



UNIVERSIDADE D  
COIMBRA

Eduarda Jorge da Silva

VISUALIZATION OF MACHINE LEARNING  
ALGORITHMS

Dissertation in the context of the Master in Design and Multimedia, advised by Professor Catarina Sofia Henriques Maçãs and Professor Nuno António Marques Lourenço, presented to the Department of Informatics Engineering of the Faculty of Sciences and Technology of the University of Coimbra

September 2023





FACULDADE DE  
CIÊNCIAS E TECNOLOGIA  
UNIVERSIDADE DE  
**COIMBRA**

# Visualization of Machine Learning Algorithms

FINAL DELIVERY

Eduarda Jorge da Silva

Master Degree in Design and Multimedia at  
Faculty of Sciences and Technology  
University of Coimbra

Supervised by  
Catarina Maças  
Nuno Lourenço

January 2023





## Acknowledgements

This work is funded by the project POWER (grant number POCI-01-0247-FEDER-070365), co-financed by the European Regional Development Fund (FEDER), through Portugal 2020 (PT2020), and by the Competitiveness and Internationalization Operational Programme (COMPETE 2020). There are no words for how grateful I feel to have been surrounded by the most extraordinary people.

To my parents, thank you for teaching me to work hard to achieve my goals and helping me to never give up, no matter how challenging. Without their unconditional support, I wouldn't have got this far in my academic career. To my sisters Carolina and Sofia, for believing in me and my abilities, for looking after my future, and for the affinity we have. I also have to thank my closest friends. To my best friend Laura, for all the encouraging speeches and for accompanying me to the last project. To those who accompanied me on my academic journey, Francisca, Rafael, Jorge, and Angela, for not letting me go through the most tense moments alone. To jeKnowledge and the friendships I built there, with whom I grew not only professionally, but also personally.

Last but not least, I am grateful for the guidance of my tutors, Professor Catarina Maças and Professor Nuno Lourenço. For passing on their knowledge, advice, and words of encouragement. It was thanks to these professionals that I gained new knowledge and improved as a designer and as a professional in general.



## Resumo

Vivemos numa era em que as técnicas de Inteligência Artificial (IA) são cada vez mais requisitadas em diversos domínios, sobretudo tendo em conta os avanços tecnológicos e a disponibilidade de dados. A área de *Marketing Digital* não é exceção, e está a sofrer uma profunda transformação impulsionada pelas rápidas mudanças da própria sociedade, levando à necessidade de desenvolver soluções modernas e competitivas. Numa área onde a interação com o cliente é tão frequente e essencial, é necessário encontrar formas de identificar a melhor abordagem para aumentar a satisfação do cliente. Neste âmbito, as técnicas de IA, nomeadamente o *Machine Learning* (ML), podem auxiliar na análise e classificação dos diferentes perfis de clientes. No entanto, a tarefa de analisar/interpretar os resultados obtidos por um modelo de ML não é simples, especialmente no caso de modelos Black-Box, com alto grau de complexidade.

Neste trabalho, propomos a utilização de técnicas de Visualização de Dados (VD) para interpretar algoritmos de ML. A visualização permite a comunicação de relações complexas entre dados de uma forma interpretável. No contexto desta dissertação, as técnicas de visualização, que visam representar conjuntos de árvores de decisão, podem ser acedidas através de uma aplicação *web*, permitindo aos seus utilizadores analisar e melhorar o modelo de ML de uma forma mais eficiente. Na visualização de modelos de ML, a interatividade é um aspeto fundamental na interpretação do modelo.

Depois de analisar e interpretar os dados, estudámos várias técnicas de visualização baseadas em árvores. Decidimos pela técnica da árvore radial para representar a *random forest* e pela técnica do *sunburst* para representar as árvores de decisão. Em vez de representar a *random forest* através de várias árvores, optámos por fazê-lo através de uma única árvore cujos nós revelam as *features* mais importantes em cada nível de profundidade e as árvores que as contêm. A importância das *features* é avaliada consoante o número de árvores da *random forest* em que elas aparecem. Para testar esta solução, desenvolvemos uma aplicação *web* denominada RaVi, que avaliamos através de um questionário. Com base nas respostas dos participantes, foi possível avaliar não só quais os aspetos das visualizações que funcionam melhor, mas também o que precisa de ser melhorado para garantir a melhor experiência do utilizador. Neste documento, apresentamos todo o processo de trabalho, desde o desenvolvimento do conceito até à implementação e validação das visualizações e da aplicação *web*.

**Palavras-chave:** Visualização de Dados, *Random Forest*, Árvores de Decisão, *Features*



# Abstract

We live in an era in which Artificial Intelligence (AI) techniques are increasingly requested in several domains, especially taking into account the technological progresses and data availability. The Digital Marketing area, is no exception, and is suffering a profound transformation driven by the rapid changes in society itself, leading to the need to develop modern and competitive solutions. In an area where interaction with the customer is so frequent and essential, it is necessary to find ways to identify the best approach to increase customer satisfaction. In this scope AI techniques, specially Machine Learning (ML) can assist in the analysis and classification of different customer profiles. However, the task of analyzing/interpret the results obtained by an ML model is not simple, specially in the case of Black-Box models, with a high degree of complexity.

In this work, we propose the use of Data Visualization (DV) techniques to interpret ML algorithms. Visualization allows the communication of complex relationships between data in an interpretable way. In the context of this dissertation, the visualization techniques, that aim to represent decision tree ensembles, can be accessed through a web application, allowing its users to analyze and improve the ML model in a more efficient way. When visualizing ML models, interactivity is a fundamental aspect in the interpretation of the model.

After analyzing and interpreting the data, we studied various tree-based visualization techniques. We decided on the radial tree technique to represent the random forest and the sunburst technique to represent decision trees. Instead of representing the random forest through several trees, we chose to do it through a single tree whose nodes reveal the most important features at each depth level and the trees that contain them. The importance of the features is assessed according to the number of trees in the random forest in which they appear. To test this solution, we developed a web application called RaVi, which we evaluated through a questionnaire. Based on the participants' responses, we were able not only to assess which aspects of the visualizations work best but also what needs to be improved to guarantee the best user experience. In this document, we present the entire work process, from developing the concept to implementing and validating the visualizations and the web application.

**Key-words:** Data Visualization, Random Forest, Decision Trees, Features



# Table of Contents

<b>1. Introdução</b>	<b>1</b>
1.1. Contributions	2
1.2. Document Structure	2
<b>2. Development Plan</b>	<b>5</b>
2.1. Methodology	5
2.2. Work plan	6
<b>3. Background</b>	<b>9</b>
3.1. Machine Learning	9
3.1.1. Interpreting Machine learning models	11
3.1.2. Decision Trees	12
3.1.3. Random Forest	14
<b>4. State of the Art</b>	<b>17</b>
4.1. Data Visualization	17
4.1.1. Design principles for visualization and data design	22
4.2. Visualization in Machine Learning	25
4.2.1. Tree-based visualization techniques	27
4.3. Tree visualizations	35
4.3.1. Baobab View	35
4.3.2. EMTree Results Viewer	36
4.3.3. PaisleyTrees	37
4.3.4. PansyTrees	38
4.3.5. ReFINE	40
4.4. Reflection and conclusion	41
<b>5. RaVi - Random Forest Visualizer</b>	<b>43</b>
5.1. Objectives	43
5.2. Requirements and tasks	44
5.3. Data analysis and structuring	45
5.4. Visualization models	50
5.4.1. Studies for the visual structure	52
5.4.2. Final visualization models	57
5.5. Web application development	60
5.5.1. Prototype design	60
5.5.2. Application implementation	64
5.6. Use cases	67
5.7. Validation	69
5.7.1. Results and analysis	71
<b>6. Conclusion</b>	<b>75</b>
6.1. Limitations and future work	76
6.2. Reflection	76

<b>7. References</b> . . . . .	<b>79</b>
<b>Appendix A</b> . . . . .	<b>83</b>
Visualization of Decision Trees questionnaire . . . . .	83
<b>Appendix B</b> . . . . .	<b>87</b>
Web application wireframes . . . . .	87
<b>Appendix C</b> . . . . .	<b>89</b>
Web application mockups . . . . .	89
<b>Appendix D</b> . . . . .	<b>95</b>
Use cases implemented screens . . . . .	95
<b>Appendix E</b> . . . . .	<b>99</b>
RaVi validation questionnaire . . . . .	99



# List of figures

Figure 2.1 - Visualization Design Process . . . . .	6
Figure 2.2 - Gantt Chart of the work plan . . . . .	7
Figure 3.1 - Machine Learning Pipeline adapted from SPI POWER subproject . . . . .	10
Figure 3.2 - Neural Network (Melo, 2023) . . . . .	11
Figure 3.3 - Decision Tree Legend . . . . .	12
Figure 3.4 - Example of a decision tree based on the work of Yaser Sakkaf . . . . .	13
Figure 3.5 - Visualization of a Random Forest model prediction . . . . .	15
Figure 4.1 - 2x2 definition of the four types of visual communication . . . . .	18
Figure 4.2. - Visualization pipeline inspired in Card et al, 1999 . . . . .	19
Figure 4.3 - Visual Mapping structures . . . . .	20
Figure 4.4 - Nested Model (Munzner, 2009) . . . . .	20
Figure 4.5 - Pittsburg Civic Commission, Report on Expenditures of the Department of (Graphical integrity, (n.d.)) . . . . .	23
Figure 4.6 - New York Times, August 8, 1978 (Tufte, 2001) (left); New York Times, December 19, 1978 (Graphical integrity, n.d.) (right) . . . . .	24
Figure 4.7 - Washington Post, October 25, 1978 (Tufte, 2001) (left); Connecticut Traffic Deaths, 1951-1959 (Tufte, 2001) (right) . . . . .	24
Figure 4.8 - The Form of Facts and Figures, Christian Behrens, 2008 . . . . .	25
Figure 4.9 - Different visualizations of decision trees (Lima, 2014) . . . . .	28
Figure 4.10 - Genealogical tree of Charles Magius from Paul Veronese, Codex Magius, 1568-73 (left); X-Men Family Tree, Joe Stone, 2011 (right) . . . . .	29
Figure 4.11 - Indented tree, Manuel Lima, 2013 (left); Writing Without Words, Stefanie Posavec, 2008 (right) . . . . .	29
Figure 4.12 - Pedigree chart, Marle R. Walter, 1938 (left); Treeviz, Werner Randelshofer, 2007 (right) . . . . .	30
Figure 4.13 - Pebbles, Kai Wetzal, 2003 (left); The Champions Ring, Deroy Peraza, 2012 (right) . . . . .	31
Figure 4.14 - Eclipse Voronoi treemap, Oliver Deussen, 2010 (left); Classification of the Occupations by Race and Nativity, Henry Gannett, 1900 (right) . . . . .	32
Figure 4.15 - ADoReVA (automatic detection of register variations) . . . . .	33
Figure 4.16 - CNN Ecosphere Project, Minivegas and Heimat Berlin, 2011 . . . . .	33
Figure 4.16 - Fisheye lens . . . . .	34
Figure 4.17 - Examples of decision trees visualizations from treevis.net . . . . .	34
Figure 4.19 - Close-up of the node information . . . . .	35
Figure 4.18 - Interface of the interactive decision tree construction software Baobab View . . . . .	35
Figure 4.20 - Screenshots of the application taken from Barlow et al (2005) . . . . .	36
Figure 4.21 - Display of the rules as part of the tree (left) and in a separate window (right) . . . . .	37

