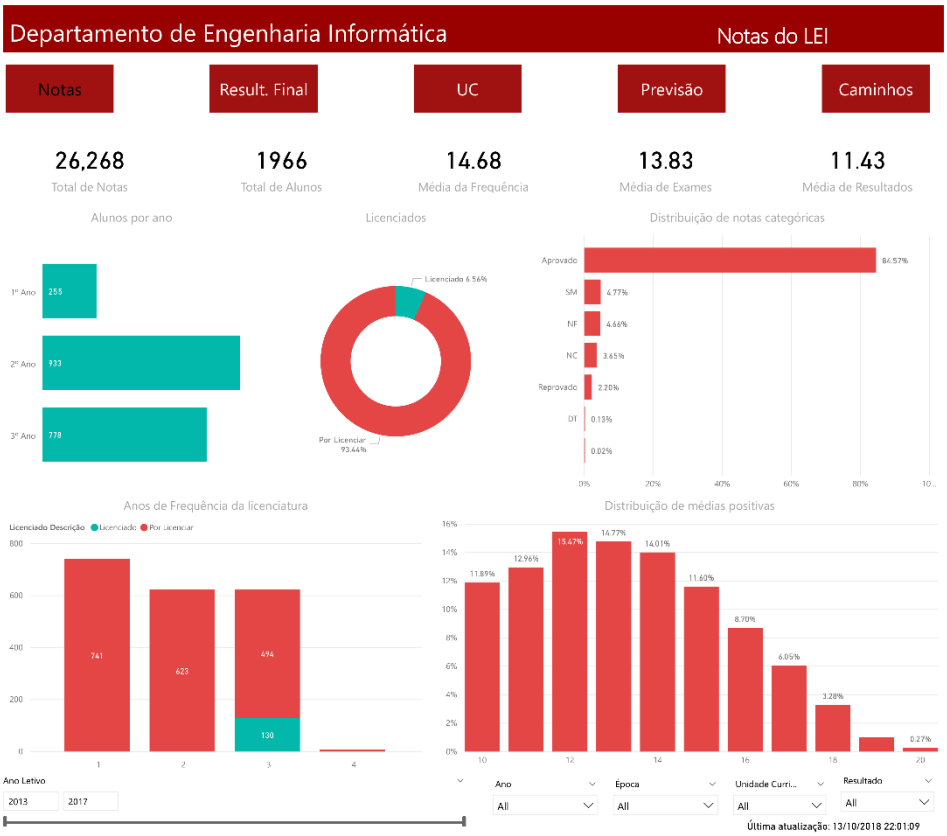



Figure 48 - Dashboard of the grades

This dashboard (Figure 48) is divided into 5 cards, 4 bar charts, and 1 pie chart. In the cards, we can see the total number of students, grades and average positive final grades for each type of evaluation. Next, from left to right and up to bottom, the following visuals are presented: 1) Students by year; 2) Graduated students; 3) Percentage of categorical grades; 4) Number of years the student is frequenting the program and 5) Distribution of positive grades.

### 6.3.2.2 Final Results

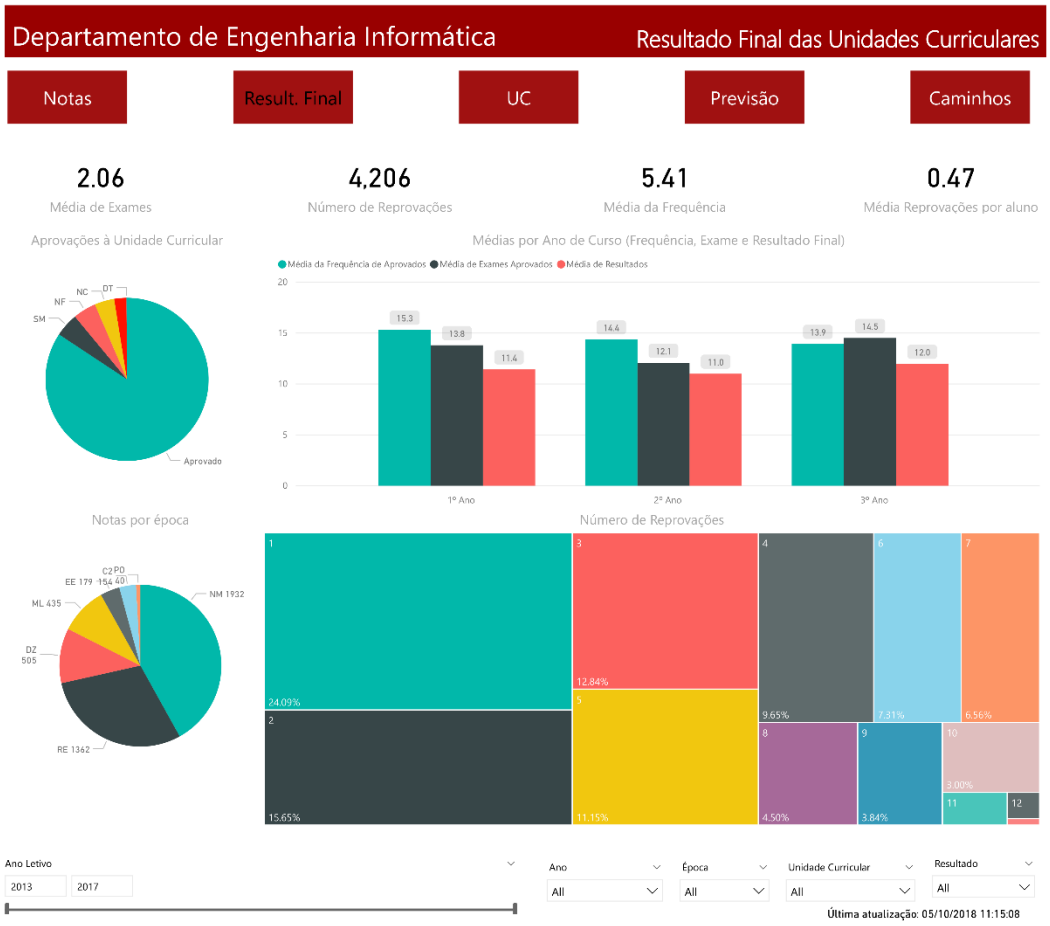



Figure 49 - Second dashboard of students' grades

The second dashboard (Figure 49) is divided into 4 cards, two pie charts, one bar chart, and a treemap. The cards show: 1) the average of students' exam grades without excluding the negative grades; 2) a total number of students' failures; 3) the average of students' frequency grades without excluding the negative grades, and 4) the average of failures for each student.

The visuals are distributed in the following way (from left to right and up to bottom): 1) Pie chart with the categorical grades; 2) Average of the grades by year and type of grade (frequency,

exam and final grade); 3) pie chart with the distribution of grades by seasons (see section 6.1.1.1) and; 4) Treemap with the distribution of the number of failures each student has.

#### 6.3.2.3 Subjects analysis

In Figure 50, there's a focus on the analysis of the grades for each subject which is shown as 1 box plot and 2 bar plots. The visuals are (from left to right and up to down): 1) Range of grades for each subject (maximum, minimum, average and standard deviation); 2) Top 5 subjects with the highest number of failures and; 3) Total number of students by subject and final result (approved or failed).

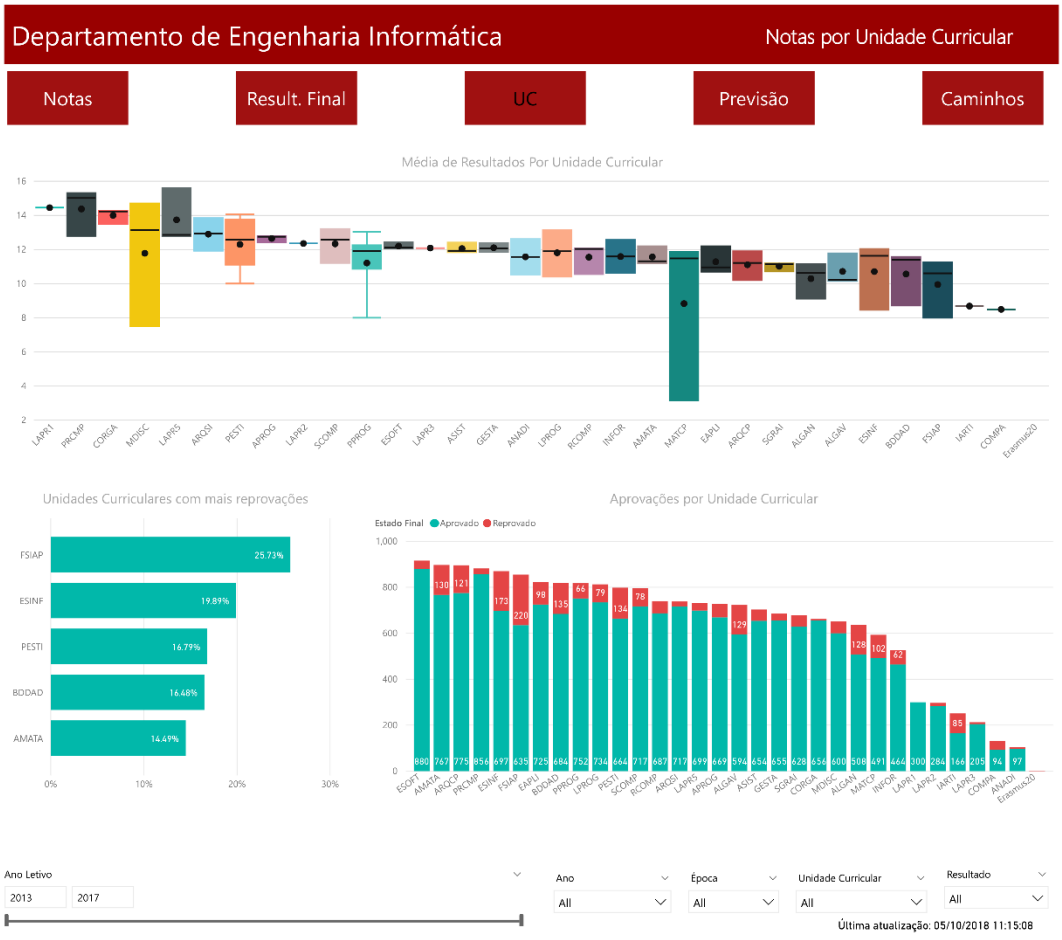



Figure 50 - Subject analysis

### 6.3.3 Prediction's Dashboards

#### 6.3.3.1 Prediction of students' outcomes

The results of the models were also integrated into the dashboard (for the detail of the development of these models see section 6.2). In Figure 51 the result of the prediction of each students' grade is presented in a single matrix. As the models required that the student completed some subjects for it to be able to predict, no predictions were made during the first semester of the first year. The predictions were also made only for the semester following the last subject the student had done (ex: if the students' last subject was FSIAP, the predictions would be made for the subjects in the second semester of the second year). If the student was predicted to have passed, it appears with a green label and, if it failed, in red. At the right side of each prediction, there's the algorithm used with the respective accuracy, to give the user an idea of how good the prediction was.

| Ano | | | | | 1º Ano | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Semestre | | | 1 | | | | | 2 | | |
| Num | ALGAN | AMATA | APROG | LAPR1 | PRCMP | ESOFT | LAPR2 | MATCP | MDISC | PPROG |
| 358751 | Por fazer | Por fazer | Por fazer | Por fazer | Por fazer | | | | | |
| 358756 | C | B | C | A | C | C | B | A | B | B |
| 359457 | A | A | B | B | B | C | C | B | A | B |
| 359753 | Por fazer | Por fazer | C | B | C | C | | C | C | C |
| 359757 | F | B | C | C | C | C | Passa RF 88.31% | C | C | C |
| 360456 | C | A | C | A | B | C | B | C | A | B |
| 360755 | Por fazer | Por fazer | Por fazer | Por fazer | Por fazer | | | | | |
| 360758 | B | C | A | B | A | C | C | A | B | C |
| 361455 | Por fazer | Por fazer | F | A+ | Por fazer | C | B | C | | |
| 361757 | Por fazer | Por fazer | B | A | B | B | B | B | A+ | C |
| 361759 | A | B | C | A | C | C | A | A | B | A+ |
| 362454 | A | C | C | C | C | Passa RF 86.76% | Reprovado RF 88.31% | Passa RF 90.60% | C | Passa RF 89.24% |
| 362759 | Por fazer | Por fazer | Por fazer | Por fazer | Por fazer | | | | | |
| 362760 | B | C | B | B | C | C | A | A | A+ | A |
| 363453 | C | C | C | C | C | Passa RF 86.76% | Passa RF 88.31% | Passa RF 90.60% | Passa RF 91.36% | Passa RF 89.24% |
| 363761 | A | A | C | A | A | B | A | C | B | A |
| 364452 | C | A | B | A | B | C | B | C | A | C |
| 364762 | C | C | C | A | C | C | C | C | B | C |
| 364763 | Por fazer | Por fazer | Por fazer | Por fazer | Por fazer | | | | | |

Figure 51 - Students' Prediction

If the user presses the button with the label "Avançados" it will be presented with a detailed view of the created models (see Figure 52 and Figure 53). In this page the user is able to: 1) See a table with a history of all the models created and; 2) See a bar plot of the best models for each subject and; 3) Make calls to a REST API to either compute the models or predict the students' grades.

The table with the models' performance can be seen in Figure 52. This table is divided into 6 columns: 1) Algorithm; 2) Subject; 3) Type of performance evaluation; 3) Model that takes into account precedents or not; 4) Value of the performance in percentage and; 5) Date of the model creation.

