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 — 8 de junio de 2024
Tutores: Carlos López Nozal
Ismael Ramos Pérez

Índice general

Índice de figuras

Índice de tablas

Apéndice A

Plan de Proyecto Software

A.1. Introducción

A.2. Planificación temporal

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.

You can see the burndown of the sprint in the following graph.

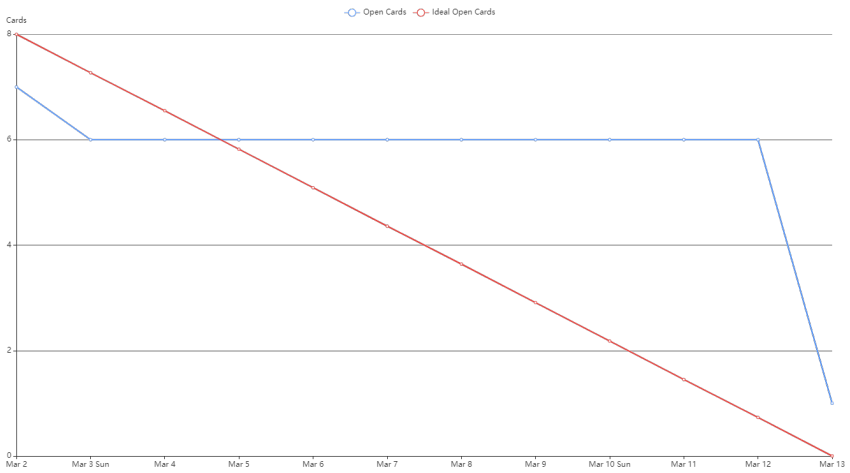

Figura A.1: Burndown Sprint 1

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 page 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 the following image, everything was completed except for the prototype displaying a decision tree. Due to sickness during the sprint, this task was left unfinished.

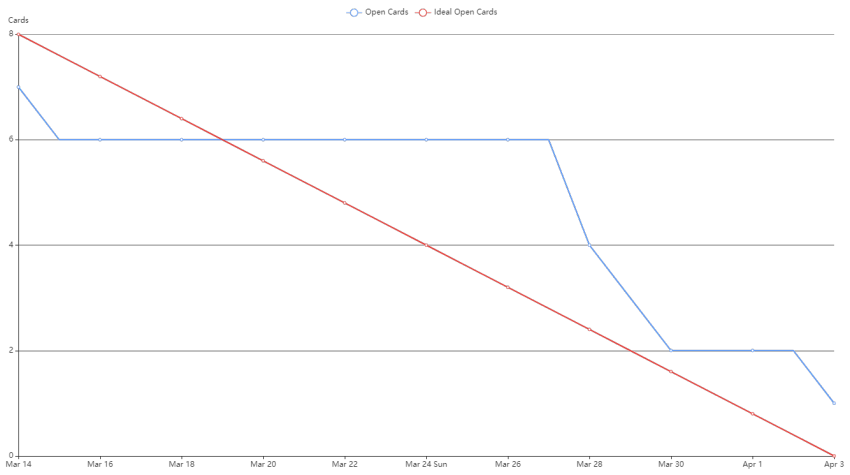

Figura A.2: Burndown Sprint 2

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.

The task of creating a prototype displaying a decision tree was pushed back for this sprint.

Results: As you can see on the burndown in the following image, all the tasks of this sprint have been completed in time.