Figure 4.22 - Drilling into a PaisleyTree, K.Etemad, D.Baur, J.Brosz, S.Carpendale and F.F.Samavati, 2014 . . . . .	38
Figure 4.23 - PansyTrees merging three datasets from the Chinese National College Entrance Examination (CNCEE) results in 2017 . . . . .	39
Figure 4.24 - Description of element design (a) and overview element (b) . . . . .	39
Figure 4.25 - Relations between the main panels of the ReFINE, Kuznetsova, 2014 . . . . .	40
Figure 5.1 - Original data structure: random forest properties . . . . .	45
Figure 5.2 - Original data structure: “estimator” properties . . . . .	46
Figure 5.3 - Outline of the first decision tree of the data set . . . . .	47
Figure 5.4 - Structure of a radial tree . . . . .	47
Figure 5.5 - Example of a random forest representation (below, left) composed by two decision trees (above) and the data structure of a node (below, right) . . . . .	48
Figure 5.6 - Structuring of the data to visualize the Random Forest . . . . .	49
Figure 5.7 - Structuring of the data to visualize the Decision Tree . . . . .	50
Figure 5.8 - Radial tree layout example . . . . .	51
Figure 5.9 - Study sunburst vs icicle plot . . . . .	51
Figure 5.10 - Six decision trees based on different bootstrap samples . . . . .	52
Figure 5.11 - Six decision trees transformed into sunbursts . . . . .	52
Figure 5.12 - Studies to distinguish the direction of tree branching in different tree structures . . . . .	53
Figure 5.13 - Results of the “Decision Tree Visualization” questionnaire . . . . .	54
Figure 5.14 - Leaf-nodes visuals tests . . . . .	54
Figure 5.15 - Study of the pie chart visualization technique for class presentation (left) and the visualization technique for presenting the “impurity” property (right) . . . . .	55
Figure 5.16 - Study of colours to be assigned to each feature . . . . .	55
Figure 5.17 - Colour Palette . . . . .	56
Figure 5.18 - Tests for the random forest visualization . . . . .	56
Figure 5.19 - Feature importance graph . . . . .	57
Figure 5.20 - Conversion of decision tree data to sunburst . . . . .	58
Figure 5.21 - Visualization of different decision trees . . . . .	58
Figure 5.22 - Conversion of two decision trees into a radial tree . . . . .	59
Figure 5.23 - Radial trees with different number of trees . . . . .	59
Figure 5.24 - Wireframes for the intermediate delivery . . . . .	60
Figure 5.25 - Application user flow . . . . .	61
Figure 5.26 - Updated Wireframes . . . . .	61
Figure 5.27 - “Radial Tree Visualization” screens . . . . .	62
Figure 5.28 - “Multiple Trees Visualization” screens . . . . .	63
Figure 5.29 - “Sunburst Visualization” screens . . . . .	63
Figure 5.30 - Code structure . . . . .	65
Figure 5.31 - “Radial Tree Visualization” implemented Screen . . . . .	66
Figure 5.32 - “Sunburst Visualization” implemented Screen . . . . .	66
Figure 5.33 - Radial tree screens . . . . .	67
Figure 5.34 - Radial tree and multiple trees screens . . . . .	68

Figure 5.35 - Radial tree (above, left), multiple trees (above, right), and sunburst screens (bellow) . . . . .	69
Figure 5.36 - Section 4: Number of “Yes” and “No” answers per question (left); Section 5: Number of “Yes” and “No” answers per question (right) . . . . .	72
Figure 5.37 - Section 6: Number of “Yes” and “No” answers per question . . . . .	72
Figure 5.38 - Section 8: Semantic Differential Scale Results . . . . .	74
Figure A.1 - Visualization of Decision trees questionnaire: introduction . . . . .	83
Figure A.2 - Visualization of Decision trees questionnaire: visualization models . . . . .	84
Figure A.3 - Visualization of Decision trees questionnaire: user information . . . . .	84
Figure A.4 - Visualization of Decision trees questionnaire: tree visualizations . . . . .	85
Figure A.5 - Visualization of Decision trees questionnaire: comments . . . . .	86
Figure B.1 - Web application wireframes: random forest visualization screen . . . . .	87
Figure B.2 - Web application wireframes: multiple trees visualization screen . . . . .	88
Figure B.3 - Web application wireframes: decision visualization screen . . . . .	88
Figure C.1 - Web application mockups: radial tree visualization screen . . . . .	89
Figure C.2 - Web application mockups: radial tree filter screen . . . . .	90
Figure C.3 - Web application mockups: radial tree zoom screen . . . . .	90
Figure C.4 - Web application mockups: radial tree legend filter screen . . . . .	91
Figure C.5 - Web application mockups: multiple tree visualization screen . . . . .	91
Figure C.6 - Web application mockups: multiple trees visualization zoom screen . . . . .	92
Figure C.7 - Web application mockups: sunburst visualization screen . . . . .	92
Figure C.8 - Web application mockups: node information screen . . . . .	93
Figure D.1 - Implemented screens: radial tree visualization . . . . .	95
Figure D.2 - Implemented screens: radial tree visualization zoom . . . . .	96
Figure D.3 - Implemented screens: radial tree visualization filter . . . . .	96
Figure D.4 - Implemented screens: radial tree legend filter . . . . .	97
Figure D.5 - Implemented screens: radial tree visualization cursor hover . . . . .	97
Figure D.6 - Implemented screens: multiple tree visualization . . . . .	98
Figure D.7 - Implemented screens: sunburst visualization . . . . .	98
Figure E.1 - RaVi validation questionnaire: introduction section . . . . .	99
Figure E.2 - RaVi validation questionnaire: models transformation section . . . . .	100
Figure E.3 - RaVi validation questionnaire: user information section . . . . .	101
Figure E.4 - RaVi validation questionnaire: random forest section . . . . .	102
Figure E.5 - RaVi validation questionnaire: multiple tree section . . . . .	103
Figure E.6 - RaVi validation questionnaire: decision tree section . . . . .	104
Figure E.7 - RaVi validation questionnaire: open-ended answer section . . . . .	105
Figure E.8 - RaVi validation questionnaire: semantic differential scale section . . . . .	106



## **Acronyms**

**AI** - Artificial Intelligence

**ML** - Machine Learning

**DV** - Data Visualization

**RaVi** - Random Forest Visualization



# 1. Introduction

In an era in which marketing is undergoing a profound transformation, driven by the accelerated changes of society itself, it is essential to develop modern and competitive solutions that respond to the new interaction and customer engagement paradigms in the digital marketing domain. In arriving at this solution, we are also able to respond to the unique needs and challenges for the marketers.

To address the aforementioned challenges and to facilitate the analysis and classification of different customer profiles, Artificial Intelligence (AI) techniques have been applied. There is no doubt that AI has been expanding its application domains, but, most AI techniques, specially ML models tend to be Black-Box, and given its complexity, it becomes difficult to analyze how and why a given model produces a given output. To enable the analysis of these models by researchers and other end users, Data Visualization (DV) techniques have been applied.

This work consists on the development of a visualization model to help understand ML algorithms. Our research was developed in the context of the project POWER - Empowering a digital future, which results from a collaboration between several institutions, namely, Altice S.A., University of Coimbra, IT Aveiro, and Instituto Pedro Nunes (IPN).

This dissertation is part of a specific work package within POWER, in which the main goal is to expand the current functional scope of Altice's ACM (Active Campaign Management) product. It is focused on the functional areas of Campaign Analytics, and to transform it into a more comprehensive Digital Marketing product, with a coherent cross-channel interaction logic and strongly leveraged on data, analytics, and Artificial Intelligence (AI) / Machine Learning (ML) (data-driven and AI-driven). More specifically, our work is inserted within a specific task, which aims to develop intelligent tools for the analysis and construction of computational models to identify the most convenient channel (e.g. e-mail, SMS, voice) to contact each customer, increasing the probability of successful interaction and customer satisfaction. Since we already have the ML model—random forest—implemented, our contribution is to analyze and structure the data obtained and develop a visualization to facilitate its interpretation.

The main objectives of this dissertation are the (i) analysis of the state of the art and related work on Data Visualization to assist in ML models interpretation, (ii) survey of the type of visualization models and type of data from ML models and designing of the corresponding taxonomy, (iii) with the knowledge acquired in the previous points, develop the a project that consist in a web framework with visualizations using the DV techniques previously explored, and (iv) test the application and analyze the

results. By achieving these objectives, we can develop a tool, through DV, that aids in the analysis and improvement of Random Forest models.

## **1.1. Contributions**

During the work conducted in this dissertation, we had to face two challenges. The problem centered on the researcher's difficulty in understanding and improving the ML model, and the users' difficulty in interpreting the model to accept or question its results. That said, our main contribution was to develop a solution to alleviate this problem.

For users to be able to better analyze the model's data, we created a framework that consists of visualizing it. We took advantage of tree-based visualization techniques and developed a visualization not only for the random forest but also for decision trees. The design of the application is simple and neutral in color, with the exception of the colors assigned to the features present in the data, thus emphasizing them. In addition to the tree-based visualizations, we implemented commonly known techniques such as bar charts and pie charts to transmit data more efficiently. As interaction is also an essential part of understanding the models, we have implemented techniques such as zooming and filtering.

The simple design, the use of symmetrical and hierarchical visualizations, as well as the use of well-known techniques and some interactions, make it easier to interpret the model. In this way, both researchers and the end user can extract data from the model and analyze the data more efficiently.

## **1.2. Document Structure**

This document contains seven chapters. In the introduction, we provide information about the context and motivation for the dissertation, as well as its objectives. In the second chapter, we present the development plan which includes the methodology used and the work plan to be carried out throughout the internship period. In the third chapter, Background, we present a brief overview of the ML area and its process, the importance of ML model interpretation, and address the concept of Decision Tree and Random Forest.

The fourth chapter is the state of the art, where we synthesize our research process for the dissertation. In this chapter, we start by briefly mentioning the topic of Data Visualization and present some design principles for data visualization. Furthermore, we take a look at the role of Visualization in ML, explore the techniques used in tree-based visualizations, and specify the visualization of decision trees. Also in the fourth chapter, we analyze some previous works that are related to the topic of the dissertation, enabling us to better understand the challenges and requirements, and end with a reflection on the state of the art.



Chapter five is named after the tool developed for the practical component of the thesis. In this section, we start by indicating the objective of the project, and the requirements and tasks. Moreover, we present the work done on the analysis and structuring of the data, and the visualization models, including the studies for the visual properties done on them until reaching the final models. In this chapter, we also provide an overview of the development of the web application from prototypes to its implementation and tools used. Furthermore, we provide two use cases in which we describe the interaction with the application, and we present the user tests to evaluate the visualizations of the application and the analysis of the results obtained. Finally, in the conclusion, we summarize the work carried out.



## 2. Development Plan

In this chapter, we present the plan regarding the development of this work. Firstly, we discuss the methodology used. Then, we present the work plan divided into tasks briefly explained and presented in a Gantt chart along with the intermediary and final deadlines.

### 2.1. Methodology

The methodology chosen for the development of this project follows the Visualization Design Process (Maçãs et al, 2021). In design, the user is strongly involved throughout the process of creating any tool. The term design thinking is widely used in design processes to support solving complex problems in a user-centric way. According to the Nielsen Norman Group, the six phases of design thinking are empathize, define, ideate, prototype, implement and test (Schilling, 2022). Empathize and define, help us understand the problem. Ideate and prototype, help us explore solutions. Test and implement, help us materialize the final product.

When a product is created within the design-thinking process, the user is considered in every step. This results in a thoughtfully designed product to meet their needs which turns out to be an added value in a data visualization design process.