Desempenho dos modelos

| Algoritmo | Unidade Curricular | Avaliação de Performance | Tipo | Valor | Date |
|---|---|---|---|---|---|
| RF | ANADI | accuracy | PR | 97.70% | 26-Sep-18 22:28:41 |
| RF | ANADI | mean | PR | 97.70% | 26-Sep-18 22:28:41 |
| DT | ANADI | accuracy | PR | 97.35% | 26-Sep-18 22:28:41 |
| DT | ANADI | mean | PR | 97.35% | 26-Sep-18 22:28:41 |
| RF | CORGA | accuracy | PR | 96.30% | 26-Sep-18 22:28:43 |
| RF | CORGA | mean | PR | 96.30% | 26-Sep-18 22:28:43 |
| RF | ANADI | accuracy | PR | 95.94% | 26-Sep-18 22:13:49 |
| RF | ANADI | mean | PR | 95.94% | 26-Sep-18 22:13:49 |
| DT | ANADI | accuracy | PR | 95.76% | 26-Sep-18 22:13:49 |
| DT | ANADI | mean | PR | 95.76% | 26-Sep-18 22:13:49 |
| DT | CORGA | accuracy | PR | 95.59% | 26-Sep-18 22:28:43 |
| DT | CORGA | mean | PR | 95.59% | 26-Sep-18 22:28:43 |
| RF | LAPR4 | accuracy | PR | 95.41% | 26-Sep-18 21:56:00 |
| RF | LAPR4 | mean | PR | 95.41% | 26-Sep-18 21:56:00 |
| RF | LPROG | accuracy | PR | 95.05% | 26-Sep-18 22:04:16 |
| RF | LPROG | mean | PR | 95.05% | 26-Sep-18 22:04:16 |

Figure 52 - Models' performance

This list of models can be seen on the bar plot (Figure 53) ranked from best to worst. The legend differentiates the year in which the subject is taken.

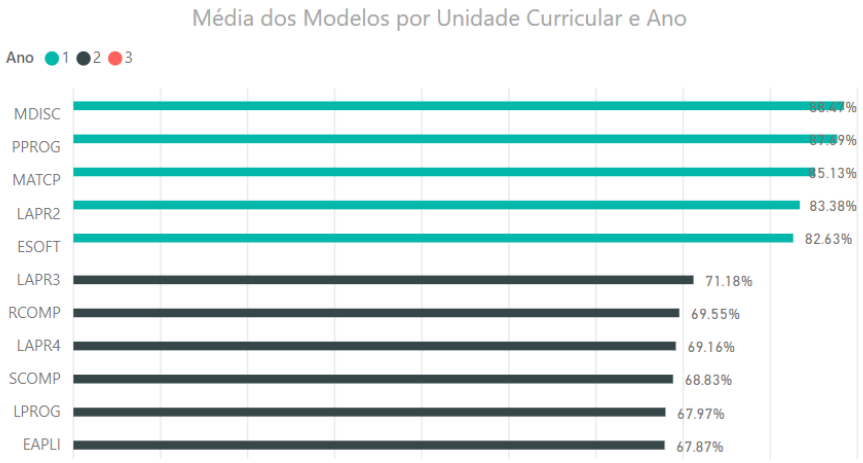



Figure 53 - Bar plot of the models' performance

The Rest API is, at the time of writing, only available in a local machine. The user is able to select the years he wants to take into account for creating the model and which algorithm is preferred when predicting the students' grades (Figure 54).

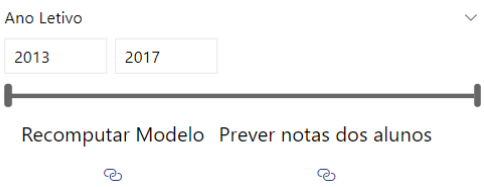



Figure 54 - API's options

#### 6.3.3.2 Students' most common paths

The last page of the dashboard (Figure 55) is dedicated to presenting the most common paths in a single table ordered by its support (see section 3.2.1.3). Each element represents a subject and respective grade in an ECTS scale. Each year of the student is separated by "{}" (i.e. {LPROG_C → RCOMP_C} refers to students that had a C in LPROG followed by a C in RCOMP in the same year).

Sequências de cadeiras mais frequentes

| Sequência | Suporte |
|---|---|
| {LPROG_C --> RCOMP_C} | 15.41% |
| {RCOMP_C --> EAPLI_C} | 14.55% |
| {SGRAI_C --> ALGAV_C} | 14.55% |
| {PPROG_C --> ESOFT_C} | 14.50% |
| {ESOFT_C --> LAPR2_C} | 14.34% |
| {ASIST_C --> GESTA_C} | 13.43% |
| {PPROG_F --> ESOFT_F} | 13.28% |
| {LPROG_C --> EAPLI_C} | 13.22% |
| {RCOMP_C --> BDDAD_C} | 13.07% |
| {RCOMP_C --> ARQCP_C} | 12.72% |
| {RCOMP_C --> LAPR3_C} | 12.46% |
| {BDDAD_C --> FSIAP_C} | 12.31% |
| {ARQCP_C --> EAPLI_C} | 12.21% |
| {RCOMP_C --> FSIAP_C} | 12.21% |
| {ESOFT_F --> LAPR2_F} | 12.11% |
| {ESOFT_C} --> {BDDAD_C} | 11.90% |
| {LPROG_C --> BDDAD_C} | 11.85% |
| {SGRAI_C --> GESTA_C} | 11.80% |
| {SGRAI_C --> ASIST_C} | 11.70% |
| {EAPLI_C --> LAPR3_C} | 11.65% |
| {EAPLI_C} --> {GESTA_C} | 11.65% |
| {BDDAD_C --> EAPLI_C} | 11.60% |
| {EAPLI_C} --> {SGRAI_C} | 11.50% |
| {LPROG_C --> ARQCP_C} | 11.50% |
| {ESOFT_C} --> {RCOMP_C} | 11.44% |
| {ESOFT_C} --> {LPROG_C} | 11.34% |

Figure 55 - Most common paths page

# 7 Analysis of the Case Study (LEI's Grades)

Following the development of the solution, this chapter used it for the analysis of LEI's program using the dataset provided by the LEI' direction. This analysis is presented in the following way: 1) Grades; 2) Final Results; 3) Subject Analysis; 4) Prediction Analysis; 5) Students' most frequent paths and; 6) Data Analysis Conclusion.

## 7.1 Grades

In the first graph (Figure 56) the students are divided into the three years of the program where we can see that there is a higher number of students in the second year and a considerable lower number of students in the first year.

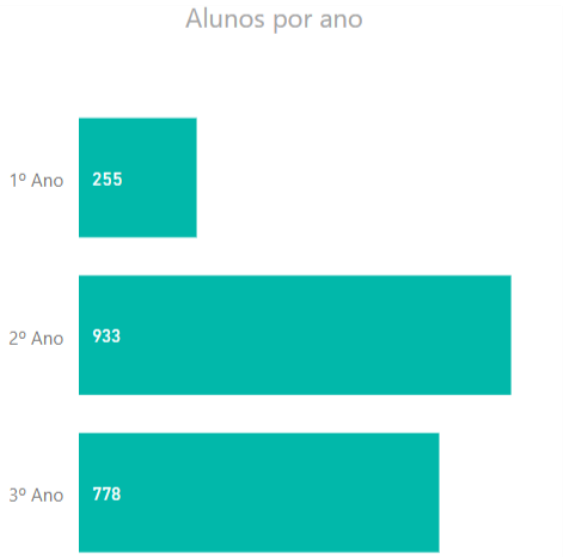



Figure 56 - Students by year

Figure 57 shows the distribution of grades for the students. As can be seen, the vast majority of grades are positive. There's a distinction between failure grades and the one with label "Reprovado" which are the ones that got a final grade below 10. This type of grade is less frequent than failures associated with students dropping due to failing to have a minimum grade at one of the evaluations ( "SM") or simply missing evaluations ("NF", "NC").

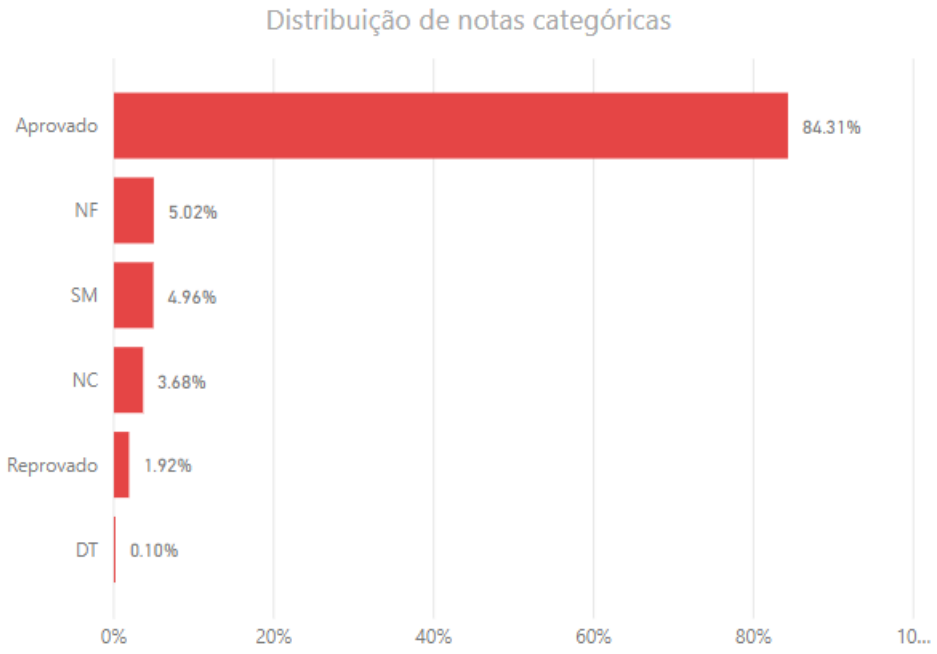



Figure 57 - Range of categorical grades

The visualization of the number of student registrations (years in the program) shows not only how many were able to finish the program in three years but also the distribution of the students at any given moment. In Figure 58 it's presented the sum of all the years showing that most students have up to three years. The blue region indicates the students that already finished the program.

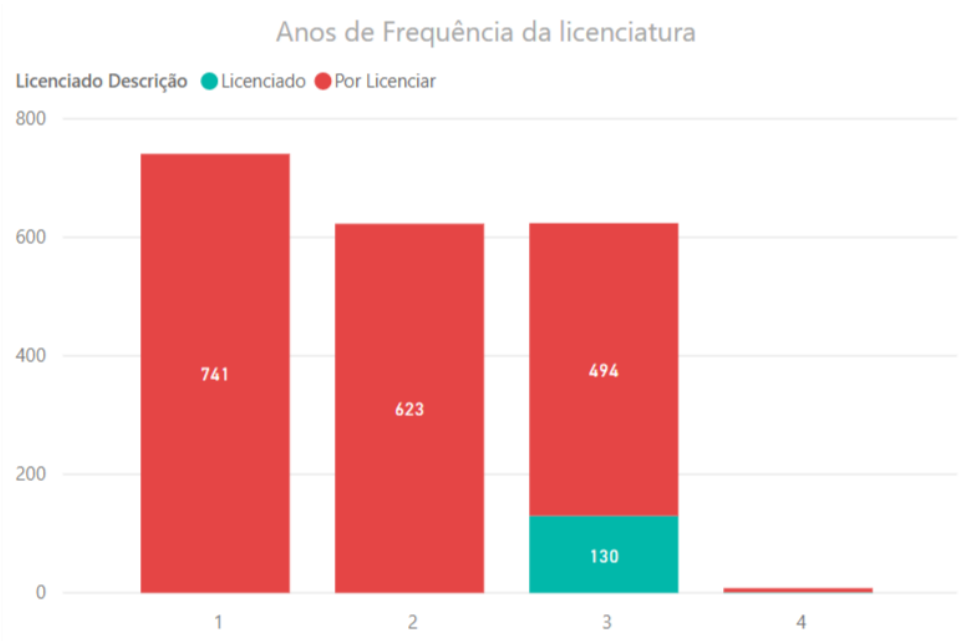



Figure 58 - Number of years in the program

The final chart in the dashboard (Figure 59) presents the distribution of the students' grades (above 10). In this case, all subjects are represented and it's possible to detect that, as expected, the student's grades follow a Gaussian distribution.

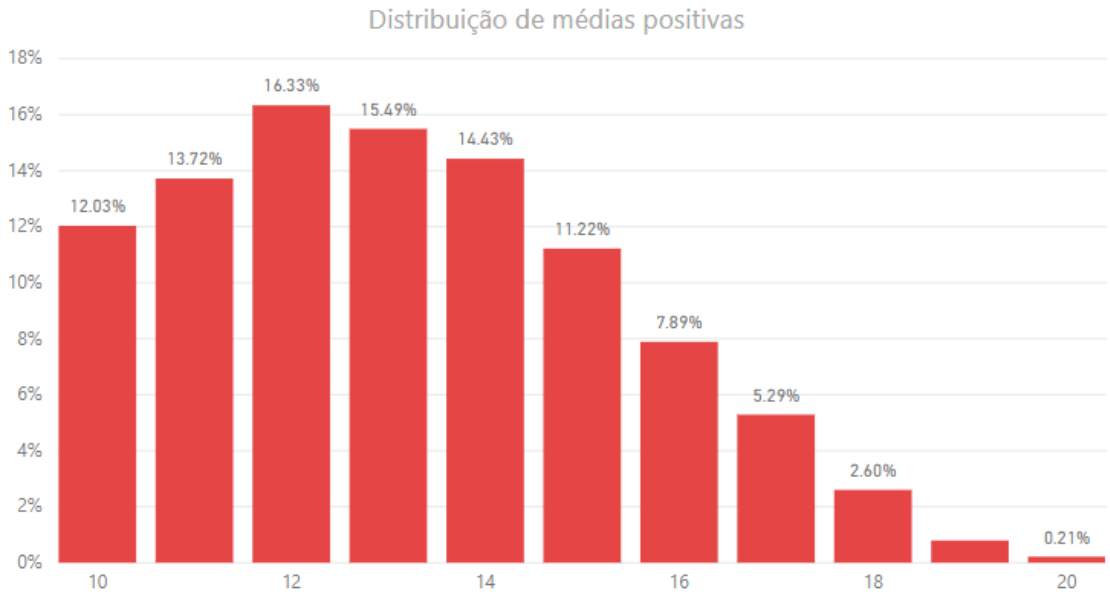



Figure 59 - Distribution of positive grades

## 7.2 Final Results

The bar chart shows that most students have the worst grades in the second year and the highest grades on the final year.

