UNIVERSIDAD DE BURGOS
ESCUELA POLITÉCNICA SUPERIOR
Grado en Ingeniería Informática

TFG del Grado en Ingeniería Informática

Simulador árboles de decisión Documentación Técnica

Presentado por Daniel Drefs Fernandes
en Universidad de Burgos — June 10, 2024
Tutores: Carlos López Nozal
Ismael Ramos Pérez

Contents

List of Figures

List of Tables

Appendix A

Software Project Plan

A.1 Introduction

This section presents how time management was handled in the course of this project. Throughout the development, bi-weekly meetings were held that served the purpose of reviewing what had been done and discussing what the tasks for the next sprint would be. To give better insight, a burndown of every sprint will be displayed. They are automatically created by Zube, the project agility tool that was used. The issues which these graphs are based on are all found in the "Issues" section[1] of the project's GitHub repository.

A.2 Time planning

Sprint 1 (29/02/2024 - 13/03/2024): Kick off project

Objectives: The main objectives of this sprint were to set up the Github repository structure, link it to Zube for a better overview of each sprint's tasks, learn about decision trees and to create a first web application displaying a tree using SVG.

Results: Almost all the tasks that were intended for this sprint were completed, except for the documentation of the Decision Trees concept in the Memoria.

Figure A.1 shows the burndown of the sprint.

[1] GitHub issues: https://github.com/danieldf01/TFG-decision-trees-sim/issues

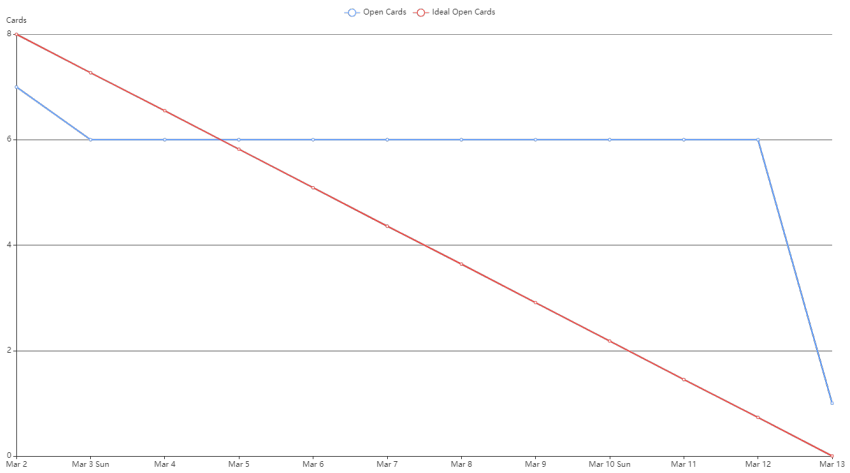

Figure A.1: Burndown Sprint 1

Sprint 2 (14/03/2024 - 03/04/2024): Implementation of tree graphics

Objectives: For this sprint, the intention was to create the first two prototypes, one displaying the entropy function with a calculator and the other one displaying a decision tree, both making use of the D3.js library. To display these prototypes, a GitHub Pages repository was to be created. Solidifying knowledge about conditional entropy and making entries to the "Theoretical concepts" section of the Memoria were also part of this sprint.

Results: As seen on the burndown in figure A.2, everything was completed except for the prototype displaying a decision tree. Due to sickness during the sprint, this task was left unfinished and pushed back to a later sprint for the time being.