A design process offers great value while working in the field of data visualization. By focusing on the user, we find out the best way to enable or enhance their data-informed decision-making, starting by understanding the user and their needs and ending with testing solutions with the end-user to get feedback.

That said, the data visualization design process, is based on five steps (Figure 2.1):

1. Definition of the problem and target audience
2. Translation of the data
3. Designing visual encodings and interaction
4. Implementation
5. Validation

By defining the problem and the audience, we can more easily come up with the best possible solution that is beneficial to the user in question. After this phase, it is important to translate the data that has been provided and that has to be worked on throughout the project to understand how it should be transmitted to the user through visualization (Brush et al, 2022).

Having said that, the next phase is the design of the visualization model that best fits the data provided taking into account the interactions with the user, not only to improve their experience with the visualization but also to facilitate interpretation (How to design an information visualization, 2023). In the next phase, the designer is dedicated to the implementation of the idealized visualization, followed by the validation phase with end users which is fundamental for the refinement of the project.



**Figure 2.1 - Visualization Design Process**

We opted for this methodology because, in our opinion, it is the most appropriate for a job involving data visualization. It describes the process of arriving at a final product, taking into account fundamental steps for data visualization, such as data analysis and interpretation.

## **2.2. Work plan**

This section corresponds to the presentation of the work plan (Figure 2.2) carried out during the development of the dissertation. This work started in September 2022 and was scheduled to end in July 2023, however, it was extended to September 2023. The work plan is divided into five principal parts:

### **1. Understanding of the current state of the art**

The first part of the work plan is focused on the theoretical portion of the dissertation. It is composed of five sub-tasks such as (i) definition of the problem we aim to solve; (ii) search and selection of articles that are related to the theme of the thesis; (iii) research on existing visualization platforms; and (iv) reading and analysis of the articles and collection of information for the writing of the document.

### **2. Requirements and data processing**

The second part of the work is composed of two sub-tasks, namely (i) establishing the tasks and requirements for the web application and (ii) analyzing and structuring the data according to the tasks and visualization.

### **3. Design of visual encodings**

The third part of the work plan refers to the more visual component of the practical project, not only the visualization models but also the web application itself. This section is composed of three sub-tasks: (i) research on the type of visualization according to the data that was given; (ii) elaboration of studies on the visualization

models and mockups for the web application; and (iii) validation of the final visualization models.

#### 4. Implementation and validation

This fourth part of the work plan is related to the technical portion of the development of the project and is divided into four different sub-tasks: (i) familiarization with the D3.js library which includes viewing and following some tutorials; (ii) implementation of the web application; (iii) performing of tests with end users; and (iv) refinement of the application according to the test results.

#### 5. Writing of the document and presentation

The last part of the work plan focuses only on the written document and the defense presentations. That said, this strand is divided into five sub-tasks: (i) writing the dissertation document for the intermediate defense; (ii) preparing the presentation for the intermediate defense; (iii) writing the dissertation document for the final defense taking into account the comments of the first defense; (iv) preparing the presentation for the final defense; and (v) revising and refining the document.

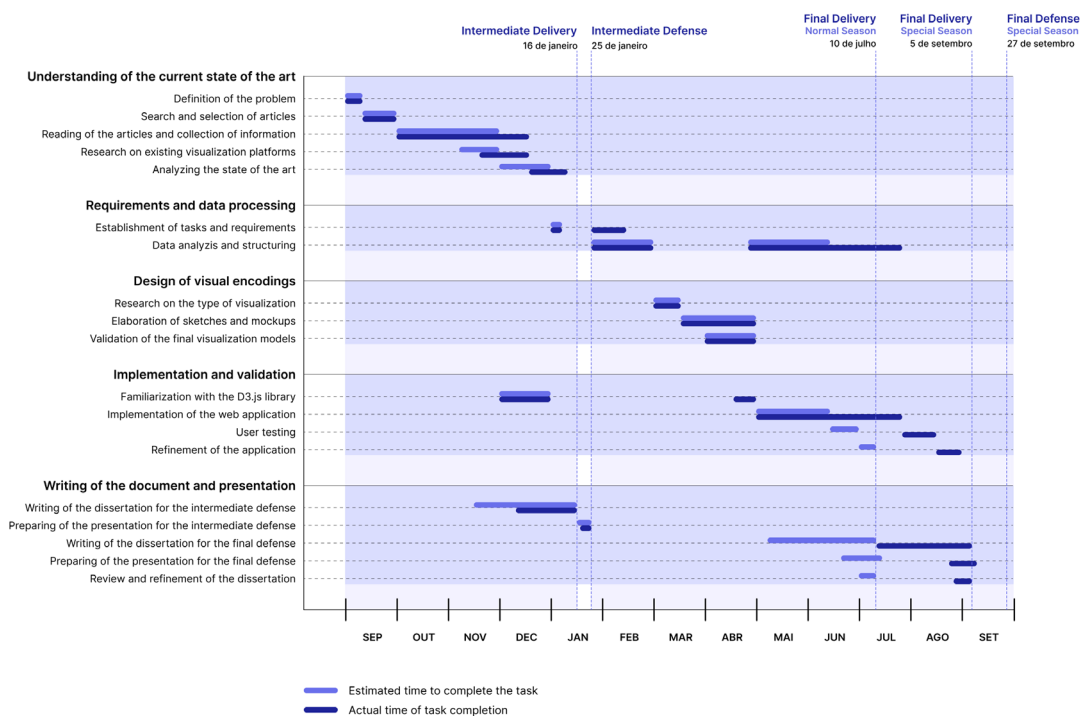


Figure 2.2 - Gantt Chart of the work plan



## 3. Background

To develop a project of this magnitude, it is essential to have an understanding of the study areas in question. In this chapter, we present a brief analysis of some important concepts for the background of the dissertation.

First, we provide a general the basics concepts of ML and the importance of interpreting ML models. We do not intend to give an exhaustive description of the ML field, but rather provide the reader with the knowledge needed to understand the work being developed. For an extended description the interested reader is referred to Bishop (2006). Secondly, we focus specifically on Random Forest and decision tree algorithms to better contextualize the work carried out.

### 3.1. Machine Learning

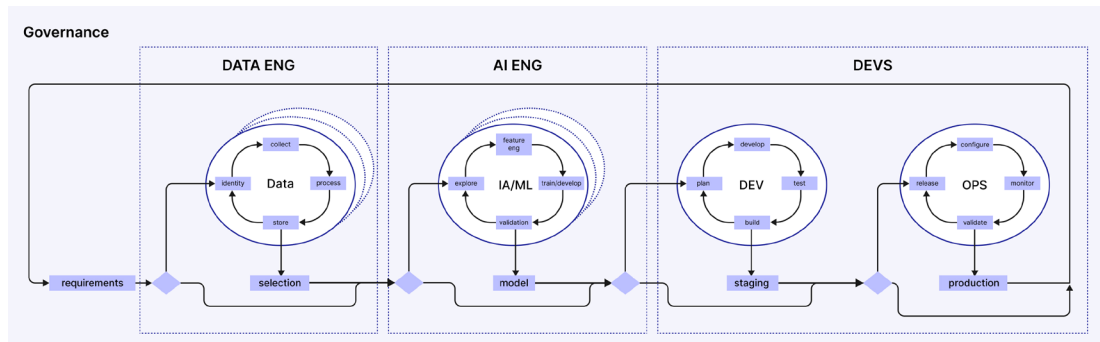
Over the years, and fueled by the technological advances and the amount of information being collected, the interest in the field of Machine Learning (ML) and the application of its techniques in different fields has increased substantially. . With this increased interest comes the need for transparency and interpretation of ML models and algorithms (Scalco, 2021). Being a subfield within Artificial Intelligence (AI), ML is the area that studies the ability of computers to learn through algorithms, making it possible for the machine to make decisions autonomously . Currently, ML is used in several domains, with various purposes, such as complex data classification and/or decision-making (Somvanshi et al, 2016).

The concept of learning can be understood as broad, from which a system learns from past experiences to improve its performance. Within ML, there are many types of algorithms that can be divided into three main categories (Somvanshi et al, 2016):

**Supervised Learning:** we have both input and output variables available. To train the model—so it can learn to predict an output based on a new input—it is necessary to provide annotated examples. Supervised learning is the most suitable algorithm in cases where there is knowledge of what a model should predict based on a specific input.

**Un-Supervised Learning:** the opposite of supervised learning that is mainly used in exploring processes. It can be used if we have a dataset with only input data, but do not have the desired output. The goal is to search for groups or categories in the existing data. The machine will scan the incoming data and structure these into categories or clusters.

**Reinforcement Learning:** This is a form of ML in which a machine learns based on trial and error. With this method, the model will be optimized via feedback on previous actions and experiences. Usually, the optimization is based on rewards, meaning that some of those actions might provide more meaningful experiences. In the context of this work, we will focus on two supervised learning methods. These methods are Decision Trees and Random Forests.



**Figure 3.1** - Machine Learning Pipeline adapted from SPI POWER subproject

The ML pipeline presented in Figure 3.1, consists of a set of steps to build and deploy an ML-based system. This process covers three subjects: (i) Data Engineering, (ii) AI Engineering, and (iii) Development/Operations.

Before entering the data processing phase, it is important to define the problem and collect requirements for the desired outcome. After this phase, data collection and processing follows. This includes acquiring, cleaning, and processing the data in a format that can be used to train the model. This step is crucial since the quality and quantity of data that is collected can determine the quality of the outcomes. After the data is selected, we go through a series of procedures that includes data exploration, feature engineering - which refers to the process of selecting, manipulating, and transforming raw data into features that can be used in supervised learning -, development, training, and validation of the algorithm.

After the modeling stage, we move on to the fine-tuning stage where experiments are carried out on the developed model. Once built and after going through the staging process, the final model is launched in a production environment and is monitored.

It should be noted that there are several cycles, not only within each stage but also in the process in general. This is due to the fact that the ML process is not linear, since at any stage there may be an event that changes the data or even the algorithm itself, that is also why the validation and test stages are so crucial during the process.



### 3.1.1. Interpreting Machine learning models

Black-Box models have gained relevance, being used countless times in different problem domains.. However, despite the results they obtained, there is a flaw in the transparency of this technology, since they are unable to provide a reason as to why they made a certain decision. Since neither the programmers nor the end users understand how the output is reached, it can be said that Black-Box models, such as neural networks (Figure 3.2), are too complex to be interpreted (Rudin, 2019).

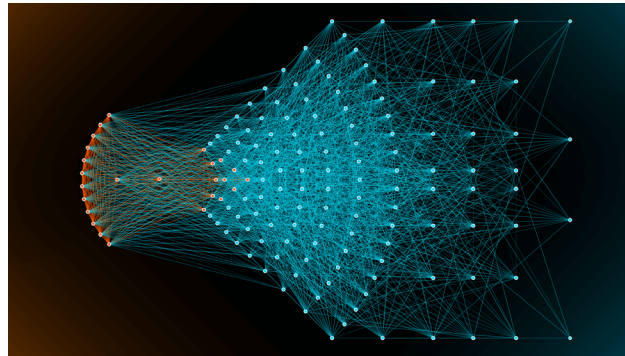


Figure 3.2 - Neural Network (Melo, 2023)

That being said, whatever the objective is when it comes to data processing and visualization, the end-user will prefer models that present an interpretable and understandable solution. For instance, for a data scientist, the interpretability of the model is more beneficial in terms of validating and improving its work (Hulstaert, 2018).