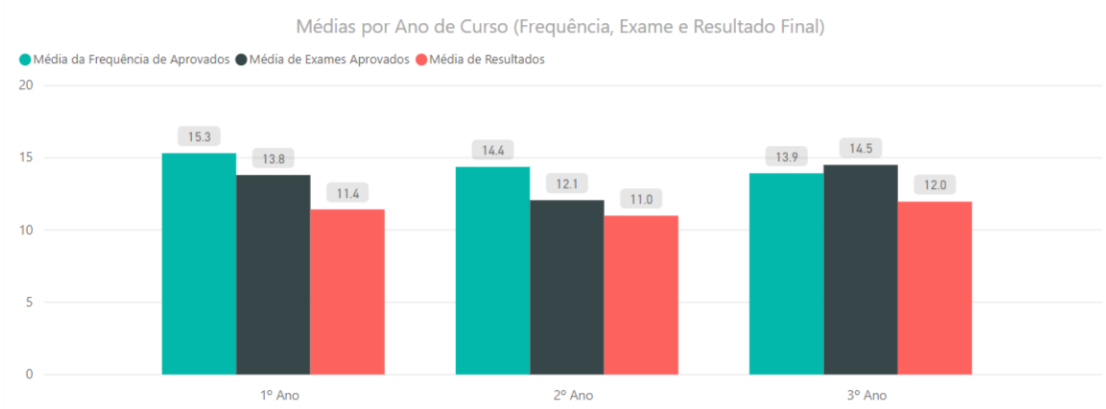



Figure 60 - Average of grades by year and type of evaluation

Interestingly, if we select only evaluations of type "Melhoria" (ML) grades are significantly higher (Figure 61), indicating that students do improve their grades when they take the effort to try.

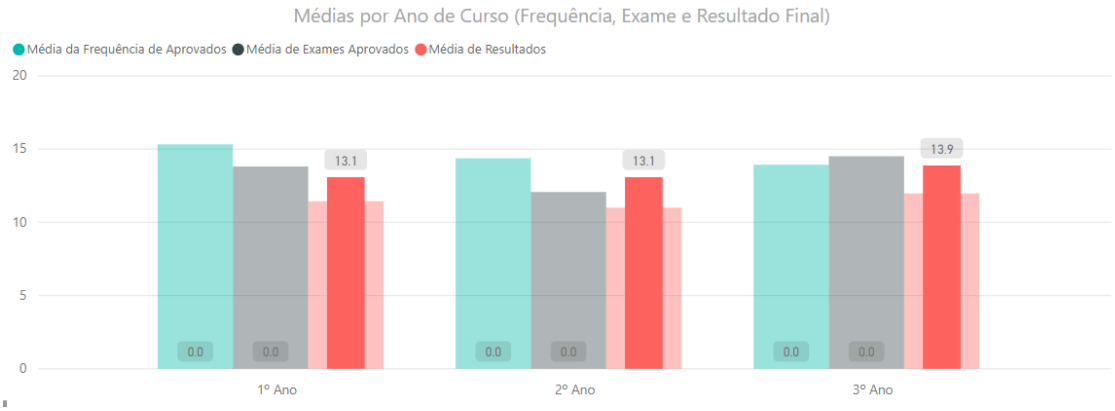



Figure 61 - Average of grades by year and type of evaluation considering only ML grades

As can see from the pie chart in Figure 62, almost 42% of the students were able to pass the subjects without the need of going to other phases except the “NM” (normal).

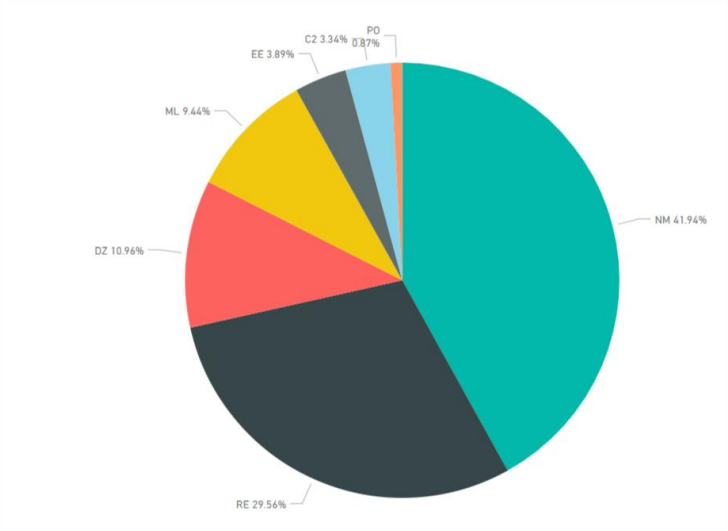



Figure 62 - Grades by phase

Finally, the treemap shows that almost half of the students (45%) never fail at a subject. Additionally, it can be observed that the higher the number of failing subjects, the lower the number of students.

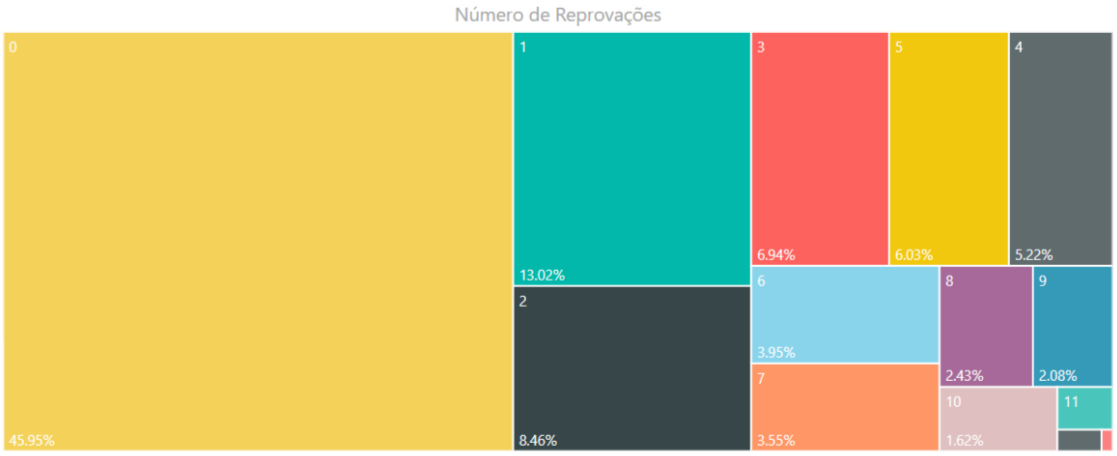



Figure 63 - Number of failures for each student

## 7.3 Subject Analysis

In the boxplot showing the range of results for each subject (Figure 64), it's possible to obverse that subjects like FSIAP, BDDAD, ESINF, PPROG and MATCP have the highest ranges and lowest averages.

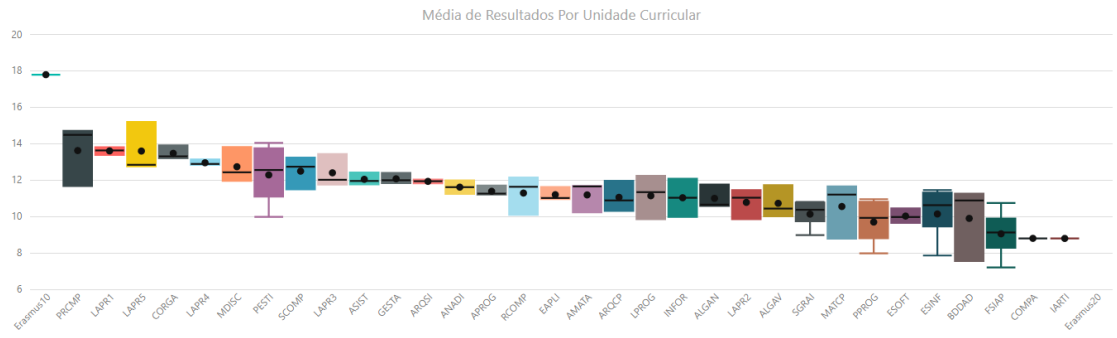



Figure 64 - Range of grades by subject

Interestingly, approximately the same subjects are the ones showing up in the second second visual (Figure 65) and thus not only have we the lowest results and highest ranges but also present the highest number of failures. FSIAP is a physics subject, ESINF focus on algorithms, MATCP is a mathematical subject and PPROG and ESOFT are programming subjects of an introductory level.

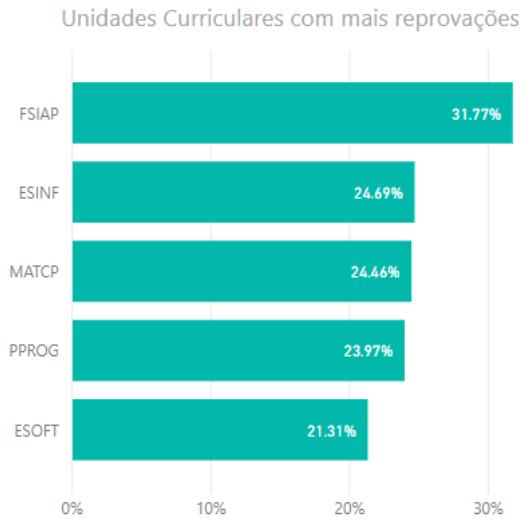



Figure 65 - Top 5 subjects by a number of failures

In Figure 66 we can see that the subjects with more students are either from the first year (ESOFT, MATCP, MDISC, PPROG, AMATA) or are the ones with more failures (i.e. FSIAP and MATCP, PPROG, ESOFT, ESINF). From this figure and the help of the previous visual, we can detect that there are many students failing in mathematical and physics subjects (MATCP, FSIAP). A subject that is of particular notice is PESTI. Although at first glance it seems many students are given a negative grade, the high number of failures is due to many of them, instead of taking the subject to the end (delivering the final work and doing the presentation) give up earlier. In the dataset, these students are represented with a grade of "NC".

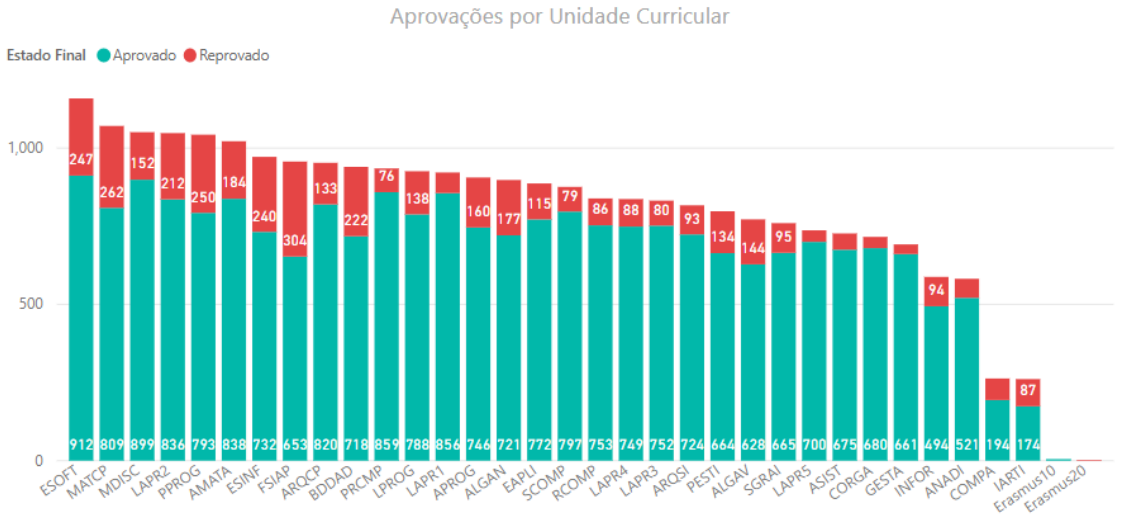



Figure 66 - Total of students by subject and final result

## 7.4 Prediction Analysis

The students' predictions are presented as a matrix (Figure 67). No predictions were made in the first semester of the program. Predictions for one semester are calculated after a grade is attributed to a subject of the previous semester. Thus, for example in the case of student 218471, despite the fact that he/she has passed only at one subject with a grade of B (LAPR1), this is enough to predict that it will pass at ESOFT, LAPR2, and MATCP. However, the system alerts that he is at risk of failing at MDISC and PPROG. Another situation we can consider is the one from the student 219473 which failed at all subjects in the first semester and the models conclude that he is at risk of failing at all subjects in the second semester. This situation occurs frequently in the current dataset as there are many students for which we don't have all their previous grades.