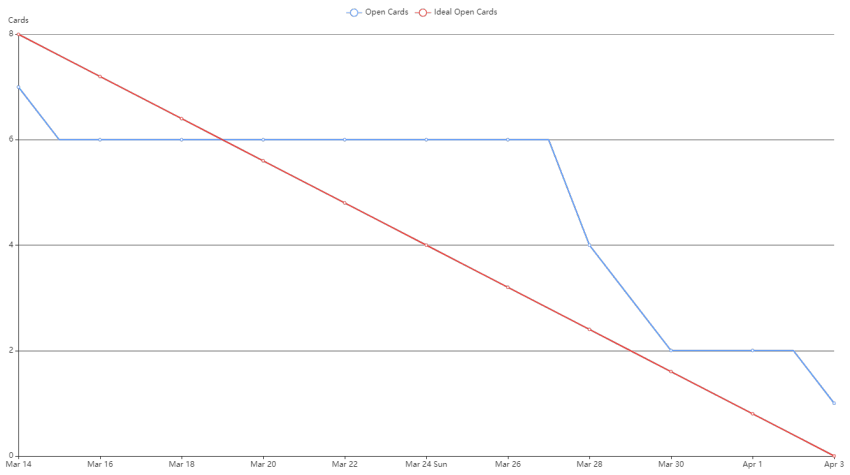

Figure A.2: Burndown Sprint 2

Sprint 3 (04/04/2024 - 17/04/2024): Prototype for conditional Entropy

Objectives: During this sprint, the main tasks were to refactor the GitHub repository structure, upgrade the visual presentation of the Entropy prototype using the Bootstrap framework, start documenting technical tools used in the Memoria and to create a prototype displaying a calculator for conditional Entropy.

Results: As figure A.3 shows, all the tasks of this sprint were completed in time.

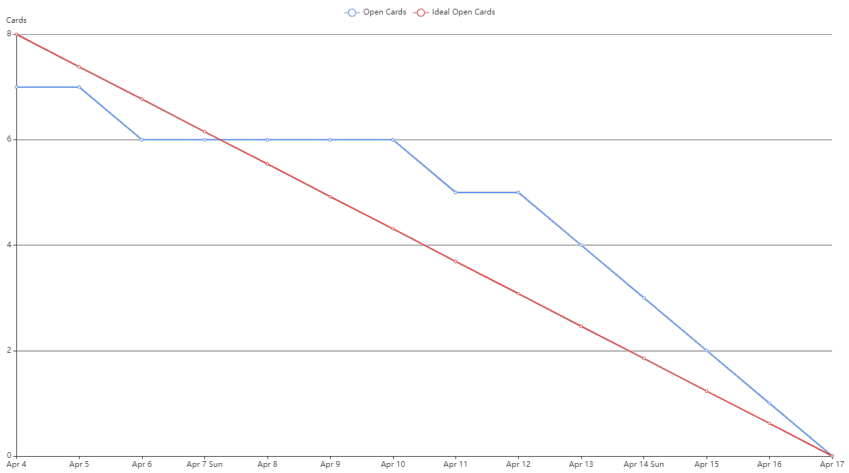

Figure A.3: Burndown Sprint 3

Sprint 4 (18/04/2024 - 02/05/2024): Prototype Decision Tree

Objectives: The main tasks of this sprint were to, on one hand, improve the existing prototypes with exceptions and enhance the overall code quality and, on the other hand, create a prototype that displays a decision tree based on an example dataset. Besides that, it was also asked to continue working on the Memoria by documenting some technical environments that were used.

Results: As seen in figure A.4, all tasks were completed except for two issues regarding the documentation of related works and a theoretical concept. This shortcoming was due to time constraints caused by assignments and exams in other classes.