While the ability to train models is critical concerning data processing, it is important to be able to look at the bigger picture. Interpretability of data and ML models is crucial in the practical ‘usefulness’ of a data science pipeline for the following reasons: (i) identifying and mitigating bias; (ii) accounting for the context of the problem; and (iii) improving generalization and performance (Hulstaert, 2018).

Interpretability and explanations are crucial for enriching learning and satisfying curiosity about machine-generated predictions and behaviors. In situations where substantial problems are addressed using vast datasets and untransparent machine learning models, shifting the source of knowledge from data to the model itself, interpretability becomes vital for extracting the model’s hidden insights. It also serves as a key tool in uncovering bias, such as potential discrimination against historically marginalized groups in models like credit application approvals. As machines and algorithms become increasingly integrated into daily life, interpretability becomes essential for promoting their social acceptance. Furthermore, interpretability enables effective debugging and auditing of machine learning models, proving valuable even in seemingly low-risk scenarios like movie recommendations, spanning research, development, and deployment phases (Molnar, 2020).

The need for interpretability in machine learning arises from situations where problem formalization is incomplete, necessitating explanations alongside predictions.

Correct predictions alone may not fully address the original problem. The demand for interpretability is driven by several factors (Doshi-Velez et al, 2017):

**Fairness:** Interpretable models can help ensure unbiased predictions that don't discriminate against underrepresented groups. The model's explanations can reveal whether decisions are influenced by learned biases, such as racial biases, making it easier for human judgment.

**Privacy:** Interpretability aids in safeguarding sensitive data, ensuring that confidential information remains protected.

**Reliability or Robustness:** Interpretable models aim to maintain consistent predictions despite minor changes in input data, preventing drastic fluctuations in output.

**Causality:** Interpretability facilitates checking that the model captures only causal relationships between variables, enhancing the understanding of how inputs relate to outputs.

**Trust:** Systems that provide explanations for their decisions are more likely to gain human trust compared to unclear Black-Box models.

### 3.1.2. Decision Trees

To better understand Random Forests, it is essential to learn what a decision tree is. A decision tree, as represented in Figure 3.3, is a hierarchical structure where each internal node represents a feature, each branch represents a decision rule based on that feature, and each leaf node represents the outcome or the prediction (What is a Decision Tree, (n.d.)). Decision trees are relatively easy to interpret and can be powerful in making decisions based on the data.

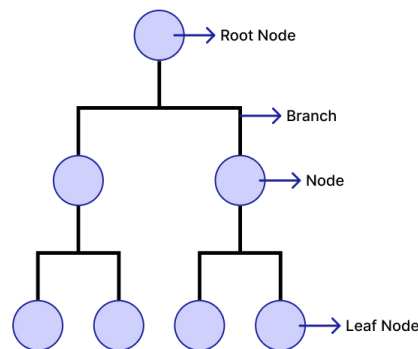


Figure 3.3 - Decision Tree Legend

Decision trees are one of the most popular types of predictive models used to partition a large amount of data into smaller subsets by applying a series of rules until the subsets cannot be further partitioned (Barlow et al, 2001). It can also be seen as a flexible classification technique with simple and interpretable structures based on a divide-and-conquer strategy (Van den Elzen et al 2011).

The decision tree structure, represented in the example of Figure 3.4, consists of a root node (line i), nodes (line ii), leaf nodes (line iii), and branches that link the nodes together. The root node represents the original dataset, the decision node specifies a test over one of the attributes and the leaf node represents one of the two possible outcomes. Following the figure below, we give an example taken from the article Decision Trees: ID3 Algorithm Explained (Sakkaf, 2021), on how to interpret a decision tree.

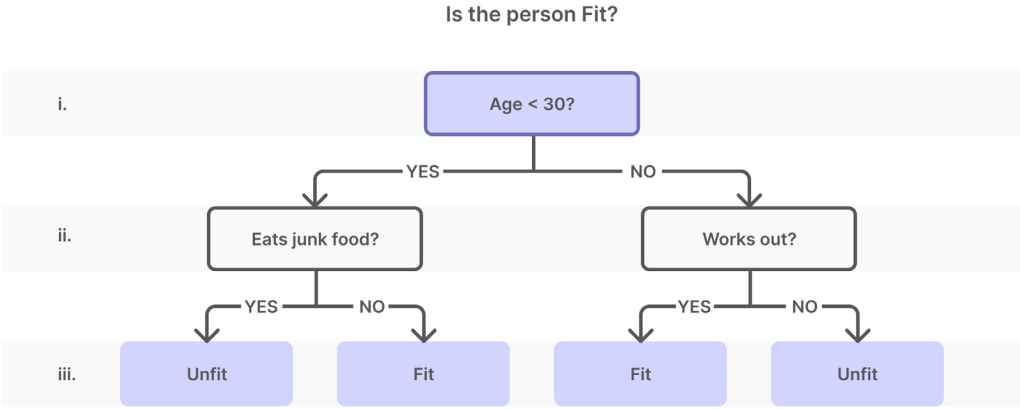


Figure 3.4 - Example of a decision tree based on the work of Yaser Sakkaf

The decision tree above is used to classify whether a person is fit or not. Through the questions asked to the system, represented in the decision nodes, we can reach two possible outcomes, Fit and Unfit.

Analyzing the decision tree we can conclude that for a person to be Fit, the set of rules is to be under 30 and not eat junk food or be over 30 and exercise. On the other hand, for the person to be Unfit, he/she must be under 30 and eat junk food or be over 30 and not exercise.

The decision tree is often used by data scientists. Nonetheless, every algorithm has its advantages and disadvantages. Some of the disadvantages of a Decision Tree according to Analytix Labs (Kapil, 2022) are:

**Overfitting**

Given that it is a high-variance algorithm, it can easily overfit due to the fact that it has no mechanism to stop creating complex decision rules. In addition to that, as the number of nodes increases exponentially with depth, and more nodes are created, it becomes more difficult to understand the decision rules of the tree.

### **Feature Reduction & Data Resampling**

The same feature may occur multiple times at different levels in the tree, making it time-consuming and difficult for the viewer to understand how a feature is used by the model across all rules it generates.

### **Optimization**

The decision tree algorithm looks for the pure node and does not consider how the recent decision will affect the next few stages of splitting. This makes the model interpretable but does not ensure that the algorithm will return the globally optimal result.

Besides the disadvantages, decision trees are generally used by data scientists to start with due to their various features and following advantages:

#### **Interpretability**

Perhaps, one of the greatest advantages of Decision Trees is its highly intuitive and easy-to-learn characteristics that induces confidence in the stakeholders and provides detailed information about what happens during the process of decision-making. The data representation through a flow chart-like manner makes it effortless for data scientists and other professionals to interpret, visualize, and comprehend.

#### **Fewer Data Preparation**

In the development of the majority of models, the quality of the predictions made by the model is dependent upon the quality of data being fed to the model. In the case of decision trees, the preparation steps including standardization and missing value treatment are not required.

#### **Non-Parametric**

The Decision Tree algorithm is non-parametric which means that there are no significant assumptions to be fulfilled or data distribution to be considered.

#### **Versatility**

What makes this algorithm so versatile is its capacity to perform multiple roles apart from standard predictions. It has the ability to perform data exploration and solve regression as well as classification problems.

There is a vast variety of ML algorithms, so choosing the right algorithm for a given dataset and problem is the main key to remember while creating a model. Decision trees end up being an ideal choice given its ability to mimic human thinking while making decisions that can be easily understood because of its tree-like structure.

### **3.1.3. Random Forest**

A significant part of ML algorithms are used to perform classification tasks. The capability to accurately categorize observations holds significant worth across diverse business cases, such as predicting whether a specific user will make a purchase or not.

Data science provides a plethora of classification algorithms such as logistic regression, support vector machine, and decision trees, but near the top of the classifier hierarchy and one of the most popular and common algorithms used by data scientists is the Random Forest classifier. Random Forest is a supervised machine learning algorithm that is widely used for both classification and regression problems. It is an ensemble learning method developed by Leo Breiman and Adele Cutler, that combines multiple decision trees to make more accurate and robust predictions (Yiu, 2021).

A Random Forest is an ensemble method. Ensemble learning methods are made up of a set of classifiers, for example, decision trees, and their predictions are aggregated to identify the most popular result. By combining individual models, the ensemble model tends to be more flexible and less data-sensitive (Chen, 2021). Two of the most recognizable ensemble techniques are bagging, which is sometimes referred to as bootstrap aggregation, and boosting (What is Random Forest?, (n.d.)). Within the bagging approach, a training set's data is randomly sampled with replacement, implying that individual data points have the potential to be selected multiple times. After several data samples are generated, these models are then trained independently, and depending on the type of task, the average or majority of those predictions yield a more accurate estimate.

That said, Random Forest is a classification algorithm consisting of many decision trees. It uses bagging and features randomness when building each tree to try to create an uncorrelated forest of trees whose prediction by committee is more accurate than that of any individual tree (Yiu, 2021).

The random aspect comes into play when constructing individual decision trees. In addition to using a random subset of the training data, when it comes to splitting a node, the algorithm selects a random subset of features instead of considering every possible feature. This random feature selection forces even more variation amongst the trees in the model helps reduce overfitting, lowers correlation across trees, and increases the diversity of the trees in the forest (Yiu, 2021).

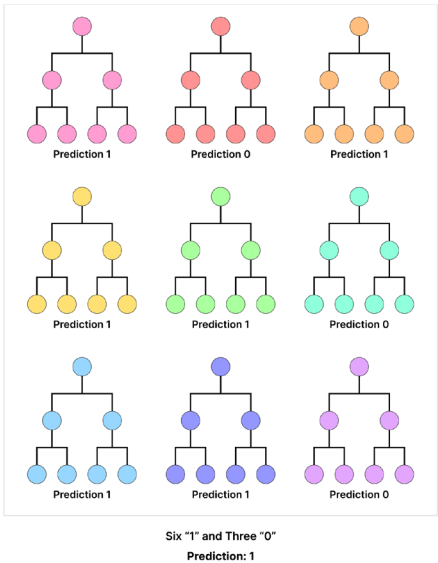


Figure 3.5 - Visualization of a Random Forest model prediction

Once all the decision trees are built, when predicting a new data point, each tree in the forest predicts an outcome, and the final prediction is determined by either taking a majority vote (for classification tasks) or averaging the individual predictions (for regression tasks). Figure 3.5 shows how the prediction is determined in a random forest.

Although the algorithm presents some challenges, such as being a time-consuming process, requiring more resources and higher complexity, using this method offers benefits such as reduced risk of overfitting, providing flexibility, and facilitating the determination of feature importances (What is Random Forest?, (n.d.)).

The strength of the Random Forest algorithm lies in its ability to handle large and complex datasets, as well as its resistance to overfitting. It tends to generalize well on new data, making it a powerful and widely used ML algorithm in various domains, including image recognition, natural language processing, and finance. The approach, which combines several randomized decision trees and aggregates their predictions, has shown excellent performance in settings where the number of variables is much larger than the number of observations. Moreover, it is versatile enough to be applied to large-scale problems, is easily adapted to various learning tasks, and returns measures of variable importance (Biau et al, 2016).

**“Data visualization is the language of decision-making. Good charts effectively convey information. Great charts enable, inform, and improve decision-making”**  
**Dante Vitagliano (Pandey, 2023)**

## 4. State of the Art

In the rapidly evolving landscape of Data Visualization and Machine Learning, understanding the current advancements and methodologies is crucial for researchers and practitioners alike. This chapter delves into the State-of-Art,” providing an exploration of techniques, design principles, and tools that constitute the forefront of DV and visualization in ML.