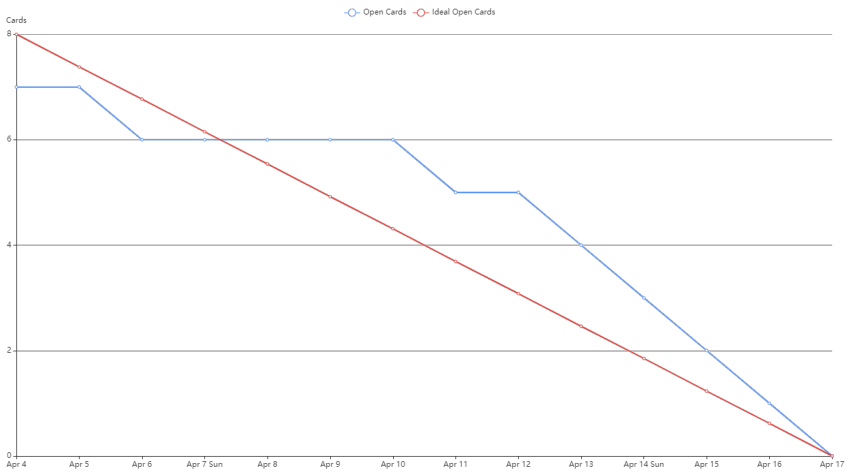

Figura A.3: Burndown Sprint 3

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 the following graph, except for two issues regarding the documentation of related works and a theoretical concept, were completed. This, was due to time constraints caused by assignments and exams in other classes.

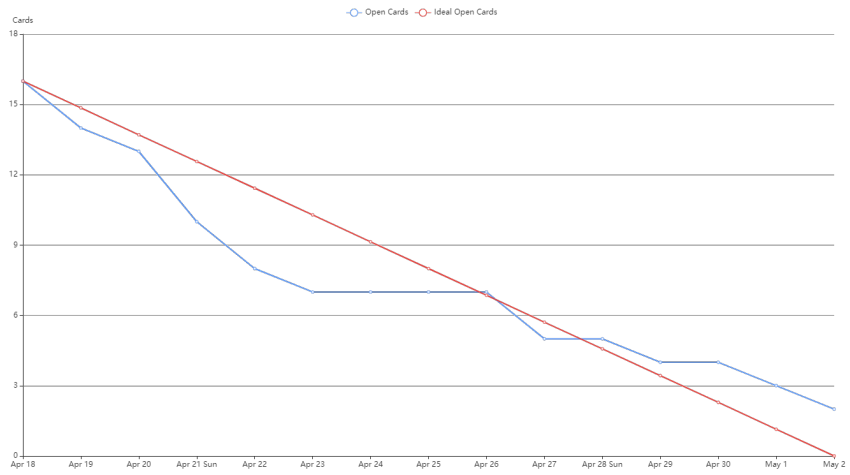

Figura A.4: Burndown Sprint 4

Sprint 5 (03/05/2025 - 16/06/2024): 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: The following graph shows that all the proposed tasks were able to be completed in time.

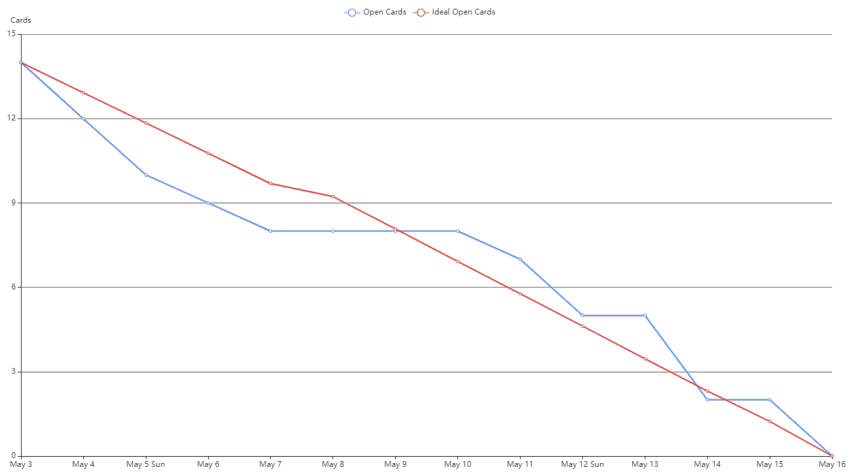

Figura A.5: Burndown Sprint 5

A.3. Estudio de viabilidad

Viabilidad económica

Viabilidad legal

Apéndice B

Requirements Specification

B.1. Introduction

Una muestra de cómo podría ser una tabla de casos de uso:

B.2. General objectives

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.

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.

B.4. Requirements specification

Use case diagram

Use cases

Use cases

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

Tabla 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

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

Apéndice C

Especificación de diseño

C.1. Introducción

C.2. Diseño de datos

C.3. Diseño procedimental

C.4. Diseño arquitectónico

Apéndice 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

Apéndice E

Documentación de usuario

E.1. Introducción

E.2. Requisitos de usuarios

E.3. Instalación

E.4. Manual del usuario

Apéndice 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.

Bibliografía