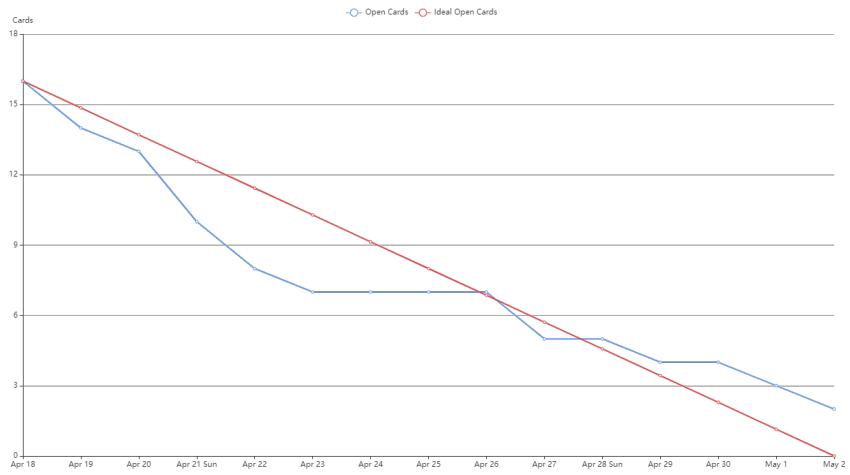

Figure A.4: Burndown Sprint 4

Sprint 5 (03/05/2025 - 16/05/2024): step-by-step Decision Tree simulation

Objectives: This sprint's main objective consisted of implementing a step-by-step visualization for the decision tree prototype that was created in the previous sprint. To achieve that, the decision tree creation had to be made dynamic, which, at the time, it was not. Other tasks included the creation of a header and footer for the web application and documenting relevant aspects of the development.

Results: Figure A.5 displays this sprint's burndown which shows that all the proposed tasks were done in time.

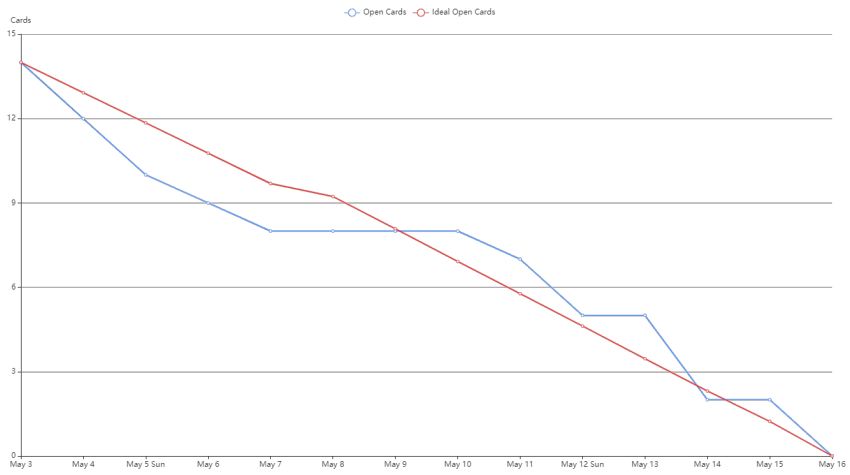

Figure A.5: Burndown Sprint 5

Sprint 6 (17/05/2024 - 30/05/2024): Decision Tree value table, CSV data loading, interactive data

Objectives: One of this sprint's main goals was to upgrade the decision tree's prototype by adding a dynamic value table that would display relevant values, like each feature's information gain, at each step. The other main objectives were to make it possible for the user to use their own datasets in CSV file format and to allow them to add and remove rows and columns from a currently loaded dataset.

Results: Figure A.6 shows that, due to the sprint having been during the final exam phase, not all tasks were completed. Besides issues like scaling text sizes based on their width and a cleanup of the project layout, one of the main objectives was left unfinished. While the addition of user-uploaded CSV datasets was successful, the "interactive data" goal was not met. In the end, it was discarded altogether as other refinements took priority due to the lack of time.