Within the realm of data visualization, analyze the intricate interplay between aesthetics and functionality by taking a look into the Design Principles for Visualization and Data Design that we find most suitable for the development of our project. Advancements in ML have undeniably transformed the landscape of data analysis which leads us to investigate the role of visualization in enhancing our understanding of ML models. Specifically, Tree-based Visualization Techniques are dissected, shedding light on the visualization methods that make complex tree-based models comprehensible, even to non-experts.

The section on related work presents a comprehensive survey of existing tools and frameworks. EMTree Results Viewer, BaobabView, PaisleyTrees, PansyTrees, and ReFINE are briefly examined. These tools, each unique in their approach, showcase the diversity of efforts to address the challenges in visualizing complex data structures and model outcomes, revealing the spectrum of available solutions.

### 4.1. Data Visualization

**“The use of computer-supported, interactive, visual representations of abstract data to amplify cognition.”**  
— Card, Mackinlay, & Shneiderman, 1999

In this section, we address the topic of Data Visualization, an interdisciplinary field that intersects graphic design, human-computer interaction, computer graphics, and statistics. Data Visualization consists of developing visualization techniques to represent abstract data through the use of common graphics, such as charts, plots, infographics, and even animations. The visual displays of information communicate complex data relationships and data-driven insights in an understandable and interpretable way (What is Data Visualization?, (n.d.)).

Decision-making increasingly relies on data, which is being generated in an overwhelming velocity, and in such volume, that it is difficult to comprehend it without

visual support, making visual communication essential. An important aspect to note about data visualization is its ability to enhance cognition. Cognition refers to the acquisition or use of knowledge, and external cognition is concerned with the interaction of representations and cognitive processes across the external/internal boundary to support thinking. According to Card & Shneiderman (Card et al, 1999) visualization can amplify cognition by: (i) increasing the memory and processing resources available to the users, (ii) reducing the search for information, (iii) using visual representations to enhance the detection of patterns, (iv) enabling perceptual inference operations, (v) using perceptual attention mechanisms for monitoring, and (vi) encoding information in a manipulable medium.

According to Harvard Business Review (Berinato, 2020), visual communication is categorized into four key types (Figure 4.1): (i) idea illustration, (ii) idea generation, (iii) visual discovery, and (iv) everyday dataViz - as shown in the figure below.

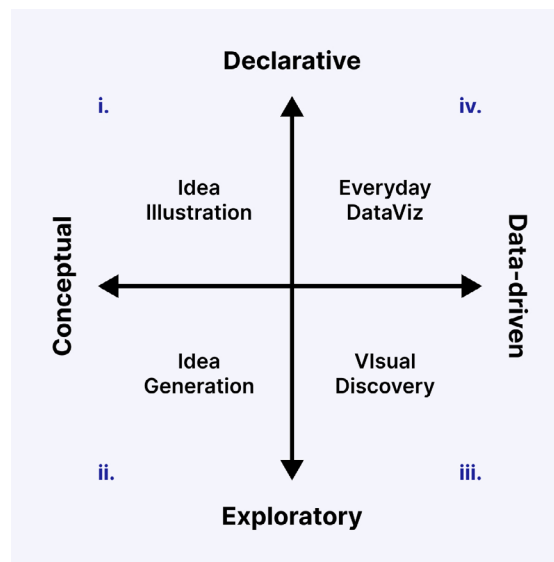


Figure 4.1 - 2x2 definition of the four types of visual communication

Visual Discovery and Everyday DataViz are more closely aligned with the work of Data Science. While visual discovery helps data analysts, data scientists, and other data professionals with the two main objectives of communicating relevant information and exploring patterns, trends, and anomalies within a dataset, everyday dataViz supports the subsequent storytelling after a new insight has been found.

Data Visualization is one of the steps of a data science process, which states that after data has been collected, processed, and modeled, it must be visualized for conclusions to be made (Data Visualization Definition and Examples, (n.d.)). As previously mentioned in Chapter 2.1., the visualization process is divided in five steps that are briefly explained below (Maças et al 2021).

### 1. Definition of the problem and target

In this first step of the visualization process the visualization designer must define the domain and the target users and learn about their goals and what kind of data that



they are dealing with. After that, they must acquire and clean the data. At the end of this step, it's created a list of detailed questions or actions borne by the target users.

## 2. Translation of the data

In order to obtain abstract data and tasks, it is crucial to translate the problems and data of the domain into an abstract description that is in the vocabulary of data visualization.

When it comes to data abstraction, there are different types of data representation: tables, networks or spatial. The attribute types can be categorical (also nominal or qualitative) or ordered (ordinal or quantitative). The data can be ordered in a sequential, diverging, or cyclic mode. With task abstraction, there is a range of questions to be answered:

- Why is the user using visualization?
- What questions need to be answered?
- What problem is being solved?
- What are decisions made?
- What outcomes are desired?
- What is to be told?
- What tasks should the viewer be able to perform?

## 3. Designing visual encodings and interaction

At this point of the process it is crucial to conduct a brainstorm of ideas to design visual encoding and interaction to support data and task abstraction. The best way to do so is by sketching and prototyping. The visual encodings must be arranged according to a spatial organization, layout, and expressiveness and the data type must be mapped to visual marks and properties. To constitute a favorable interaction with the user, there are high-level interaction techniques, such as overview plus detail, focus plus context, panning and zooming, direct manipulation, linking and brushing, etc.

## 4. Implementation

Next to the definition of the desired design, is the development and implementation of the visualization model following the visualization pipeline, represented in Figure 4.2, which describes the process of creating data representations.

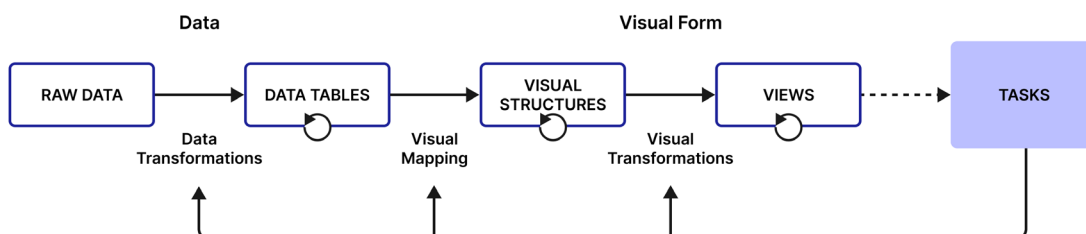


Figure 4.2. - Visualization pipeline inspired in Card et al, 1999

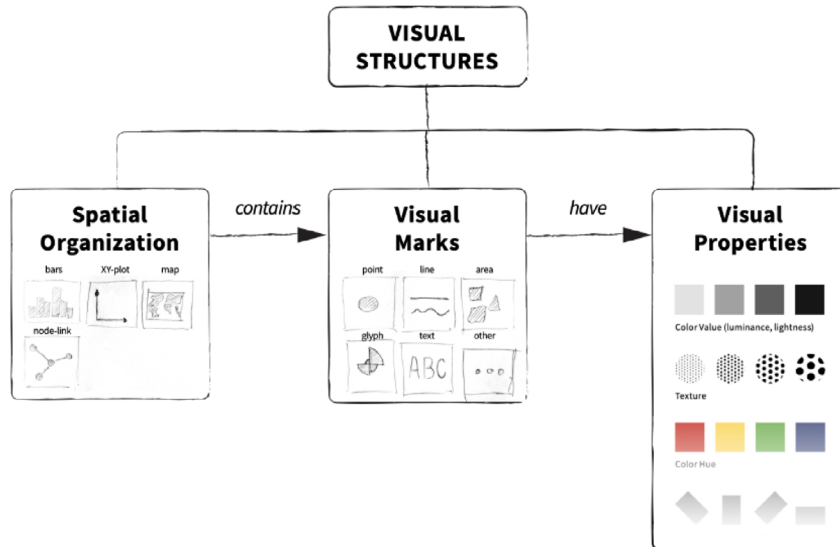


Figure 4.3 - Visual Mapping structures

The first phase - preprocessing phase - consists in data transformation where raw data is converted to data tables. The next phase covers visual mapping, i.e. the mapping of data values into visual variables and suitable graphical forms, taking into account whether the visualization addresses the capabilities of the human visual system. The visuals can assume three different structures (Figure 4.3): (i) spatial construction, (ii) visual marks, and (iii) channels.

The third phase consists in generating a visual representation of the data on the screen. One of the concerns is the limited screen space for the visualization, however, those can be managed through techniques previously mentioned such as zooming/panning, scrolling, focus plus context, and magic lenses. In the phase of task execution, the main concerns are related to perception and psychophysical matters. In the interaction stage, there is a loop between the feedback and the system. The reason for this circumstance is due to the possibility of re-filtered data, remapped graphical variables, and panned view.

## 5. Validation

The final step of the visualization process is the validation, which can depend on the context. The validation results are based on performance, interactivity, and functionality.

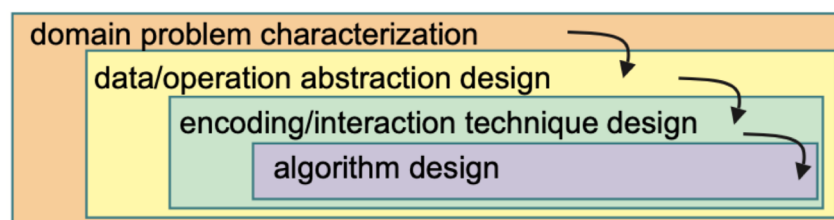


Figure 4.4 - Nested Model (Munzner, 2009)

Taking into consideration the Nested Model (Munzner, 2009) (Figure 4.4), the validation process examines four levels of design problems, these being domain problem characterization, data/task abstraction design, encoding/interaction technique design, and algorithm design.

### **Domain problem characterization**

At the initial stage, visualization designers should familiarize themselves with the tasks and data relevant to the intended users within a specific domain. Each domain typically possesses its terminology for describing data and issues, and an established workflow for utilizing the data to address challenges. It is imperative for the designer to create a tool for this audience to have a clear comprehension of their issues. Despite its apparent simplicity, designers sometimes take shortcuts by relying on assumptions instead of engaging with the actual users. Furthermore, even with access to domain-proficient users familiar with the domain's language and workflow, extracting system requirements is a challenging task. The outcome of characterizing the domain workflow frequently involves a comprehensive set of inquiries or actions performed by the users on a diverse range of data.

### **Data abstraction design**

The abstraction stage involves translating domain-specific issues and data into a more general, computer science-oriented language. This stage aims to provide the necessary inputs for decision-making in visual encoding, using operations and data types. Operations refer to tasks that are not domain-specific, while the transformation of raw data into formats suitable for visualization techniques is also a key focus. The aim is to select appropriate data types to effectively represent the problem, often necessitating the conversion of raw data into modified formats. While this stage can be challenging, it is crucial for achieving effective visualization (Munzner, 2009). Designers frequently overlook the preceding problem characterization stage and quickly assume the first abstraction is optimal, rushing to the visual encoding level. Explicitly defining problems using generic operations and data types can prompt more thoughtful design. It's important to note that this design process is rarely strictly linear.

### **Visual encoding and interaction design**

The third level is designing the visual encoding and interaction. Considerable focus has been directed towards the creation of visual representations in the foundational literature of information visualization. Munzner (2009) takes an approach by addressing visual encoding and interaction as interdependent aspects, rather than separated. She emphasizes that discussions regarding the design of this level, particularly in problem-driven visualization papers like design studies, are explicit and coherent.