| Ano | | | | | | 1º Ano | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Semestre | | | 1 | | | | | 2 | | |
| Num | ALGAN | AMATA | APROG | LAPR1 | PRCMP | ESOFT | LAPR2 | MATCP | MDISC | PPROG |
| 217599 | B | A | A | A+ | A | B | A | A | B | A |
| 217615 | B | B | B | B | A | B | A | A | A | A |
| 218471 | Por fazer | Por fazer | Por fazer | B | F | Passa RF 86.76% | Passa RF 88.31% | Passa RF 90.60% | Reprovado RF 91.36% | Reprovado RF 89.24% |
| 218598 | B | A | A+ | A+ | B | A+ | A+ | A | | A+ |
| 218616 | A | A | C | B | C | C | C | A | A+ | A |
| 219473 | Por fazer | Por fazer | F | Por fazer | Por fazer | Reprovado RF 86.76% | Reprovado RF 88.31% | Reprovado RF 90.60% | Reprovado RF 91.36% | Reprovado RF 89.24% |
| 219597 | C | C | F | A+ | B | C | C | C | B | C |
| 219617 | F | C | F | C | C | Reprovado RF 86.76% | Passa RF 88.31% | Reprovado RF 90.60% | Reprovado RF 91.36% | Reprovado RF 89.24% |
| 220475 | Por fazer | Por fazer | Por fazer | Por fazer | Por fazer | | | | | |
| 220596 | F | F | B | A | C | C | B | | B | C |
| 220618 | C | C | C | A | B | C | C | C | B | C |
| 221477 | Por fazer | Por fazer | Por fazer | Por fazer | Por fazer | | | | | |
| 221595 | Por fazer | Por fazer | Por fazer | Por fazer | F | Reprovado RF 86.76% | Reprovado RF 88.31% | Reprovado RF 90.60% | Reprovado RF 91.36% | Reprovado RF 89.24% |
| 221619 | C | B | B | B | C | C | C | A | C | C |
| 222479 | Por fazer | Por fazer | Por fazer | Por fazer | Por fazer | | | | | |
| 222594 | A+ | A | A | A | A | A | A+ | A+ | A+ | A |
| 222620 | B | A | C | C | C | C | C | C | C | C |
| 223481 | Por fazer | Por fazer | Por fazer | Por fazer | Por fazer | | | | | |
| 223593 | C | F | B | A+ | A | A | A | | B | A |
| 223621 | A | A | B | A | B | C | C | A+ | A | A |
| 224483 | F | F | F | F | F | Reprovado RF 86.76% | Reprovado RF 88.31% | Reprovado RF 90.60% | Reprovado RF 91.36% | Reprovado RF 89.24% |

Figure 67 - Students predictions

## 7.5 Students' most frequent paths

Although the sequences were divided by year, the most common ones are of students that did only one or two subjects. The most prominent type of grade is C (between 10 and 14 on a scale of 0 to 20). Another pattern that also emerges is that student grades don't change very much. In any given moment, it's more common for a student to have only C's than to have an A and C in the same year. The most predominant cases are with a single subject which help indicate the most common grade of each subject (i.e. ESOFT_C indicates most students have a grade between 10 and 14). Other cases with more than one subject, indicate stronger relations than the previous sequences (i.e. There's a relation, with the support of 13%, of the student failing at PPROG if he failed at ESOFT as seen in Figure 68).

Sequências de cadeiras mais frequentes

| Sequência | Suporte |
|---|---|
| {LPROG_C --> RCOMP_C} | 15.41% |
| {RCOMP_C --> EAPLI_C} | 14.55% |
| {SGRAI_C --> ALGAV_C} | 14.55% |
| {PPROG_C --> ESOFT_C} | 14.50% |
| {ESOFT_C --> LAPR2_C} | 14.34% |
| {ASIST_C --> GESTA_C} | 13.43% |
| {PPROG_F --> ESOFT_F} | 13.28% |
| {LPROG_C --> EAPLI_C} | 13.22% |
| {RCOMP_C --> BDDAD_C} | 13.07% |
| {RCOMP_C --> ARQCP_C} | 12.72% |
| {RCOMP_C --> LAPR3_C} | 12.46% |
| {BDDAD_C --> FSIAP_C} | 12.31% |
| {ARQCP_C --> EAPLI_C} | 12.21% |
| {RCOMP_C --> FSIAP_C} | 12.21% |
| {ESOFT_F --> LAPR2_F} | 12.11% |
| {ESOFT_C} --> {BDDAD_C} | 11.90% |
| {LPROG_C --> BDDAD_C} | 11.85% |
| {SGRAI_C --> GESTA_C} | 11.80% |
| {SGRAI_C --> ASIST_C} | 11.70% |
| {EAPLI_C --> LAPR3_C} | 11.65% |
| {EAPLI_C} --> {GESTA_C} | 11.65% |
| {BDDAD_C --> EAPLI_C} | 11.60% |
| {EAPLI_C} --> {SGRAI_C} | 11.50% |
| {LPROG_C --> ARQCP_C} | 11.50% |
| {ESOFT_C} --> {RCOMP_C} | 11.44% |
| {ESOFT_C} --> {LPROG_C} | 11.34% |
| {SCOMP_C --> ARQCP_C} | 11.34% |
| {PPROG_F --> LAPR2_F} | 11.29% |
| {RCOMP_C --> ESINF_C} | 11.29% |
| {ARQCP_C --> FSIAP_C} | 11.14% |
| {BDDAD_C --> ESINF_C} | 11.14% |

Figure 68 - Students' most common paths

## 7.6 Data Analysis Conclusion

From the dashboards presented in section 6.1, there are some conclusions that can be drawn:

- As seen in section 7.2, grades are lower in the case the student derive their final grade from the exam instead of frequencies plus exam;
- Students that take the effort to improve grades in the ML phase usually do succeed in getting higher grades than other students (section 7.2);
- The students have more difficulties in subjects related to physics, mathematics, algorithms, and programming at the introductory level (section 7.3);
- There is a high number of predicted failures. These cases occur due to many students having a lack of previous grades or a high number of negative grades (related to the limited dataset available for the present work).

From the analysis of the model's results in this chapter, we can conclude that with the availability of the students' grades it's possible to predict the students' outcome with an average of AUC of 82% (see Table 7). Further analysis indicates that the inclusion of additional background from the students would allow improvements of the models as the ones with more data are the ones that perform the better.

# 8 Conclusion

With the development of the solution and the conclusions taken from the analysis of the students' grades, we were able to: 1) Extract information from the dataset given by LEI's direction; 2) Extract insights from the data which can help the direction on the improvement of the program; 3) Set an architecture to enable the further extraction and improvement of the analysis of LEI's students.

These results were aligned with the initial objectives of this work and were presented to the user as a dashboard, built in Power BI. An API was built to enable the user to recompute the prediction models built in R and the results of both prediction models and students' most common paths were integrated into the dashboard with the support of a database. Both the API and databases were built and hosted locally. It's also important to refer that the user won't be able to run predictions of a student to a subject if there isn't at least one grade of this subject in the training dataset.

In the following section 8.1, we present possible improvements to the present solution.

## 8.1 Limitations and Future Work

### 8.1.1 Analysis of data

The analysis of the students' grades serves as a proof of concept for reports that can be created to help in the analysis of the grades of the students. From this type of reports, it's possible for the DEI's direction to detect which subjects the students have more difficulties and compare the grades year by year.

However, the development of this work didn't take into account two modules as, although important, they weren't inside the scope of this work. The modules are 1) user authentication and; 2) Data warehouse. These should be addressed in future work.

In the user authentication, the initial architecture doesn't have any component to log in as it's considered that it will be integrated into the existing infrastructure. This module is crucial to only give permission to the direction to access this data (or another type of users if the need appears).

Concerning the data warehouse, in the first iteration, the data analysis and modeling were expected to be done with the help of an excel file with the data of students' grades from 2013 to 2016. However, it's expected that in a later iteration, access will be given to ISEP's server to create a data warehouse and, in this way, automate the extraction and transformation of data (Figure 69).

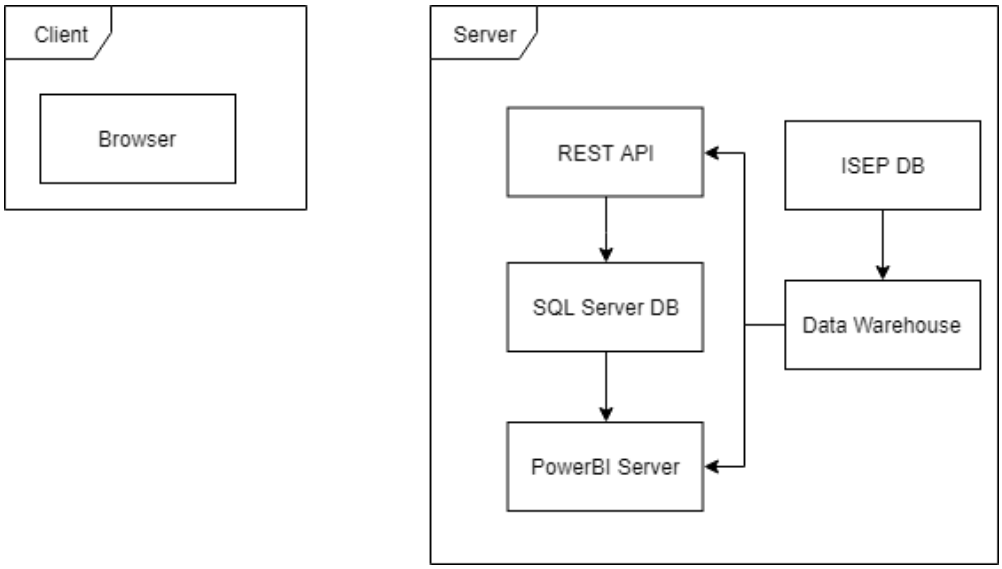



Figure 69 - Architecture without a data warehouse

In summary, the analysis can be further developed in the following areas:

- Integrate the data source with the production data or, if possible, with a data pipeline (i.e., data warehouse, cube or another multidimensional model);
- Integration with data after implementation of the Bologna' process;
- Development of reports after receiving feedback to improve the usefulness and accessibility of the data;
- Analysis of each subject individually so that the professors can use the data to improve their classes;
- Access to related data like students' attendance of each class (analyze if attendance is directly related to failure).

### 8.1.2 Evaluation of the dashboard

The dashboard proposed in the architecture section is meant to be used by the client to improve their decision-making process. However, in the present work no effort was done to evaluate whether the use dashboard provided such an improvement in the decision-making process. This should be addressed in future work.

As this type of decisions can't be measured in a short period of time, in a later date a form could be distributed so that it could be filled by the users after they've used the application. The questions will focus on its usefulness and how often have the users made decisions based on the dashboards. The answers must be analyzed using statistical methods such as to evaluate if the application has fulfilled its goals. Other types of evaluations should be done with automated

testing. These types of tests should focus on evaluation of the models to help improve them with the introduction of new data and alert when they fail to improve over the previous models.

### 8.1.3 Model Predictions

In terms of the Model predictions, the present work has focused on the prediction of students' outcome for each subject and performance analysis. However, the present proof-of-concept work aimed at producing a prototype with a very limited set of data and accordingly no effort was done to: 1) Integrate the data pipeline to the ISEP's infrastructure and; 2) deploy the structure to a shared server in order to automate the data pipeline and access to the API.

Therefore, after this initial development stage, the following improvements should be considered:

- To improve the models there can be further interaction with the parameters of each algorithm;
- The models only consider the grades of each student. The addition of background data like the students' high school grades and the program would improve the system, especially the first-year subjects;
- Implementation of a portal for access by the direction of the data analysis and prediction of the students' outcomes;
- Implementation of an alert system to identify students at high-risk of failing a subject
- Deployment of the database and API to a cloud server;
- Deployment of the R code to a server (or integrate inside the SQL Server);
- Development of a system to detect if the new models are better than the previous ones and prevent them from being deployed if they do not meet the minimum requirements.

# 9 References


[1] A. Watters, “How data and analytics can improve education - O’Reilly Media,” *O’Reilly*, 2011. [Online]. Available: https://www.oreilly.com/ideas/education-data-analytics-learning. [Accessed: 21-Jun-2018].

[2] “What is the CRISP-DM methodology?” [Online]. Available: https://www.sv-europe.com/crisp-dm-methodology/. [Accessed: 20-Nov-2017].

[3] P. Richard, D. R. Niewoehner, and L. Elder, *Engineering Reasoning*, 2nd ed. Foundation for Critical Thinking, 2013.

[4] C. Romero and S. Ventura, “Data mining in education,” *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.*, vol. 3, no. 1, pp. 12-27, Jan. 2013.

[5] J. Ferriman, “E-Learning Industry Worth $325 Billion by 2025 - LearnDash,” 2017. [Online]. Available: https://www.learndash.com/e-learning-industry-worth-325-billion-by-2025/. [Accessed: 09-Sep-2018].

[6] M. Credé, S. G. Roch, and U. M. Kieszczynka, “Class Attendance in College,” *Rev. Educ. Res.*, vol. 80, no. 2, pp. 272-295, Jun. 2010.

[7] T. Woodal, “Conceptualizing ‘Value for the customer: An Attributional ,Structural and Dispositional Analysis,” *Nottingham Trent Univ.*, 2003.

[8] P. Frow and A. Payne, “A stakeholder perspective of the value proposition concept,” *Eur. J. Mark.*, vol. 45, no. 1/2, pp. 223-240, 2011.

[9] R. S. Baker, “Educational Data Mining: An Advance for Intelligent Systems in Education,” *IEEE Intell. Syst.*, vol. 29, no. 3, pp. 78-82, May 2014.

[10] “Power BI | Ferramentas do BI de Visualização de Dados Interativos.” [Online]. Available: https://powerbi.microsoft.com/pt-pt/. [Accessed: 05-Oct-2018].

[11] R. Kimball and M. Ross, *The data warehouse toolkit : the definitive guide to dimensional modeling.* Wiley, 2013.

[12] “Drill mode in a visualization in Power BI - Power BI | Microsoft Docs.” [Online]. Available: https://docs.microsoft.com/en-us/power-bi/consumer/end-user-drill. [Accessed: 05-Oct-2018].

[13] M. Rouse, “What is Microsoft Power BI? - Definition from WhatIs.com,” *TechTarget*, 2015. [Online]. Available: http://searchcontentmanagement.techtarget.com/definition/Microsoft-Power-BI. [Accessed: 06-Feb-2018].