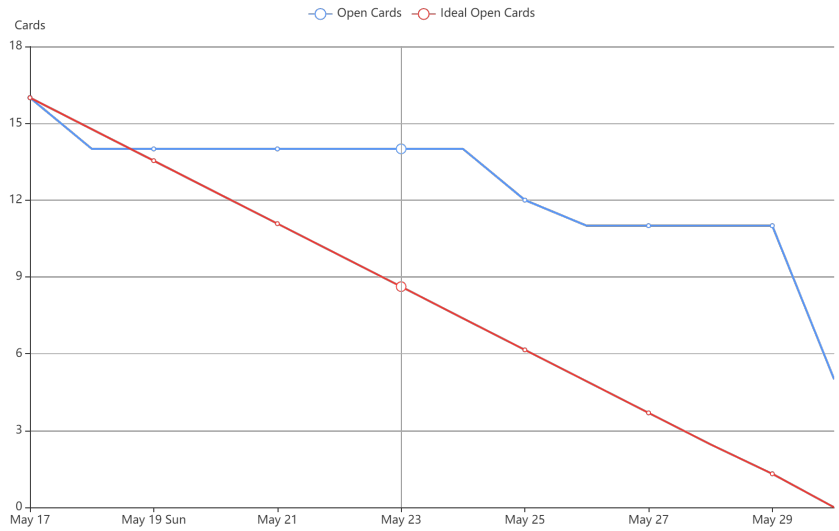

Figure A.6: Burndown Sprint 6

Sprint 7 (31/05/2024 - 06/06/2024): Decision Tree selectable example data, CSV file requirements

Objectives: The final sprint of this project's development was used to refine some of the already existing parts of the application. One issue was to add the functionality of being able to choose between different example datasets for the decision tree ID3 simulation. Another was to formulate requirements that a user-chosen CSV dataset had to meet and display them.

Results: Figure A.7 shows a burndown of the final sprint. As this sprint still took place during the exam phase, not all tasks could be finished here either. However, those were only minor issues like an improvement of the repository's README file which could were completed in the final days before the deadline.

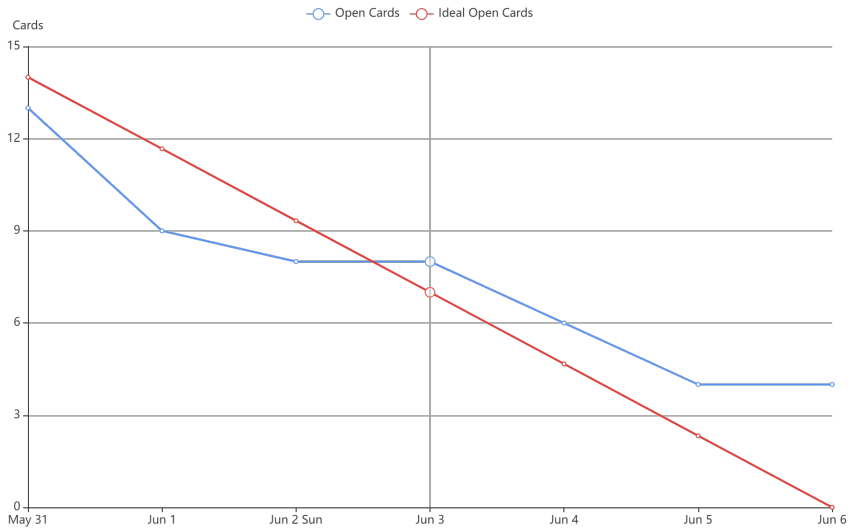

Figure A.7: Burndown Sprint 7

Overview

Figure A.8 displays an overview of the amount of issues that were resolved during each sprint.