### **Algorithm design**

Finally, the innermost step involves the development of an algorithm capable of autonomously implementing the visual encoding and interaction designs. According to Munzner (2009), the challenges associated with algorithmic design are not exclusive to visualization and are extensively explored within the field of computer science.

### 4.1.1. Design principles for visualization and data design

The significance of data visualization in effectively conveying intricate insights concealed within data is vital. While experts in data science and analysis possess the acuteness to extract fundamental insights from complex visual representations, this proficiency might elude high-level business stakeholders, marketers, or individuals without specialized knowledge. Therefore, the demand for proficient data visualization that bridges this gap is more pronounced than ever. Crafting an efficient data representation requires a balance between data accuracy and visual appeal. Nevertheless, a shortfall often arises among data scientists when confronted with the intricacies of design and aesthetics inherent to data visualization. To construct a visualization that optimally imparts the data's message, some design principles can be followed (Spring, 2019).

In the context of this dissertation, we take special interest in Tufte's Design Principles (Tufte, 2001) for visualization, more specifically, the Principles of Graphical Excellence and Graphic Integrity. Besides that, we take into consideration some rules of Data design.

Tufte describes Graphic Excellence as graphics that consist of complex ideas that are communicated with clarity, precision, and efficiency. According to the principles, graphical presentations should fulfill several key objectives. Primarily, they should accurately depict the data, guiding the viewer's focus toward understanding the information itself rather than getting caught up in aspects like methodology, graphic design, production technology, or other extraneous factors. Ensuring that the data's message remains unaltered is essential, to avoid any distortions.

Moreover, these displays should efficiently accommodate numerous numbers within a limited space, promoting coherence in expansive datasets. They should also facilitate comparisons between distinct data points, directing the viewer's gaze to discern differences. To comprehensively unfold the data's insights, graphical displays must cater to various levels of granularity, spanning from a broad overview to intricate details. Furthermore, these presentations ought to serve a clearly defined purpose, whether it's providing descriptions, facilitating exploration, tabulating data, or even serving decorative functions. Lastly, their integration with statistical and textual descriptions of the dataset should be seamless, ensuring a comprehensive and unified portrayal of the information.

That said, the Principles of Graphic Excellence are described in five sentences which we quote below (Tufte, 2001):

- Graphical excellence is the well-designed presentation of interesting data - a matter of substance, of statistics, and of design.
- Graphical excellence consists of complex ideas communicated with clarity, precision, and efficiency.
- Graphical excellence is that which gives to the viewer the greatest number of ideas in the shortest time with the least ink in the smallest space.

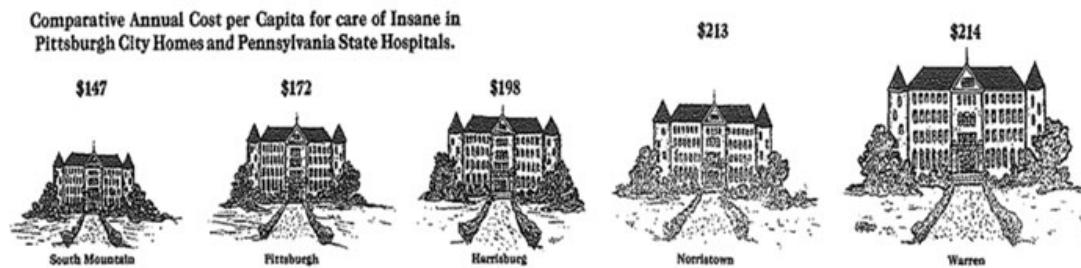


Figure 4.5 - Pittsburg Civic Commission, Report on Expenditures of the Department of (Graphical integrity, (n.d.))

- Graphical excellence is nearly always multivariate.
- Graphical excellence requires telling the truth about the data.

The Principles of Graphical Integrity serve to assist in the creation of meaningful and accurate visual representations of data. Rooted in the philosophy that data visualizations should faithfully convey the essence of information while avoiding distortion, these principles underscore the importance of fostering a genuine and unmediated interaction between the viewer and the data (Tufte, 2001).

Graphical integrity is more likely to result if the following principles are taken into account: (i) proportional representation, (ii) clear labeling, (iii) data variation, (iv) equal number of information, and (v) contextualization (Tufte, 2001).

### Proportional representation

The representation of numbers, as physically measured on the surface of the graphic itself, should be directly proportional to the numerical quantities represented. Figure 4.5 is an example of a wrong practice of proportional representation.

### Clear labeling

The designer should use clear, detailed, and meticulous labeling to counteract any potential graphical distortions and ambiguity. Data interpretations should be elaborated directly on the graphic, and labels provided for significant data events to enhance clarity and understanding. Figure 4.6 (left) is an example of a visualization that does not contain clear labeling.

### Data variation

It should be shown data variation, not design variation, and consistency with the design across the visualization. Figure 4.6 (right) reveals inconsistency in the visualization.

### Equal number of information

The number of information-carrying (variable) dimensions depicted should not exceed the number of dimensions in the data. Figure 4.7 (left) is an example of a wrong practice of equal number information.

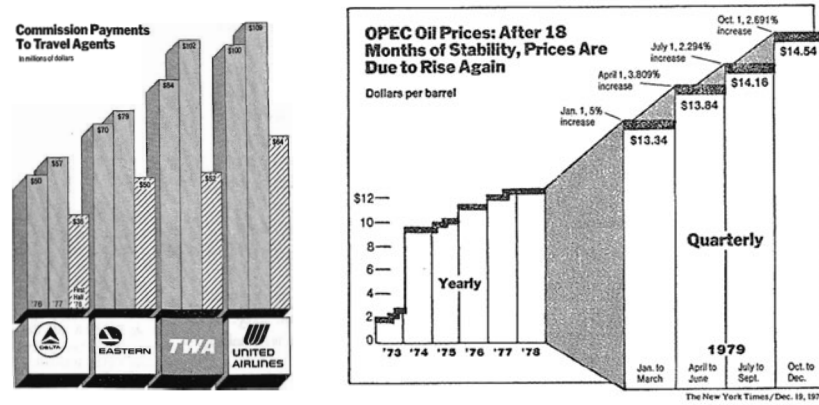


Figure 4.6 - New York Times, August 8, 1978 (Tufté, 2001) (left); New York Times, December 19, 1978 (Graphical integrity, n.d.) (right)

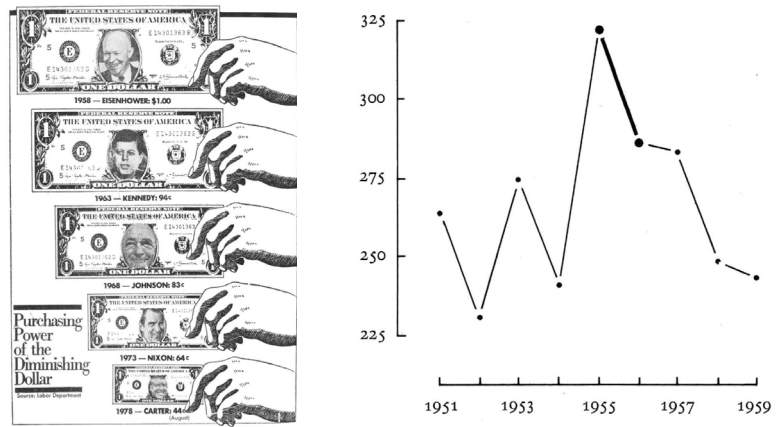


Figure 4.7 - Washington Post, October 25, 1978 (Tufté, 2001) (left); Connecticut Traffic Deaths, 1951-1959 (Tufté, 2001) (right)

### Contextualization

Context plays an essential role in maintaining graphical integrity, that is why the designer must provide sufficient context to the viewer, present the data and make observations in relation to the context. Graphics must not quote data out of context as shown in Figure 4.7 (right), where there is no legend, which means there is no context for the viewer.

In addition to Edward Tufté's principles, there are several aspects to be aware of that are addressed in Data Design to ensure graphical elegance that is often found in simplicity of design and complexity of data (Maçãs et al, 2021).

Data design involves a judicious combination of fundamental structures, namely sentences, tables, and graphics, each chosen according to the data's nature. Proper format selection, such as employing line charts for time-series data, enhances accessibility to intricate concepts (Figure 4.8).

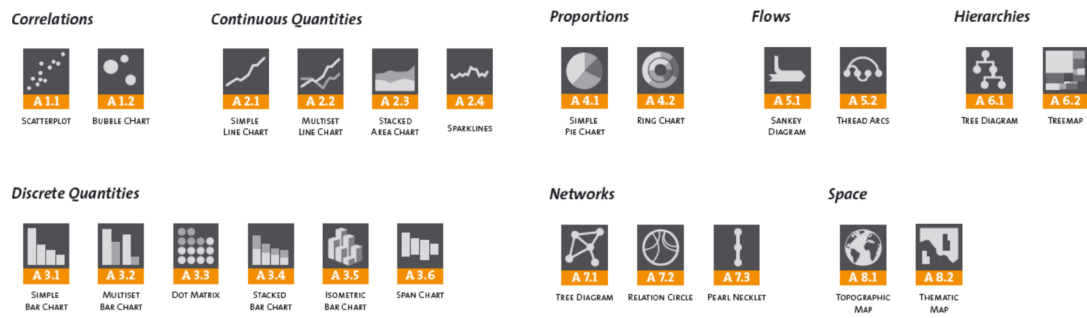


Figure 4.8 - The Form of Facts and Figures, Christian Behrens, 2008

Integration of text, legends and labels seamlessly simplifies complexity. To ensure clarity, labels and legends play vital roles elucidating encodings, while text describes what is in the data.

When incorporating color, it should accurately reflect data distinctions, avoiding problematic color schemes and considering visibility for those with color blindness. A balanced scale approach, with natural increments and sensible grid scales, contributes to readability while adhering to principles like maximizing the data-ink ratio and defining graphic boundaries based on data extremes.

Proportionate graphical elements and balanced fonts enhance visual cohesion and legibility, favoring horizontal over vertical orientations, unless data dictates otherwise. Emphasizing cause-effect relationships—placing independent variables on the horizontal axis and dependent variables on the vertical—strengthens clarity. Finally, data richness is key, where the quality and quantity of information ensure a comprehensive and precise representation, maximizing the insights conveyed.

To conclude, Tufte’s principles advocate for designs that prioritize substance over style, guiding viewers to engage deeply with the data’s inherent significance rather than becoming entangled in design intricacies or superfluous embellishments. Through a careful balance of clarity, coherence, and purpose-driven presentation, these principles aim to ensure that graphical displays genuinely uphold the integrity of the underlying information, fostering a transparent and insightful connection between data and interpretation. On the other hand, it is important to pay attention to all visual aspects such as smart use of colors and patterns, which can elevate the visualization by always trying to choose the ideal format for the data, keeping a simple and balanced design for easy interpretation, incorporating interactivity and focus on the essential.

## 4.2. Visualization in Machine Learning

ML models are trained with extensive amounts of data with the goal of increasing efficiency while making optimal decisions at the same time. The responsibility attributed to ML models implies the need to make them as transparent as possible, otherwise they do not convey trust (Jha, 2022).

The increased popularity of big data and data analysis projects have made visualization more important than ever. Companies are increasingly using ML to gather massive amounts of data that can be difficult and slow to sort through, comprehend and explain. Visualization offers aid in the process by presenting information to business owners and stakeholders in ways they can understand. In addition, when a data scientist is writing advanced predictive analytics or ML algorithms, it becomes important to visualize the outputs to monitor results and ensure that models are performing as intended (Brush et al, 2022).