[14] R. Ahmed, “Tableau Tutorial | Step by Step Guide to Learn Tableau | Edureka,” *edureka*, 2017. [Online]. Available: https://www.edureka.co/blog/tableau-tutorial/?utm_campaign=what-is-tableau&utm_medium=content-link&utm_source=blog. [Accessed: 19-Feb-2018].

[15] “How To Group Objects Into Similar Categories, Cluster Analysis.” [Online]. Available:

http://www.statsoft.com/Textbook/Cluster-Analysis. [Accessed: 20-Feb-2018].

[16] S. Ray, "6 Easy Steps to Learn Naive Bayes Algorithm (with code in Python)," *Analytics Vidhya*, 2017. [Online]. Available: https://www.analyticsvidhya.com/blog/2017/09/naive-bayes-explained/. [Accessed: 25-Feb-2018].

[17] G. Piatetsky, "Top Data Science and Machine Learning Methods Used in 2017," *Knuggets*, 2017. [Online]. Available: https://www.kdnuggets.com/2017/12/top-data-science-machine-learning-methods.html. [Accessed: 07-Oct-2018].

[18] M. Kuhn, "The caret Package," 2018. [Online]. Available: http://topepo.github.io/caret/index.html. [Accessed: 07-Oct-2018].

[19] S. Sayad, "Association Rules." [Online]. Available: https://www.saedsayad.com/association_rules.htm. [Accessed: 05-Oct-2018].

[20] M. Hahsler, B. Grün, and K. Hornik, "**arules** - A Computational Environment for Mining Association Rules and Frequent Item Sets," *J. Stat. Softw.*, vol. 14, no. 15, 2005.

[21] C. Buchta, "cspade function | R Documentation." [Online]. Available: https://www.rdocumentation.org/packages/arulesSequences/versions/0.2-20/topics/cspade. [Accessed: 06-Oct-2018].

[22] F. Rodrigues, "Model Evaluation."

[23] J. D. Kelleher and B. Mac Namee, "Fundamentals of Machine Learning for Predictive Data Analytics: Algorithms ... - John D. Kelleher, Brian Mac Namee, Aoife D'Arcy - Google Books." [Online]. Available: https://books.google.pt/books?id=uZxOCgAAQBAJ&pg=PA414&lpg=PA414&dq=tnr+analytics&source=bl&ots=eotAs7yYAz&sig=2jsOvXbfHGhpcQTKL69SHc6PsBs&hl=en&sa=X&ved=0ahUKEwjbiJDNxqzZAhUKtxQKHUcGBwwQ6AEIOjAD#v=onepage&q=recall&f=false. [Accessed: 17-Feb-2018].

[24] D. M. Green and J. A. Swets, *Signal detection theory and psychophysics / David M. Green, John A. Swets. - Version details - Trove*, Illustrated Edition. New York, 1966.

[25] Y. Tang, "GitHub - rstudio/tensorflow: TensorFlow for R." [Online]. Available: https://github.com/rstudio/tensorflow. [Accessed: 07-Feb-2018].

[26] "Search · stars:>1 · GitHub." [Online]. Available: https://github.com/search?q=stars:%3E1&s=stars&type=Repositories. [Accessed: 07-Feb-2018].

[27] S. SHARMA, "TensorFlow on Mobile: TensorFlow Lite - Towards Data Science," *Medium*, 2018. [Online]. Available: https://towardsdatascience.com/tensorflow-on-mobile-tensorflow-lite-a5303eef77eb. [Accessed: 07-Feb-2018].

[28] "Keras Documentation." [Online]. Available: https://keras.io/. [Accessed: 07-Feb-2018].

[29] "Why use Keras - Keras Documentation." [Online]. Available: https://keras.io/why-use-keras/. [Accessed: 07-Feb-2018].

[30] “Machine Learning Project at the University of Waikato in New Zealand.” [Online]. Available: https://www.cs.waikato.ac.nz/ml/index.html. [Accessed: 05-Oct-2018].

[31] G. Grolemund and H. Wickham, *R for Data Science*, 1st ed. O’Reilly Media, 2017.

[32] K. E. Arnold and M. D. Pistilli, “Course signals at Purdue,” in *Proceedings of the 2nd International Conference on Learning Analytics and Knowledge - LAK ’12*, 2012, p. 267.

[33] G. Siemens and R. S. J. d. Baker, “Learning analytics and educational data mining,” in *Proceedings of the 2nd International Conference on Learning Analytics and Knowledge - LAK ’12*, 2012, p. 252.

[34] L. Calvet Liñán and Á. A. Juan Pérez, “Educational Data Mining and Learning Analytics: differences, similarities, and time evolution,” *RUSC. Univ. Knowl. Soc. J.*, vol. 12, no. 3, p. 98, Jul. 2015.

[35] J. A. Larusson and B. White, “Introduction,” in *Learning Analytics*, New York, NY: Springer New York, 2014, pp. 1-12.

[36] T. Daradoumis, R. Bassi, F. Xhafa, and S. Caballe, “A Review on Massive E-Learning (MOOC) Design, Delivery and Assessment,” in *2013 Eighth International Conference on P2P, Parallel, Grid, Cloud and Internet Computing*, 2013, pp. 208-213.

[37] P. Mukala, J. Buijs, M. Leemans, and W. Van Der Aalst, “Learning Analytics on Coursera Event Data: A Process Mining Approach.”

[38] “Top Business Intelligence Tools - 2017 Reviews & Pricing,” 2017. [Online]. Available: https://www.softwareadvice.com/bi/?v=2#top-products. [Accessed: 26-Nov-2017].

[39] G. Duncan, “Choosing the Right Analytics Tool,” 2017.

[40] Gregory Piatetsky, “Python vs R - Who Is Really Ahead in Data Science, Machine Learning?,” *Knuggets*, 2017. [Online]. Available: https://www.kdnuggets.com/2017/09/python-vs-r-data-science-machine-learning.html. [Accessed: 05-Feb-2018].

[41] B. Darrow, “Amazon, Microsoft, Google Still Lead Gartner Cloud Rankings | Fortune,” *Fortune*, 2017. [Online]. Available: http://fortune.com/2017/06/15/gartner-cloud-rankings/. [Accessed: 10-Feb-2018].

[42] K. Bouher, “Yes, you can run R in the cloud securely | R-bloggers,” *R-bloggers*, 2017. [Online]. Available: https://www.r-bloggers.com/yes-you-can-run-r-in-the-cloud-securely/. [Accessed: 10-Feb-2018].

[43] “plumber.” [Online]. Available: https://www.rplumber.io/. [Accessed: 05-Oct-2018].