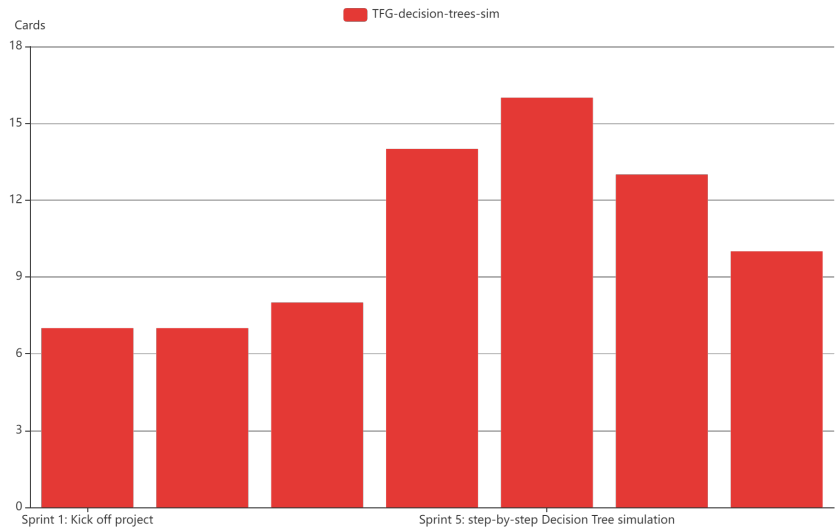

Figure A.8: Sprints overview

The workload was visibly increased starting at sprint 4 (A.2), which marked the start of the implementation of the decision tree prototype. Before that point, a lot of the sprints' work included learning and getting used to new technologies in order to be able to use them. Most of this process was not captured in the form of issues.

A.3 Viability study

Economic viability

Staff costs

This project has been carried out by one programmer over the course of approximately 3 months. Taking into consideration the developer's status of full-time student who also had to spend time in preparation for other classes, the overall development time can be estimated to a part-time employment. Considering the monthly minimum wage for general workers in Spain in 2024 of 1134,00€ per month [2], the estimated salary would be:

$$3m * 1134€/m = 3402€$$

As this represents only the gross wage, social security contributions have to be added, as well. As of 2024, these make up 36,85%, with the employee having to contribute 6,45% and the employer 30,4% [1]. Adding this to the gross wage totals up to:

$$3402€ + 3402€ * 0,3685 = 4655,637€$$

Software and hardware costs

The software cost of this project is equal to 0 as only free software has been used. As for hardware, a device with a value of 863,27€ has been used to carry out this project. Depreciation does not have to be considered as the device was bought at the beginning of development. Having used this device for the project over the course of 3 months, it makes up for a total cost of:

$$3m * 863,27€/m = 2589,81€$$

Total cost

Considering staff and hardware costs together, the total cost is summed up to:

$$2589,81€ + 4655,637€ = 7245,447€$$

Legal viability

Table A.1 shows every used dependency, its version and license

Dependency	Version	License
jQuery	3.7.1	MIT
jest	29.7.0	MIT
jest-environment-jsdom	29.7.0	MIT
D3	7.9.0	ISC
Bootstrap	5.3.3	MIT
bootstrap-icons	1.11.3	MIT
MathJax	3.2.2	Apache-2.0
Polyfill service	3.25.3	CC0-1.0
@popperjs/core	2.11.8	MIT
papaparse	5.4.1	MIT

Table A.1: Every dependency used in the project and its license

The most restrictive of these licenses would be the Apache-2.0 license, which is still a permissive license that allows, e.g., the software being used, modified and distributed in a commercial context.

Appendix B

Requirements Specification

B.1 Introduction

This section will explain the requirements of the application by specifying (non-)functional requirements and use cases.

B.2 General objectives

The main objective of this project has been to create a web application under the name of "Decision Tree Simulator" and with the purpose of helping users learn the concept of decision trees, how they are created and all necessary surrounding topics in an intuitive and simple way.

It provides dynamic calculators for entropy and conditional entropy which let the user input values to they can observe how different values affect the results. There is also a visual representation of the binary entropy graph that uses SVG and responds with markers to the user's input, if they used two classes to calculate the entropy.

The decision tree ID3 simulation presents a step-by-step visualization of the ID3 algorithm so that each user can follow the steps at their own pace. They can choose between selecting one of the example datasets or loading their own dataset in CSV file format. With a combination of a decision tree that is dynamically created using SVG, a dataset table, a value table, and visual cues at each step, the goal was to make the user's learning experience simple and intuitive.