In the ML process, as shown previously in the previous chapter, there are different possibilities for the use of visualizations during the ML process. Generally, visualization falls within the data exploration stage where it is used to visualize the data and the results of the ML process so it is easier to identify patterns, relationships, and anomalies in the data, i.e. visualization is positioned before the deployment stage of the ML model. However, visualization also plays a key role in the next stage after the creation of the model (Scalco, 2021).

Visualization of ML algorithms assumes a fundamental role for understanding, presenting and interpreting the results of these algorithms after the construction of the model. There are several types of visualizations that can be used to represent the results of ML algorithms that include graphs, diagrams, tables, and all kinds of visualizations that allow data to be presented and explained thoroughly and attentively (Scalco, 2021).

When choosing a visualization technique, it is important to choose the appropriate visualization for the problem and data in question and to make sure that it is accurate, informative, and easy to understand. The use of interactive visualizations can be particularly beneficial, as it allows the user to explore and manipulate the data and models in a way that static visualizations cannot (Layer Software GmbH, 2023).

To reinforce everything that has been stated so far, and taking into account the reference to the blog of the visual designer Abhishek Jha's (Jha, 2022), the importance of visualization in ML is connected to:

#### **Explainability**

Visualization is crucial to have a better understanding about the ML model's decision-making process. The difficulty of explainability is more evident in the case of Neural Networks.

#### **Debugging & Improvements**

The development process of a ML model can be speeded up with the aid of visualization by assisting in the task of finding the optimal combination of hyper-parameters, which can be challenging.

#### **Comparison & Selection**

Instead of choosing the best model out of an ensemble of well-performing models, we can visualize parts of the model that offer the highest accuracy or lowest loss while ensuring the model does not overfit by designing a framework that can compare different snapshots of the same model.



### Teaching Concepts

Interactive platforms can be designed where users play around with multiple datasets and toggle parameters to observe the effects on the model's intermediate states and outputs. This could seriously help to build intuition about how models work.

In ML, data visualization is the area of knowledge that is concerned with seeking appropriate graphical representations of a set of data, aiming to assist in the processes of analysis for a better human understanding.

Data visualization is one of the main tools used to analyze and study the relationships between different attributes, providing a better understanding of the data to be processed. It can be used for descriptive analysis and in the pre-processing of a data set, in the selection of attributes that have a great influence on the expected result, in the very construction, testing, and evaluation of the model (Data Visualization in Machine Learning, (n.d.)). Finally, visualizations are often used to help end users make decisions based on the data, where factors such as validation of the analysis results play a very important role.

#### 4.2.1. Tree-based visualization techniques

**“Creating and evaluating decision trees benefits greatly from visualization of the trees and diagnostic measures of their effectiveness.”**

**Barlow et al, 2005**

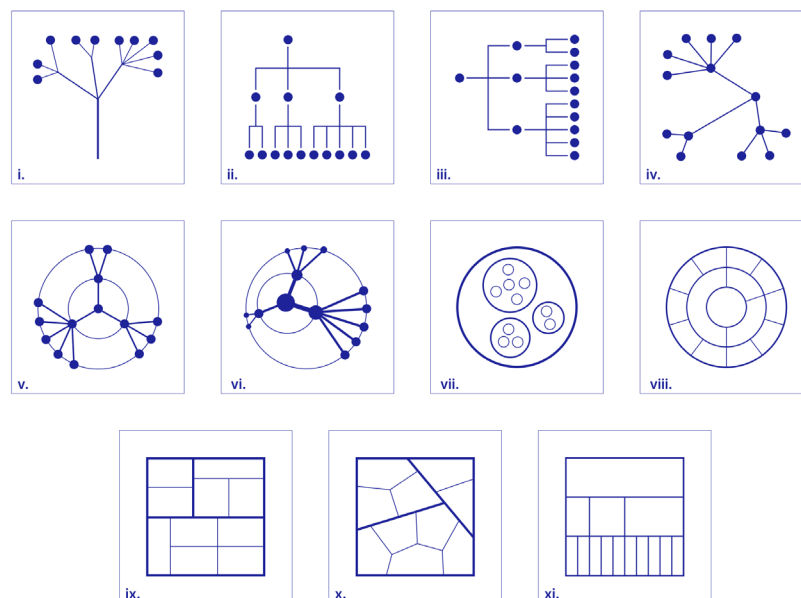
Tree-based visualizations are powerful graphical representations that serve as effective tools for the depiction and analysis of hierarchical structures, relationships, and processes. Rooted in graph theory and data visualization, these visualizations present complex data sets in a manner that simplifies comprehension by showcasing the relationships between individual elements and their hierarchical arrangement. Tree-based visualizations find application in various domains, including biology, computer science, organizational management, and information technology, aiding in decision-making, pattern recognition, and the communication of intricate data insights. This introductory overview explores the fundamental concepts and significance of tree-based visualizations in presenting intricate data structures with clarity and precision.

A decision tree, as previously mentioned, is a simple recursive structure that expresses a sequential process of classification. The simplicity of its rules and the natural tree-based visual representation makes it easier to build a visualization model. Decision tree visualization and exploration are important for two mutually-complementary reasons. Firstly, it is crucial to be able to navigate through a decision tree quickly to find nodes that need to be further partitioned. Secondly, the exploration of a decision tree aids the understanding of the tree. From the visualization, the user gains helpful knowledge about the particular dataset (Teoh et al, 2003).

Hierarchical and radial views are the most known displays of a decision tree, whereas the hierarchical view is consistent with the logic of a decision tree construction and the radial view is mainly applied in displaying object structures. We can divide the hierarchical techniques into two classes of algorithms: **space-filling** and **non-space-filling** (Ward et al, 2015).

Space-filling techniques make maximal use of the display space by juxtaposition. The two most common approaches to generating space-filling hierarchies are rectangular and radial layouts such as tree map and tree ring, respectively, which will be explained further. The most common non-space-filling technique that is used to visualize tree or hierarchical relationships is the node-link diagram. Frequent applications for this diagram can be organizational charts, family trees, and tournament pairings.

There are multiple approaches to visualize a decision tree, as shown in Figure 4.9, such as (i) figurative trees, (ii) vertical trees, (iii) horizontal trees, (iv) multidirectional trees, (v) radial trees, (vi) hyperbolic trees, (vii) circular treemaps, (viii) sunbursts, (ix) rectangular treemaps, (x) voronoi treemaps, and (xi) icicle trees (Lima, 2014).



**Figure 4.9** - Different visualizations of decision trees (Lima, 2014)

### Figurative Trees

Emerging as an immensely prevalent visual classification system, the tree diagram has gradually incorporated the genuine and natural characteristics inherent in its living, biological analog (Figure 4.10, left). It employs the imagery of trunks, branches, and extensions to depict interconnections among distinct entities. While the passage of time has brought about more abstract and artistically adapted renditions of tree diagrams, numerous associated terms such as roots, branches, and leaves persist in common usage. Their range of subjects has only expanded since their first appearance.

## Vertical Trees

The progression from realistic tree depictions to abstract representations was a natural evolution in hierarchical visualizations, with vertical structures emerging as a common choice. Vertical trees maintain a strong semblance to actual trees due to their top-down branching from a central trunk (Figure 4.10, right). While now predominantly seen on digital screens, historical versions spanned larger formats like scrolls and charts, serving as detailed visual aids. These extensive trees were used for in-depth study, often displayed as wall posters. As a preeminent instance of node-link diagrams, the vertical tree finds wide use in taxonomies, organizational charts, family trees, decision trees, evolutionary diagrams, file systems, and site maps.



Figure 4.10 - Genealogical tree of Charles Magius from Paul Veronese, Codex Magius, 1568-73 (left); X-Men Family Tree, Joe Stone, 2011 (right)

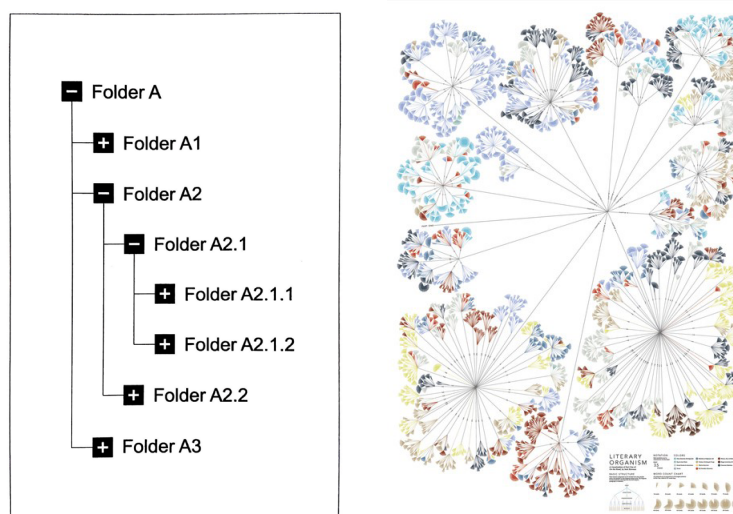


Figure 4.11 - Indented tree, Manuel Lima, 2013 (left); Writing Without Words, Stefanie Posavec, 2008 (right)



## Hyperbolic Trees

As a variation of the radial tree, the hyperbolic tree is a contemporary visualization produced through advanced computer algorithms (Figure 4.12, right). Employing a “focus and context” method, hyperbolic trees highlight specific nodes by centralizing and enlarging them, while relegating less significant connections to smaller sizes nearer the edges. This magnification capability renders hyperbolic trees valuable for presenting and navigating extensive hierarchies within restricted screen dimensions. These visualizations excel in direct manipulation scenarios, making them uncommon in print and largely restricted to their native digital environment.

## Circular treemaps

Circular treemaps are a recent innovation in treemaps that adopts circles rather than rectangles for its organizational scheme (Figure 4.13, left). Each branch or division of the tree is symbolized by a circle, subsequently filled with smaller circles representing sub-divisions. Despite the clear hierarchical arrangement and visually pleasing patterns that circular treemaps offer, their inefficiency in the use of space arises from the empty spaces between cells. This limitation makes them a less effective visualization method, especially when dealing with a substantial number of levels.

## Sunbursts

Sunbursts, also recognized as radial treemaps, tree rings, fan charts, or nested pie charts, present a space-filling visualization employing a radial design (Figure 4.13, right). Similarly to radial trees, sunbursts typically initiate from a central root, extending outward in successive tiers from the center. However, in contrast to node-link structures, sunbursts employ an arrangement of segmented rings and juxtaposed cells. Comparable to treemaps, the size of each cell generally corresponds to a specific quantity or data attribute, while color introduces an additional attribute. Due to their radial layout, sunbursts harmonize well with symmetrical, square spaces. Nevertheless, this radial distortion can have a negative effect on the distinction between layers. Sunburst diagrams have gained particular prominence for representing file systems and genealogical relationships.