# 10 Appendix

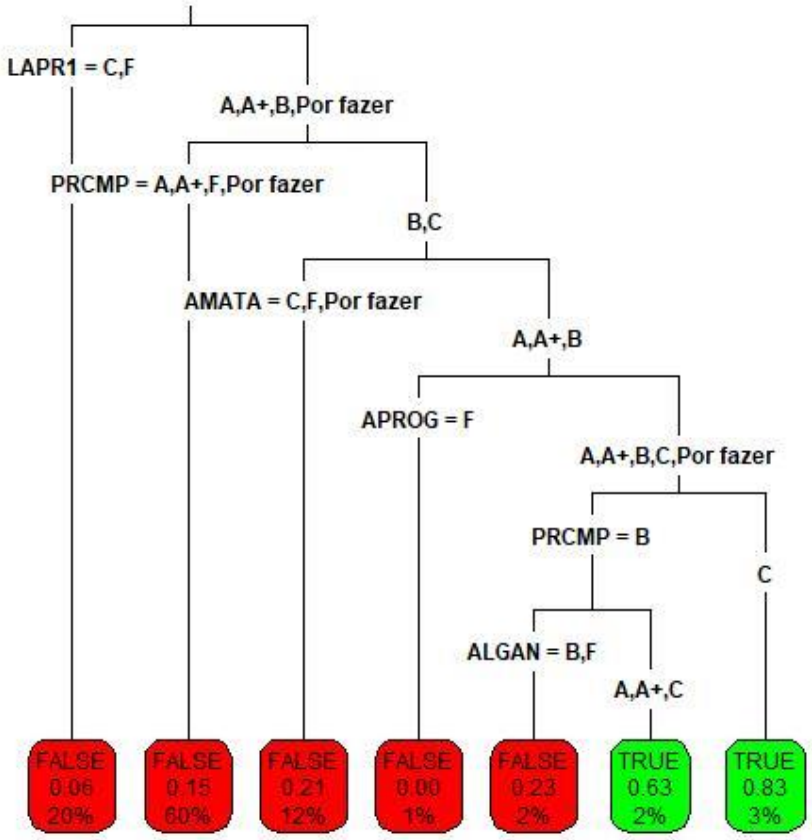



Figure 70 - Decision Tree ALGAV

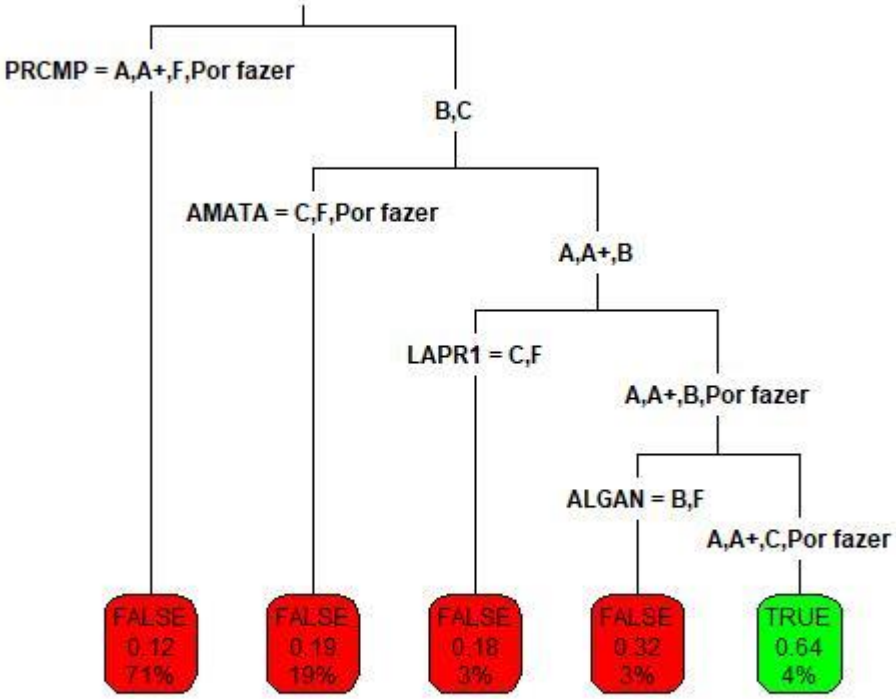



Figure 71 - Decision Tree PESTI

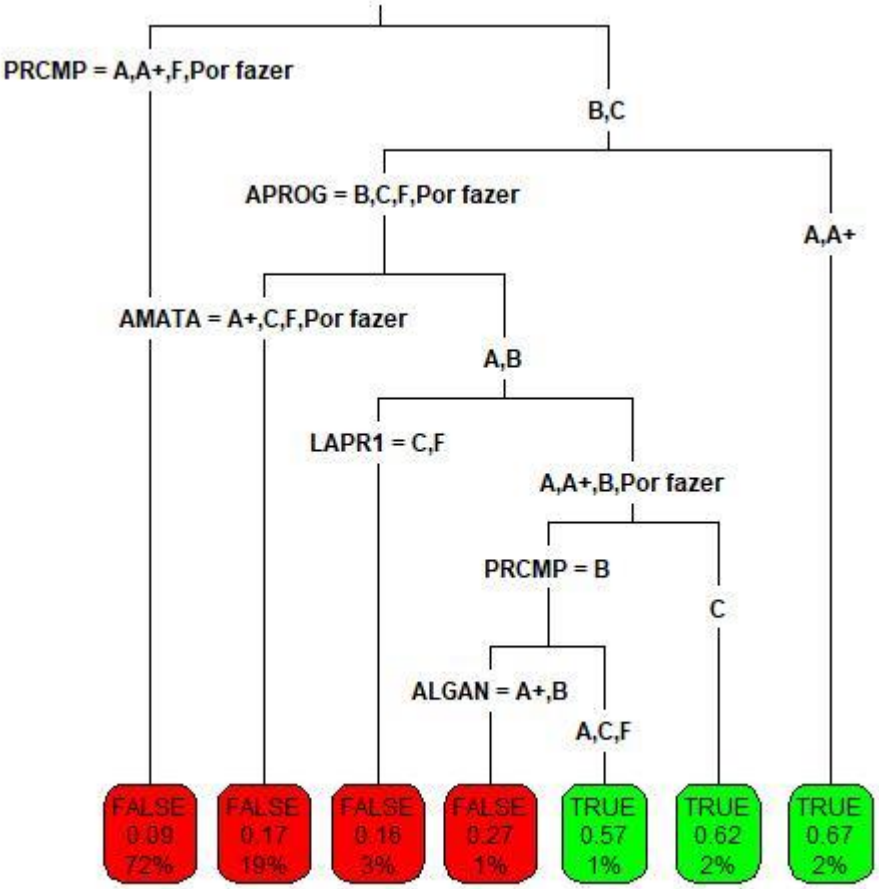



Figure 72 - Decision Tree INFOR

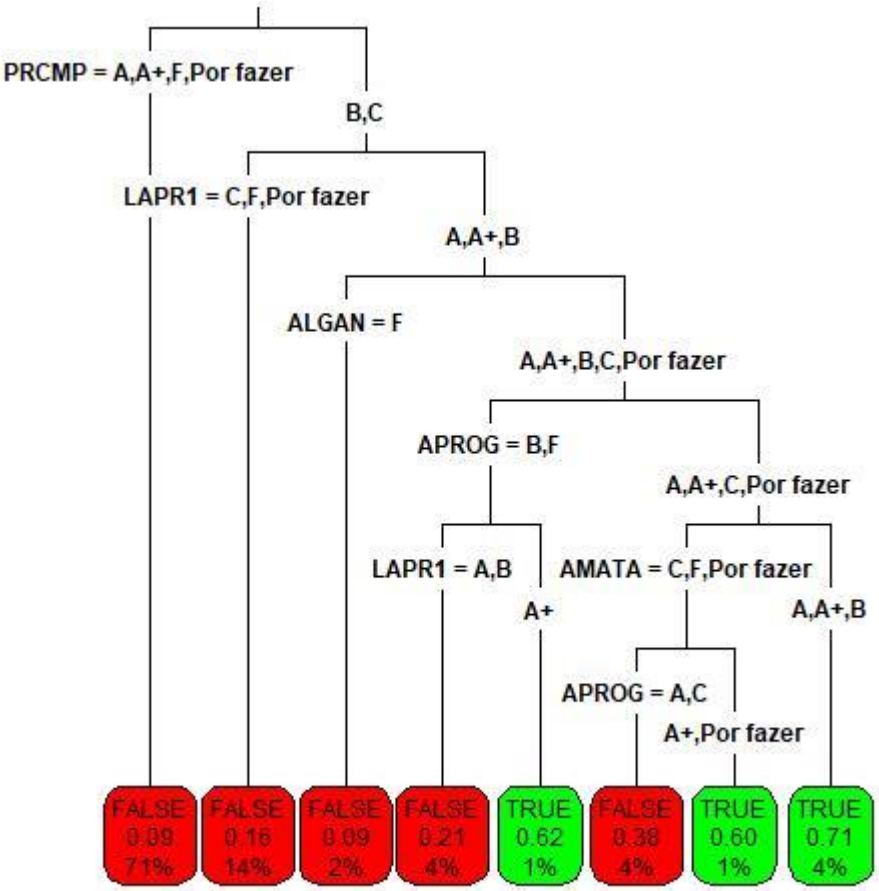



Figure 73 - Decision Tree CORGA

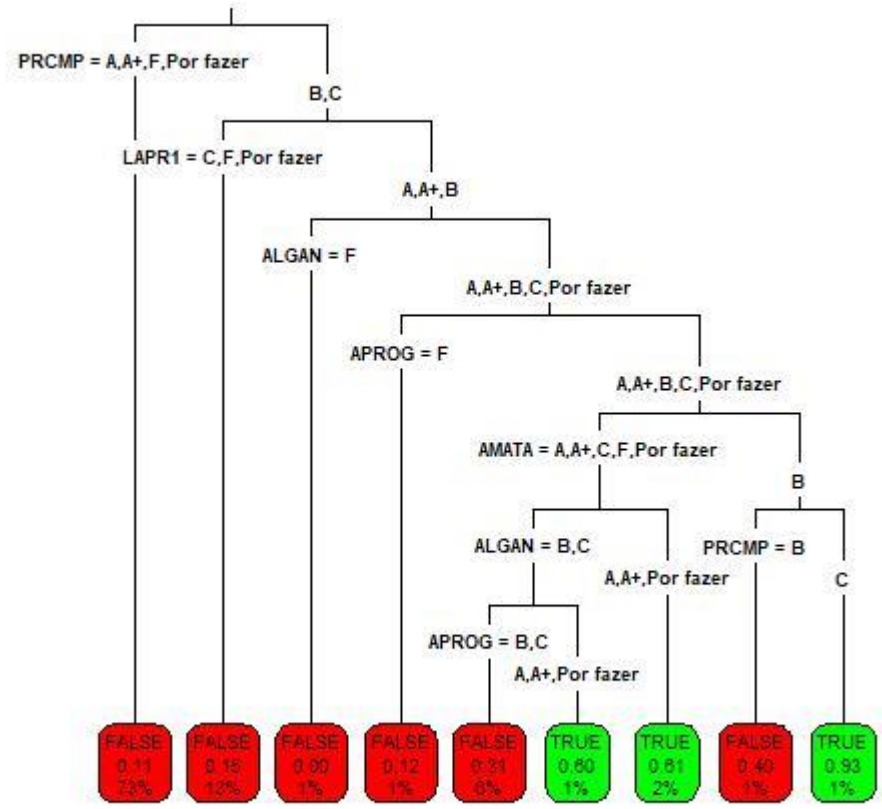



Figure 74 - Decision Tree ANADI

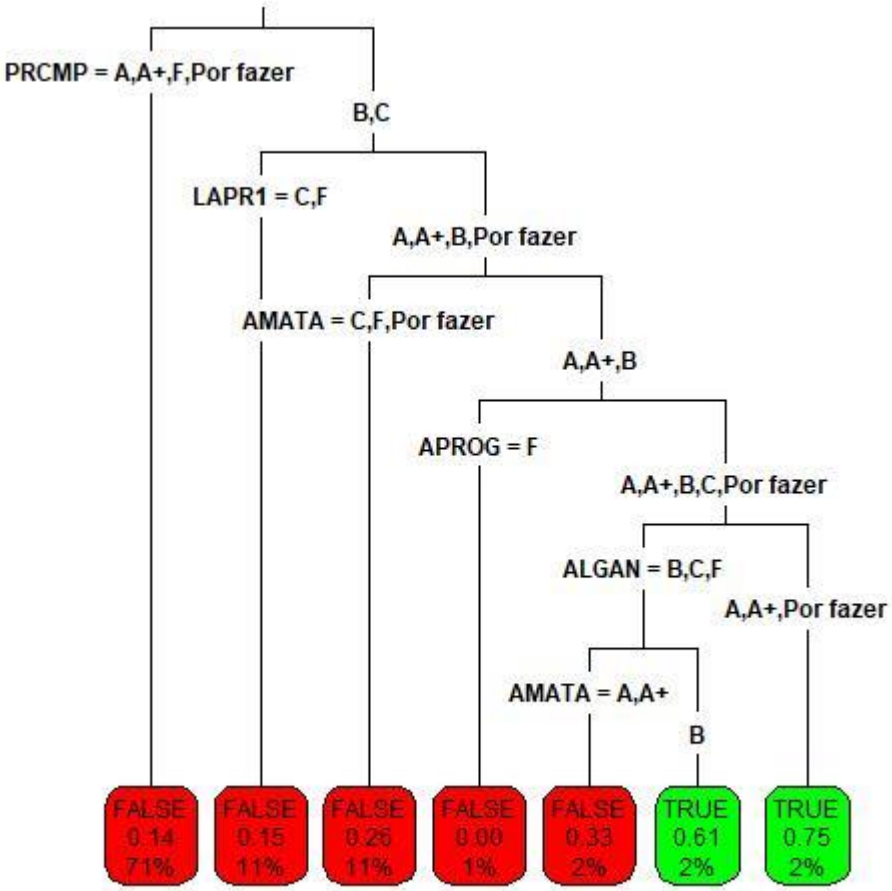



Figure 75 - Decision Tree SGRAI

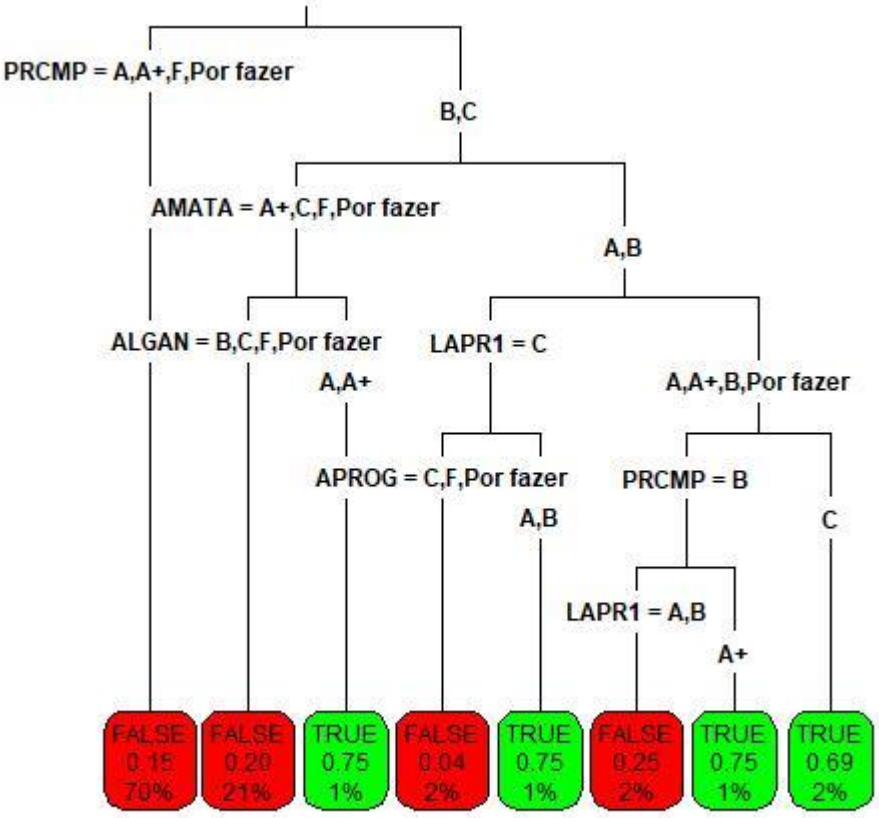



Figure 76 - Decision Tree LAPR5

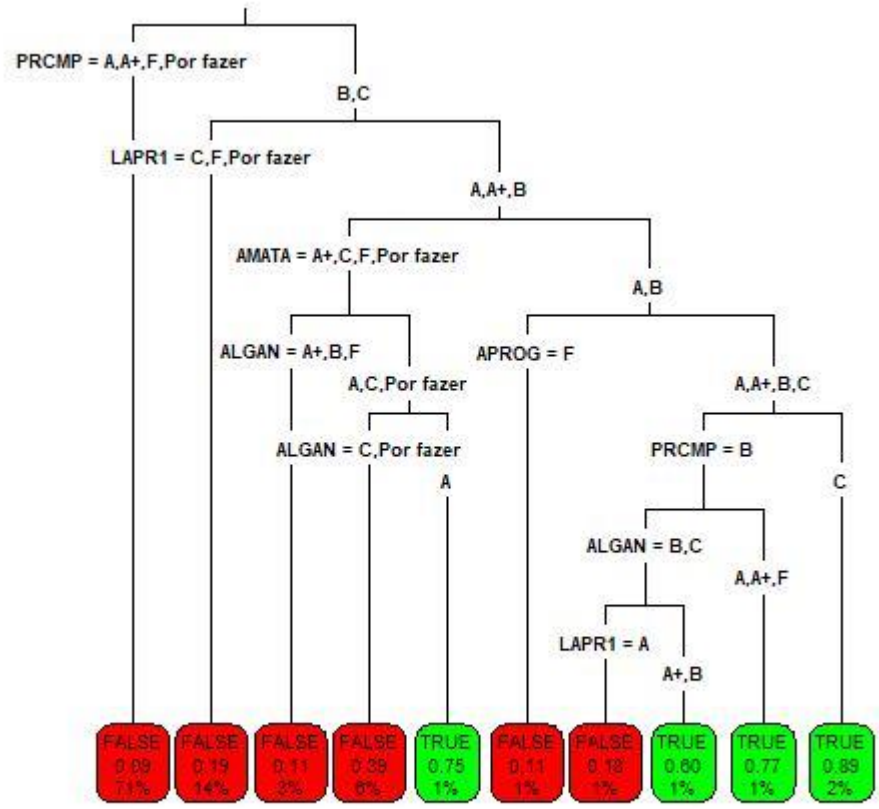