B.3 Requirements catalog

Functional Requirements

- **FR-1** From the web, it must be possible to run the Entropy calculator for calculating the entropy of given input values
  - **FR-1.1** The user must be able to enter values into the presented input fields which are positioned in the column that is given the name “Nr. of instances” by the respective column header.
  - **FR-1.2** The user must be able to add classes by clicking on the button labeled “+”.
  - **FR-1.3** The user must be able to remove the row that represents the class that was last added by clicking on the button labeled “-”.
  - **FR-1.4** The user must be able to initialize the calculation of the entropy by clicking on the button labeled “Calculate Entropy”.
  - **FR-1.5** The application must, given valid input values, correctly calculate each class’s p-value and the feature’s entropy and display those values on the corresponding Entropy table.
  - **FR-1.6** The application must, if only 2 classes were used, show the results of the entropy calculation through a red dot on the x-axis of the presented coordinate system and a red line pointing to the corresponding point on the presented Binary Entropy graph.

- **FR-2** From the web, it must be possible to run the Conditional Entropy calculator for calculating the conditional entropy of given input values.
  - **FR-2.1** The user must be able to enter values into the presented input fields which are positioned in the columns that are given the name “Class 1" and “Class 2" by the respective column headers.
  - **FR-2.2** The user must be able to add categories by clicking on the button labeled “+”.
  - **FR-2.3** The user must be able to remove the row that represents the category that was last added by clicking on the button labeled “-”.

  - **FR-2.4** The user must be able to initialize the calculation of the conditional entropy by clicking on the button labeled “Calculate Conditional Entropy”.
  - **FR-2.5** The application must, given valid input values, correctly calculate each category’s ratio, entropy, and the feature’s conditional entropy and display those values on the table.

- **FR-3** From the web, it must be possible to run the Decision Tree ID3 simulator for executing a step-by-step simulation of the ID3 algorithm.
  - **FR-3.1** The user must be able to choose a dataset from one of the example datasets that are provided by the web application.
  - **FR-3.2** Given that the button labeled "Load own CSV dataset" was clicked, the application must present a modal to the user in which the requirements for a CSV file and an input form are displayed.
  - **FR-3.3** The user must be able to select their own dataset in a CSV file format through an input form.
  - **FR-3.4** The application must, given a valid CSV file, load the dataset that is contained in the file.
  - **FR-3.5** The application must, following a successful load of a dataset, display an information card in regards to the chosen dataset, the root node of the dynamically created decision tree, a data table presenting the dataset, and a value table that presents values that are relevant to the decision tree’s creation at each step.
  - **FR-3.6** The user must be able to navigate through the step-by-step simulation with the use of the four buttons that represent the four functions “Initial step”, “Step back”, “Step forward”, and “Last step”, respectively.
  - **FR-3.7** The application must, given that the “Initial step” button was clicked by the user, go to the first step of the simulation.
  - **FR-3.8** The application must, given that the “Step back” button was clicked by the user and the simulation had not already been at the first step, go back one step in the simulation.
  - **FR-3.9** The application must, given that the “Step forward” button was clicked by the user and the simulation had not already been at the last step, go forward one step in the simulation.

  - **FR-3.10** The application must, given that the "Last step" button was clicked by the user, go to the last step of the simulation.

Non-functional requirements

- **NFR-1** The user interface must be simple and intuitive.
- **NFR-2** The application must be responsive to different screen sizes.
- **NFR-3** For the Entropy calculator and Conditional Entropy calculator, the application must recognize any positive integer value as valid input.
  - **NFR-3.1** The user must be warned through appearing alerts if any of the user-made inputs is invalid.
- **NFR-4** For the Entropy calculator, if more than 2 classes are used, the user must be informed through an appearing alert about the fact that the calculated results will not be displayed on the Binary Entropy graph.
- **NFR-5** For the Decision Tree ID3 simulator, the application must recognize CSV files that meet the file requirements that are displayed in the application as valid.
  - **NFR-5.1** The user must be warned through an appearing alert if the proposed CSV file fails to meet any of the requirements and is therefore recognized as invalid.

B.4 Requirements specification

Use case diagram

Figure B.1 displays the use case diagram.

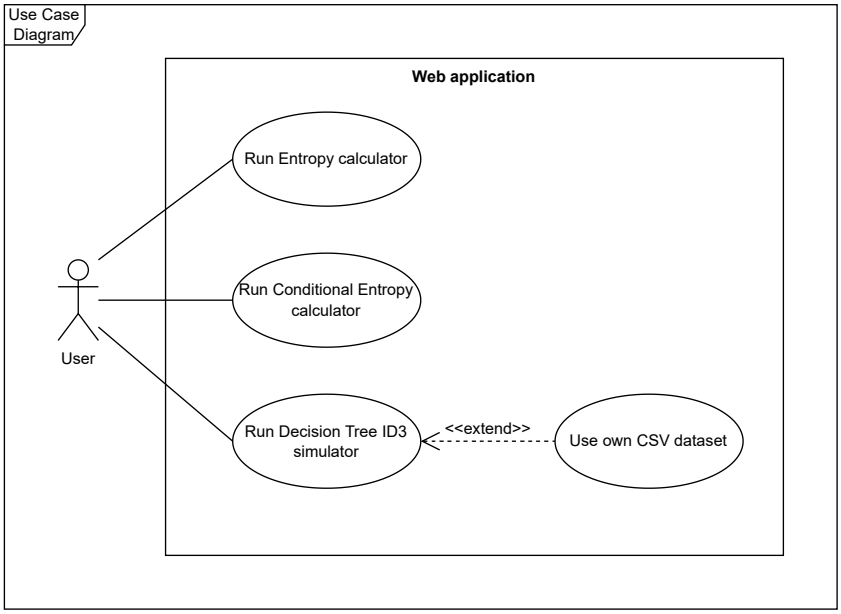

Figure B.1: Use case diagram

Use cases

The textual use cases are displayed in tables B.1, B.2, B.3, and B.4.

UC-1	**Run Entropy calculator**
Version	1.0
Author	Daniel Drefs Fernandes
Associated requirements	FR-1, FR-1.1, FR-1.2, FR-1.3, FR-1.4, FR-1.5, FR-1.6
Description	The user runs the Entropy calculator with the desired input values and receives the results in a visual format on the website.
Precondition	The input values introduced by the user are valid.
Actions	1. The user opens the application. a) The user adds one or multiple class by clicking the button with the label "+". 2. The user fills the input fields with the desired values and clicks on the button with the label "Calculate Entropy". 3. The application calculates each class's p-value and the feature's entropy and displays the results on the corresponding table. a) If the user has not added any classes, the application will show a visualization of the calculated results in SVG format on the Binary Entropy graph.
Postcondition	The results are displayed on the Entropy table.
Exceptions	If the user has introduced invalid values, the application will display an alert and inform the user to only use positive integer values.
Importance	High

Table B.1: UC-1 Run Entropy calculator.

UC-2	**Run Conditional Entropy calculator**
Version	1.0
Author	Daniel Drefs Fernandes
Associated requirements	FR-2, FR-2.1, FR-2.2, FR-2.3, FR-2.4, FR-2.5
Description	The user runs the Conditional Entropy calculator with the desired input values and receives the results in a visual format on the website.
Precondition	The input values introduced by the user are valid.
Actions	1. The user opens the application. a) The user adds one or multiple categories by clicking the button with the label "+". 2. The user fills the input fields with the desired values and clicks on the button with the label "Calculate Conditional Entropy". 3. The application calculates each category's ratio, entropy, and the feature's conditional entropy and displays the results on the table.
Postcondition	The results are displayed on the table.
Exceptions	If the user has introduced invalid values, the application will display an alert and inform the user to only use positive integer values.
Importance	High

Table B.2: UC-2 Run Conditional Entropy calculator.