Figure 77 - Decision Tree GESTA

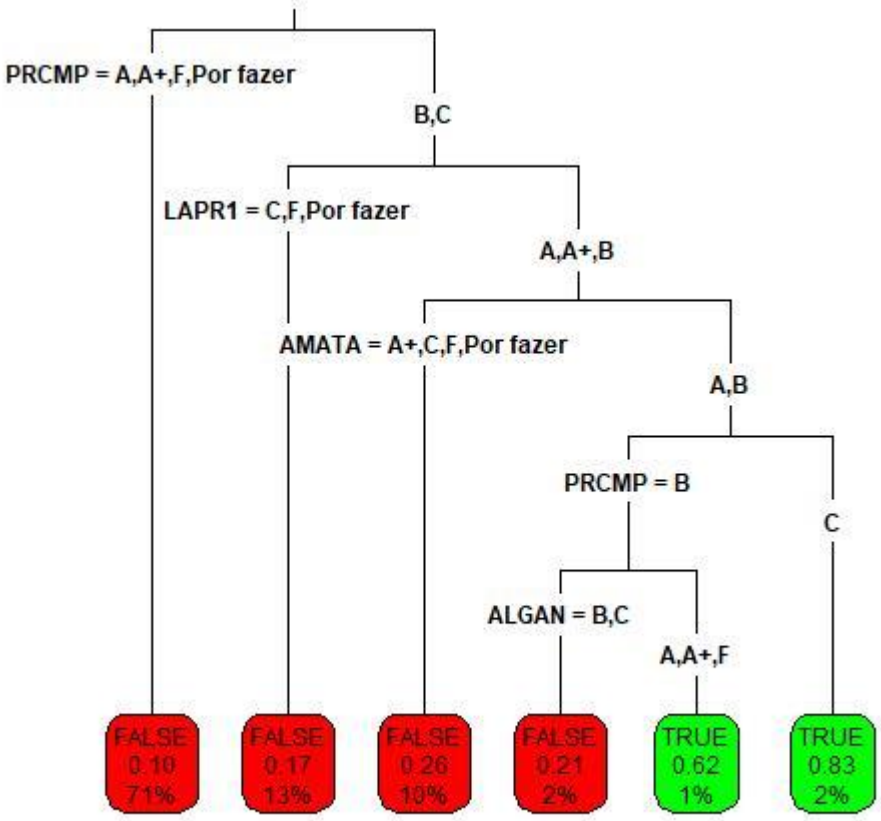



Figure 78 - Decision Tree ASIST

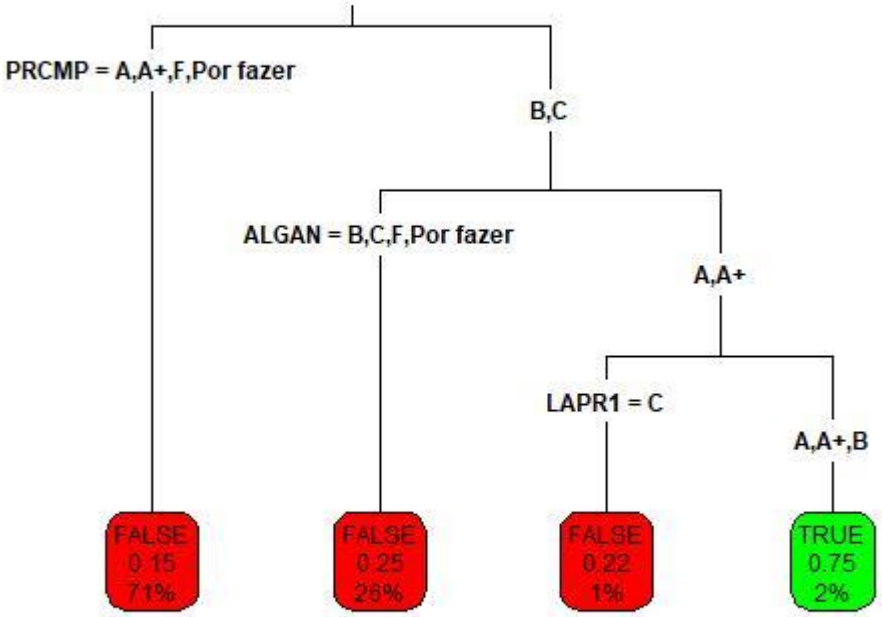



Figure 79 - Decision Tree ARQSI

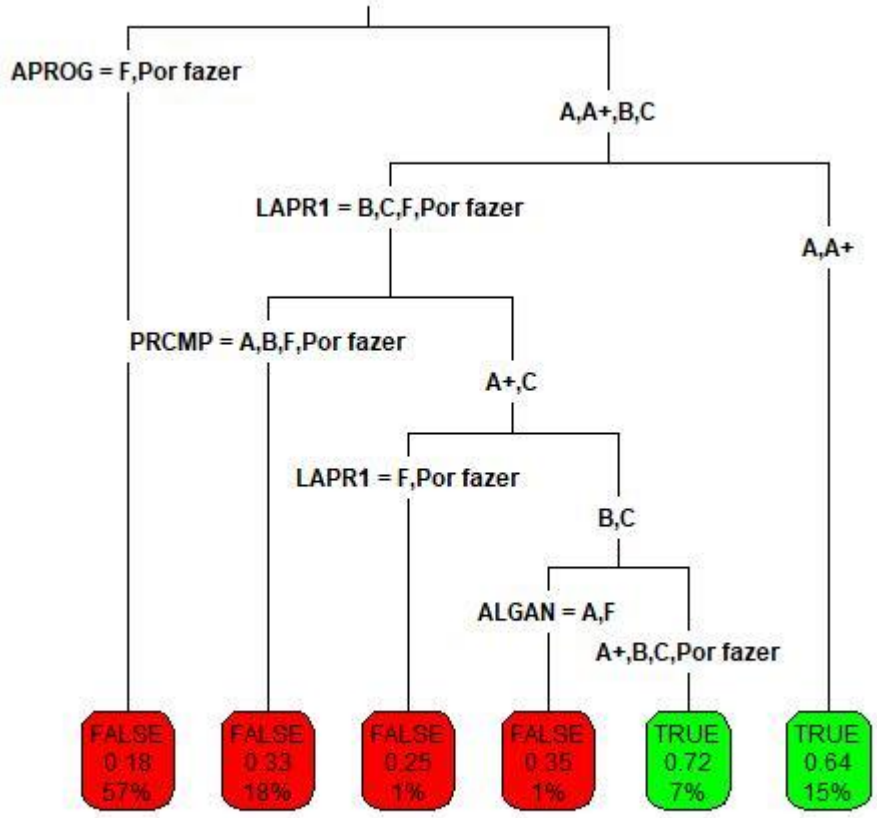



Figure 80 - Decision Tree SCOMP

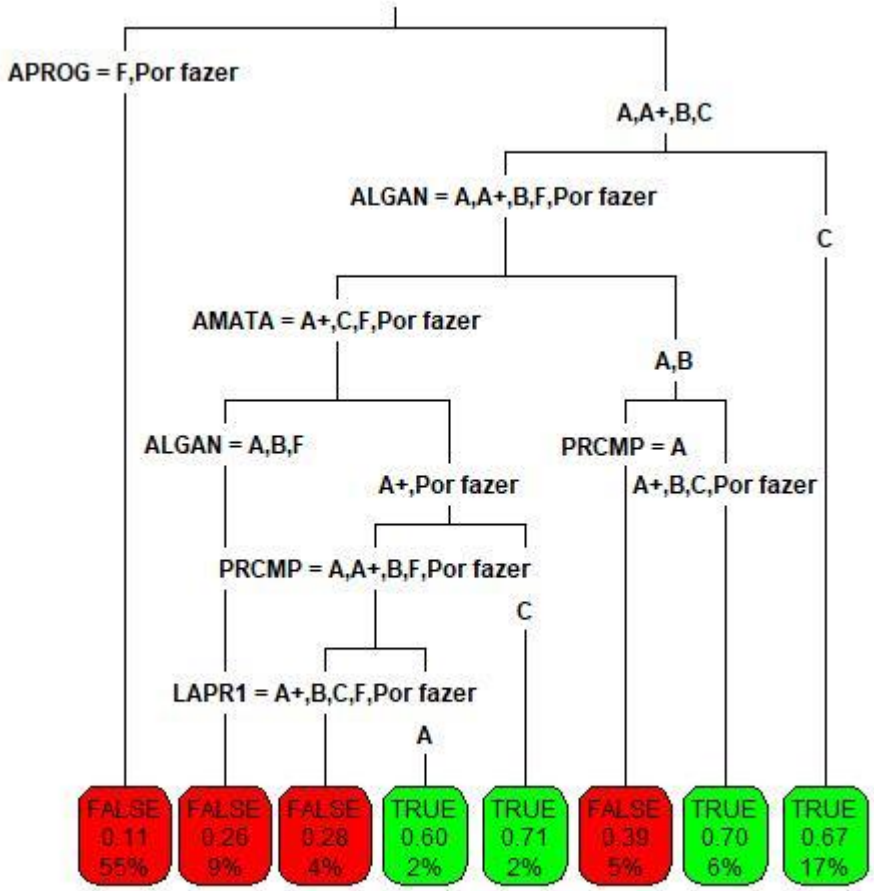



Figure 81 - Decision Tree RCOMP

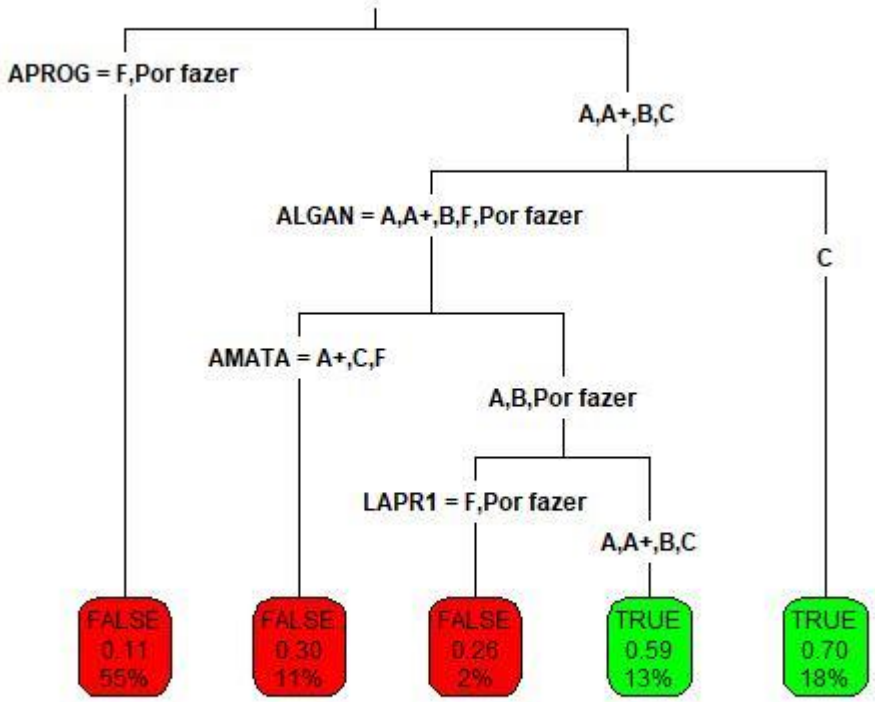



Figure 82 - Decision Tree LPROG

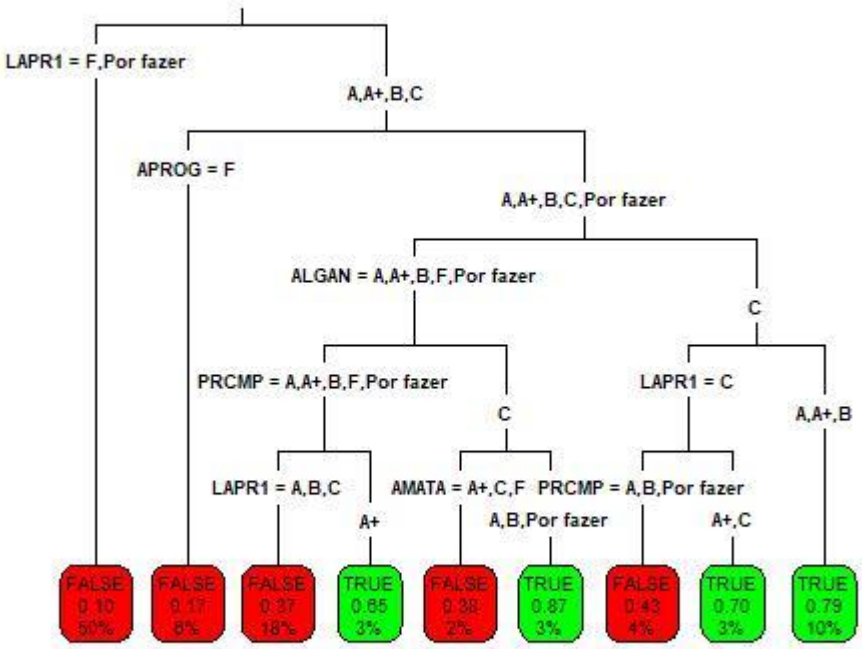



Figure 83 - Decision Tree LAPR4

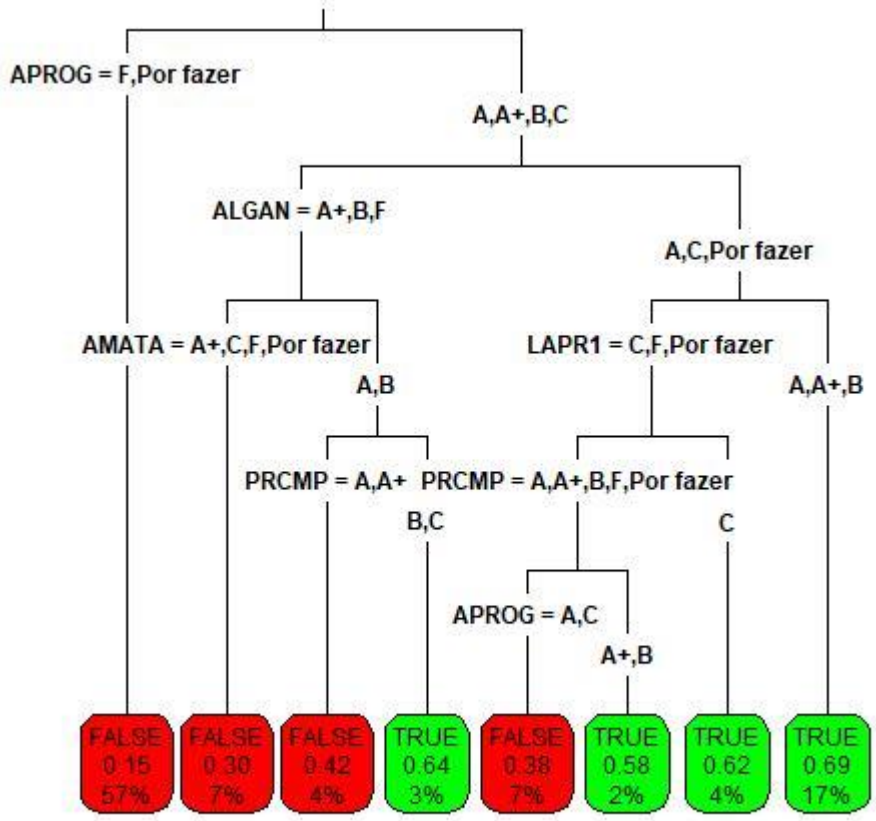



Figure 84 - Decision Tree EAPLI