UC-3	**Run Decision Tree ID3 simulator**
Version	1.0
Author	Daniel Drefs Fernandes
Associated requirements	FR-3, FR-3.1, FR-3.4, FR-3.5, FR-3.6, FR-3.7, FR-3.8, FR-3.9, FR-3.10
Description	The user runs the Decision Tree ID3 simulator with the desired dataset, receives the results in a visual format on the website and goes through the step-by-step simulation.
Precondition	None
Actions	1. The user opens the application. 2. The user clicks on the dropdown element labeled "Choose example dataset". 3. The user chooses a dataset from one of the example datasets that are provided by the application. 4. The application loads the dataset. 5. The application displays an information card designated for the dataset, the root node of the decision tree, a data table corresponding to the dataset, and the value table. 6. The user uses the four presented buttons to navigate through the step-by-step simulation.
Postcondition	The decision tree, data table, and value table are displayed at the user's desired step of the simulation.
Exceptions	None
Importance	High

Table B.3: UC-3 Run Decision Tree ID3 simulator.

UC-4	**Use own CSV dataset**
Version	1.0
Author	Daniel Drefs Fernandes
Associated requirements	FR-3, FR-3.2 FR-3.3, FR-3.4
Description	Within the Decision Tree ID3 simulator, the user selects their own CSV dataset that the application loads.
Precondition	The user has opened the application.
Actions	1. The user clicks on the button that is labeled "Load own CSV dataset". 2. The application displays a modal in which the user is presented with the file requirements and an input form. a) The user reads the file requirements. 3. The user clicks on the input form and selects the desired CSV file from their file system. 4. If given a valid CSV file, the application closes the modal and loads the dataset.
Postcondition	The user's desired dataset is loaded.
Exceptions	If the user has introduced an invalid CSV file, the application will display an alert that informs the user which requirement was not met and tells them to check the file requirements.
Importance	Low

Table B.4: UC-4 Use own CSV dataset.

Appendix C

Especificación de diseño

C.1 Introducción

C.2 Data design

Data storage

For the decision tree ID3 simulator, each example dataset is stored in a separate CSV file in a directory designated to the example data. When a certain dataset is selected by a user, a series of functions are triggered that load that set to be ready to use for the step-by-step simulation. The detailed process will be discussed in C.3.

When a user requests to load their own dataset for the simulation, that data is stored in the page's session storage. As no server was set up for this project, it had to be stored locally, which also brings advantages like faster access and the data being stored only for the duration of the page session. After storing the user's data locally in their page session, it is accessed by the application to load the dataset and process it.

Decision tree design

C.3 Diseño procedimental

C.4 Diseño arquitectónico

Appendix D

Documentación técnica de programación

D.1 Introducción

D.2 Estructura de directorios

D.3 Manual del programador

D.4 Compilación, instalación y ejecución del proyecto

D.5 Pruebas del sistema

Appendix E

Documentación de usuario

E.1 Introducción

E.2 Requisitos de usuarios

E.3 Instalación

E.4 Manual del usuario

Appendix F

Anexo de sostenibilización curricular

F.1 Introducción

Este anexo incluirá una reflexión personal del alumnado sobre los aspectos de la sostenibilidad que se abordan en el trabajo. Se pueden incluir tantas subsecciones como sean necesarias con la intención de explicar las competencias de sostenibilidad adquiridas durante el alumnado y aplicadas al Trabajo de Fin de Grado.

Más información en el documento de la CRUE https://www.crue.org/wp-content/uploads/2020/02/Directrices_Sosteniblidad_Crue2012.pdf.

Este anexo tendrá una extensión comprendida entre 600 y 800 palabras.

Bibliography

[1] PwC Spain. Spain individual - other taxes, 2024. [Internet; visited 09-june-2024].

[2] WageIndicator. Salario mínimo – españa, 2024. [Internet; visited 09-june-2024].