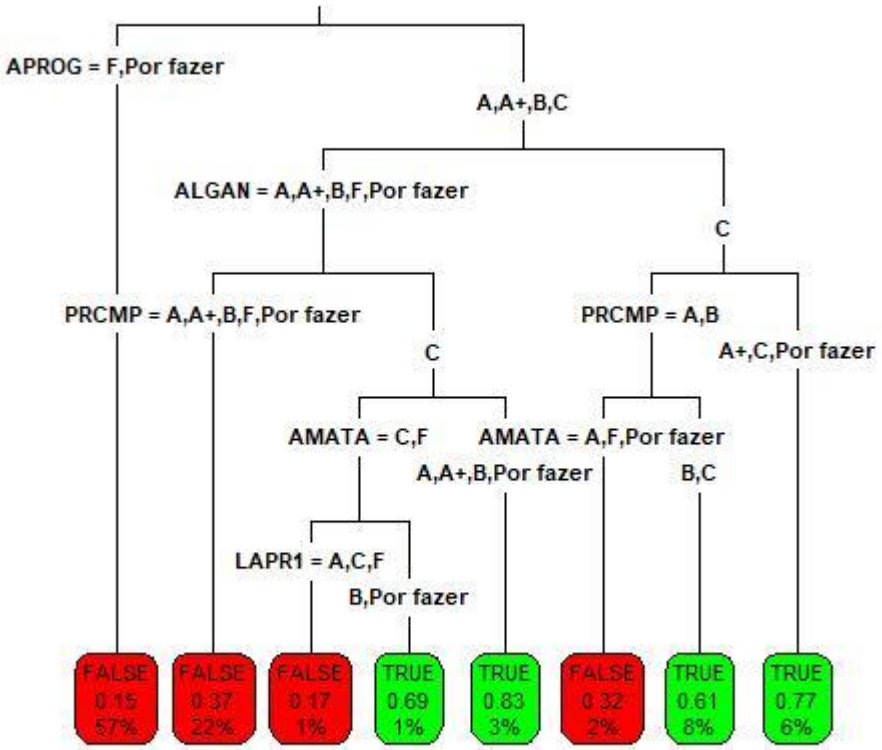



Figure 85 - Decision Tree LAPR3

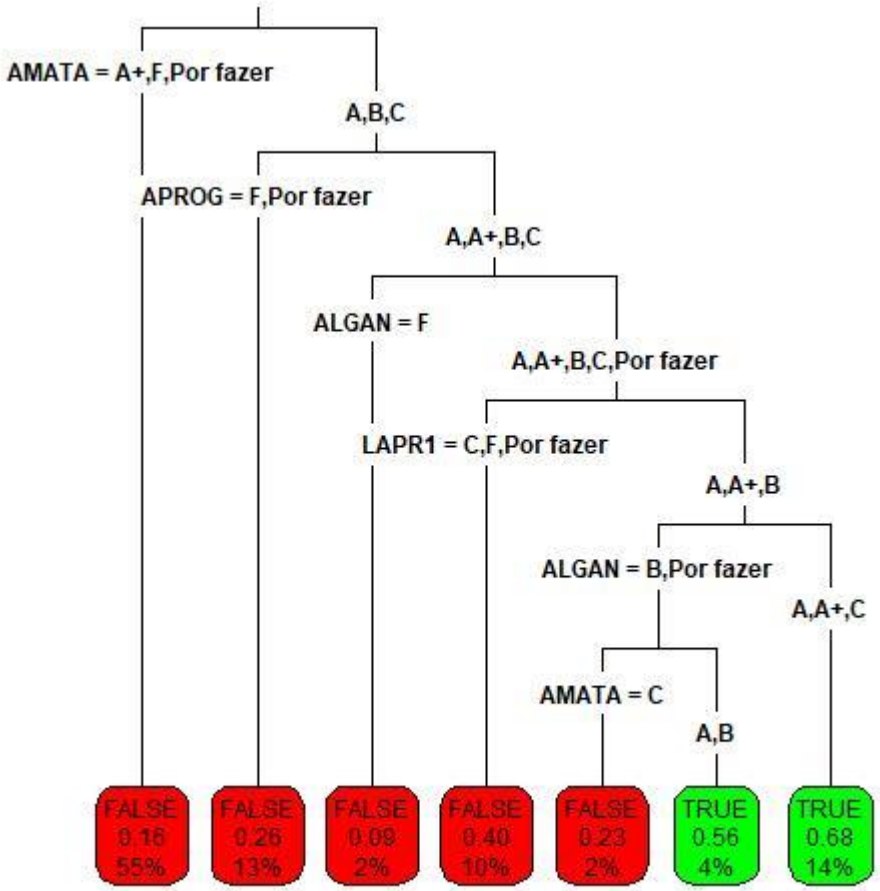



Figure 86 - Decision Tree FSIAP

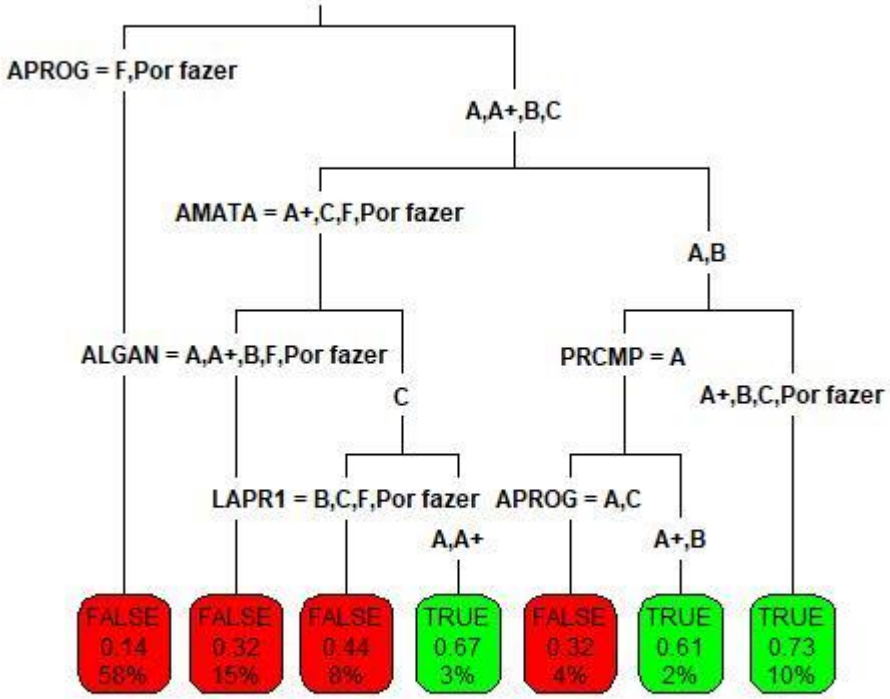



Figure 87 – ESINF

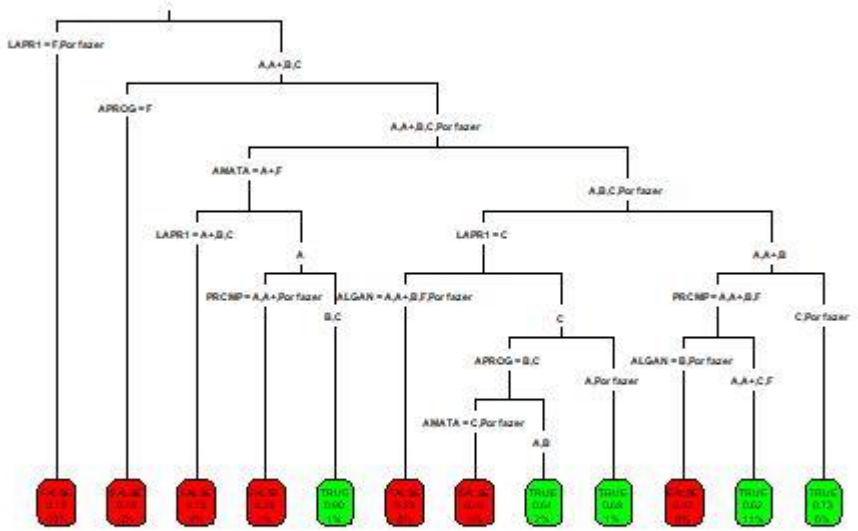



Figure 88 - Decision Tree BDDAD

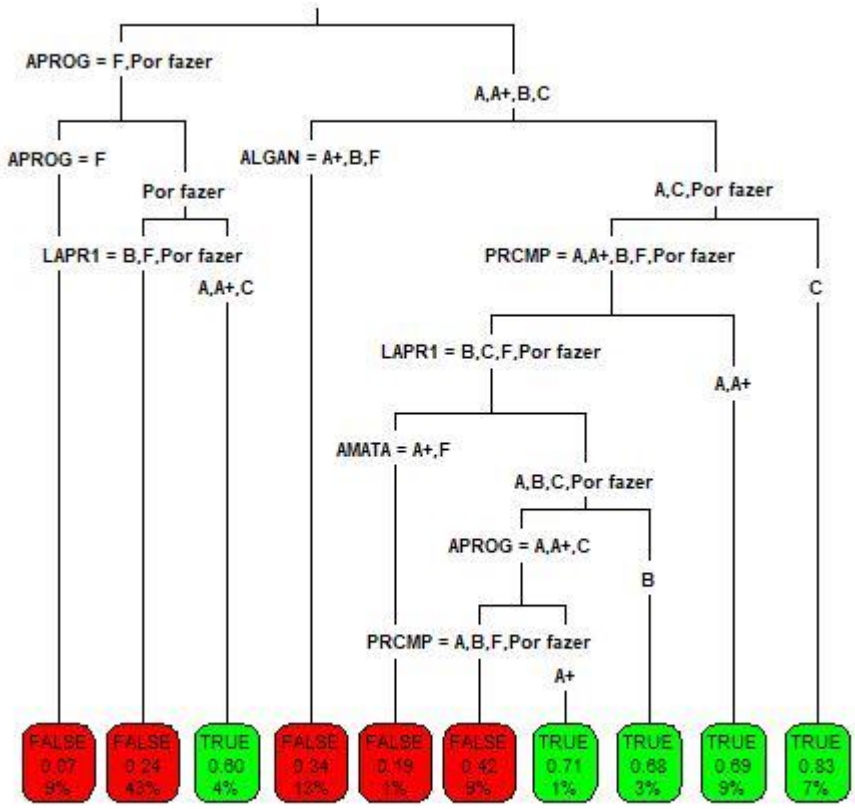



Figure 89 - Decision Tree ARQCP

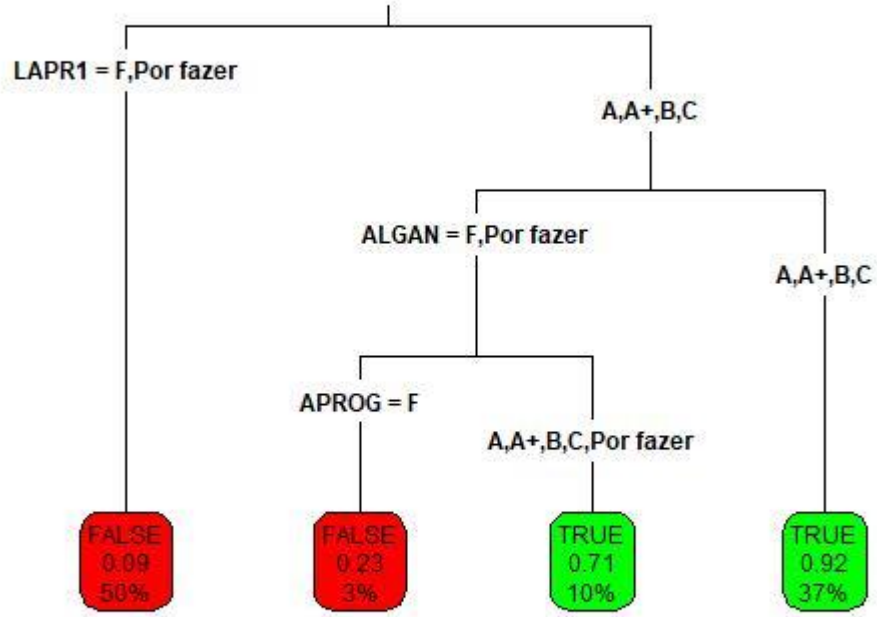



Figure 90 - Decision Tree PPROG


Figure 91 - Decision Tree MDISC

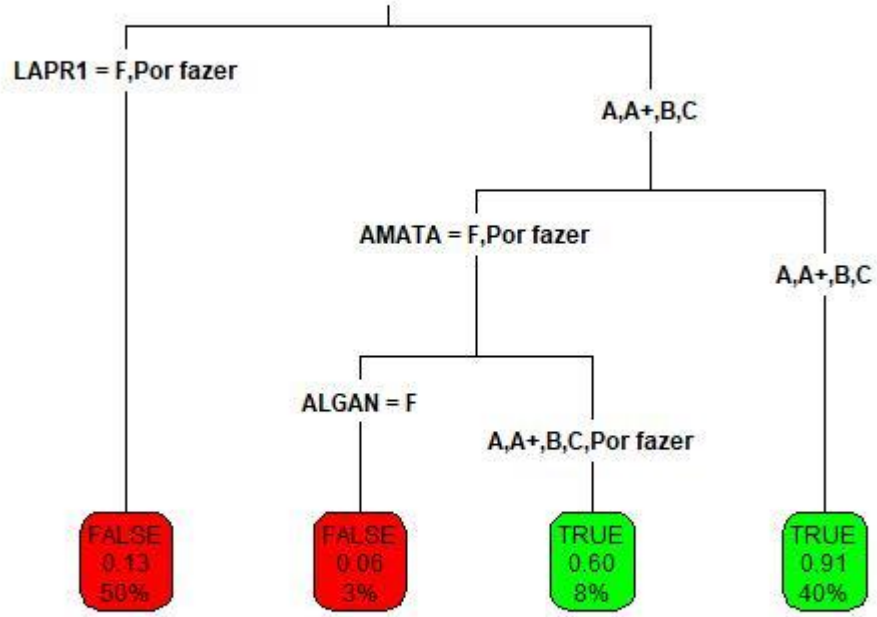



Figure 92 - Decision Tree MATCP


Figure 93 - Decision Tree LAPR2