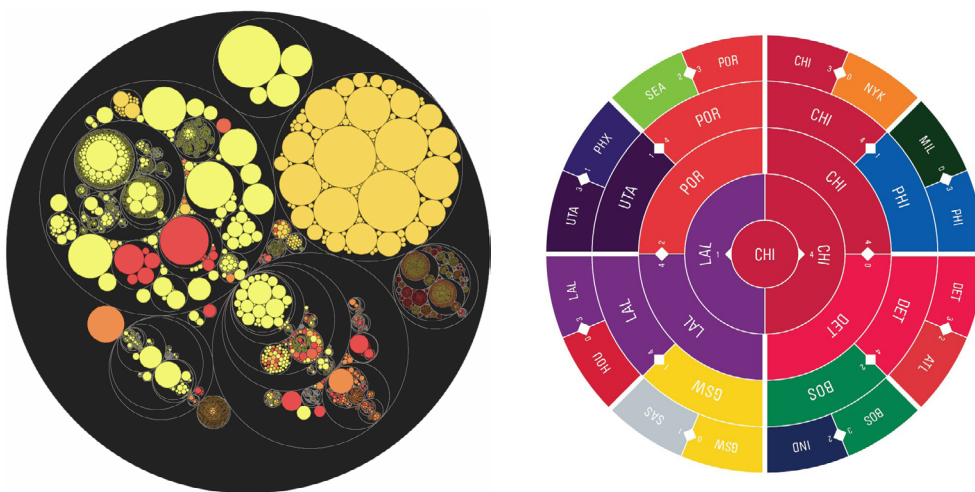
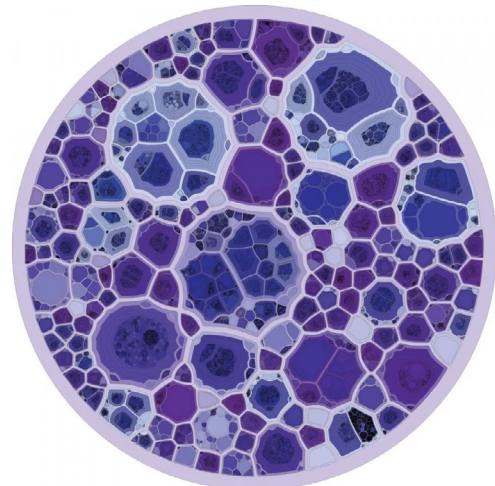
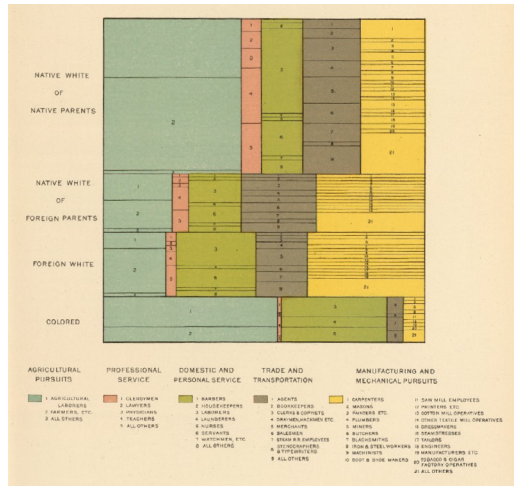


Figure 4.13 - Pebbles, Kai Wetzel, 2003 (left); The Champions Ring, Deroy Peraza, 2012 (right)





**Figure 4.14 - Eclipse Voronoi treemap, Oliver Deussen, 2010 (left); Classification of the Occupations by Race and Nativity, Henry Gannett, 1900 (right)**

### Rectangular treemaps

The rectangular treemap is a space-filling visualization that employs nested rectangles to display hierarchical data (Figure 4.14, left). Each branch is represented by a rectangle which is then subdivided into smaller rectangles to depict its subcategories. The size of each rectangle represents data attributes such as size or length, and colors indicate qualities such as type or class. Treemaps are popular due to their efficient use of space, clear structure, and capacity to show many entities at once while remaining legible. They carry substantial significance in today’s computer science, serving as a notable example of progress in data visualization.

### Voronoi treemaps

Voronoi treemaps are a recent variation rooted in Descartes’ mathematical partitioning, that uses polygonal cells for points in a set of seeds, prioritizing proximity to individual seeds (Figure 4.14, right). Their recursive breakdown allows for hierarchical clustering. Unlike rectangular treemaps, Voronoi treemaps offer adaptable cell configurations, yielding diverse, organic layouts resembling stained glass or natural patterns. This model finds broad use, especially in visualizing file systems and genome data.

### Icicle trees

Icicle trees, also referred to as icicle plots, share similarities with node-link diagrams (Figure 4.15). However, they diverge by employing an adjacency-area approach, using side-by-side rectangles to signify hierarchy. Icicle trees are versatile in adapting to spatial and layout limitations, accommodating either a vertical, top-to-bottom arrangement or a horizontal, predominantly left-to-right orientation. Since icicle trees lack a nesting mechanism for hierarchical tiers, their use of space is less efficient compared to treemaps and similar space-filling methods, which can result in disproportionate representations.

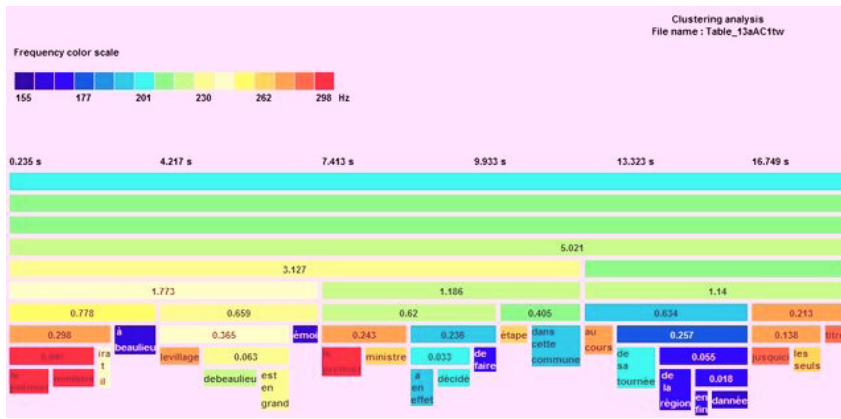


Figure 4.15 - ADoReVA (automatic detection of register variations)

Within these considerations, it is of extremely important to incorporate an overall view of the model and, at the same time, a detailed view of the tree, a navigation support and integration with data visualization.

Although it applies to most data visualization techniques, tree-based visualization benefits greatly from interactivity techniques. Interactions such as zooming, panning and rotation, which are very common in data visualization, allow the viewer to incrementally build up a mental and detailed perspective of the object.

Interactions with graphs can happen in different ways from modifying the structures to interacting with the elements. Most interactions regarding graph elements are linked to the selection operation, where one or more components of the graph are isolated for some action such as highlighting, deleting, masking, moving, or obtaining details. A problem that might occur is when the graph has an agglomeration of elements in a certain region of the drawing, as shown in the example of Figure 4.16, which makes it difficult for the user to select certain elements, therefore exposing the need for other interactions such as zooming as distortion techniques.

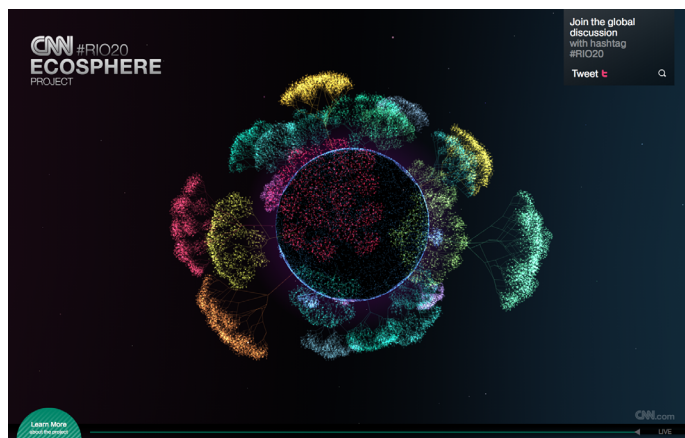


Figure 4.16 - CNN Ecosphere Project, Minivegas and Heimat Berlin, 2011

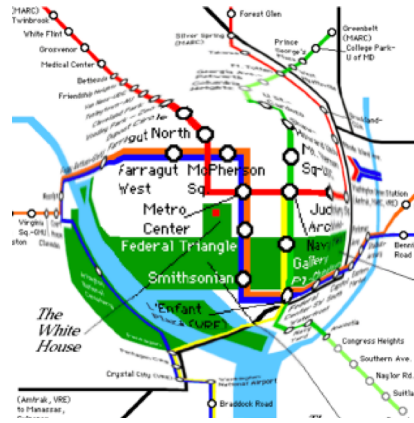


Figure 4.16 - Fisheye lens

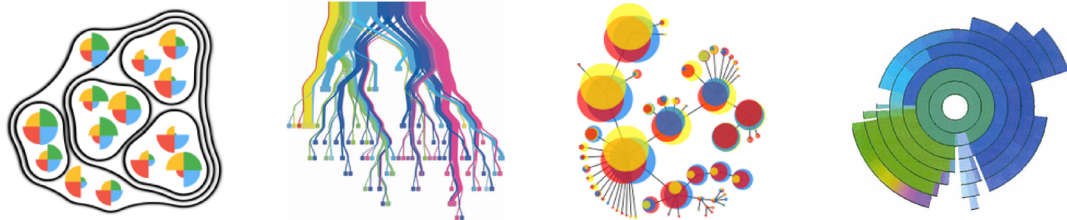


Figure 4.17 - Examples of decision trees visualizations from treevis.net

Through interaction, there is also the possibility of changing the graph structure, for example reordering the branches of a tree which can lead to exposing new relationships. Also associated with the graph structure interaction is the focus+context technique where a selected subset of the structure is presented in detail. An example of this technique is the fisheye lens (Figure 4.17) where the parts of the visualization falling within a focal region are enlarged, while the remaining parts are proportionally shrunk to maintain their presence in the display.

There are situations where removal of certain sections of a graph aids in the visualization process. For example, once a branch of a tree has been thoroughly investigated, the user might have the need to remove it from the display in order to clear the space for unexplored regions.

To conclude, all the techniques and interactions mentioned thus far, have contributed to an evolution in the visualization of decision trees, becoming more and more elaborate, interactive and aesthetically innovative as can be seen in the treevis.net platform - a Visual Bibliography of Tree Visualization (Figure 4.18).



### 4.3. Tree visualizations

In this section, we discuss the related work regarding the visualization of random forests and decision trees. The different approaches presented below are organized by relevance for the development of the project in which this dissertation is inserted, with the last one being the one that best fits the context of this dissertation. While the first two platforms only present the visualization of a decision tree more generically, the third presents the decision tree innovatively and creatively. The last two options offer solutions that visualize more than one tree, with only the last one specifically mentioning the visualization of a random forest.

#### 4.3.1. Baobab View

This method of visualizing decision trees is not only scalable but also enables experts to inject their domain knowledge into the construction of decision trees. It relies on visualization, interaction, and algorithmic support to enable the users to enhance the ML method. The name of this project comes from the similarity between some of the tree visualizations and the shape of an African Tree called Baobab (Van Den Elzen et al, 2011).

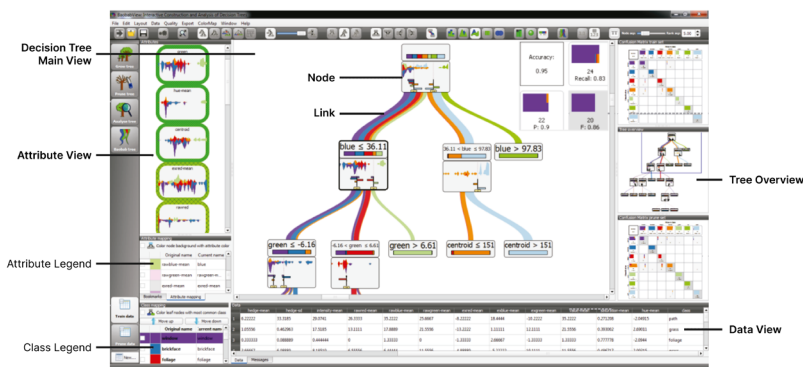


Figure 4.18 - Interface of the interactive decision tree construction software Baobab View

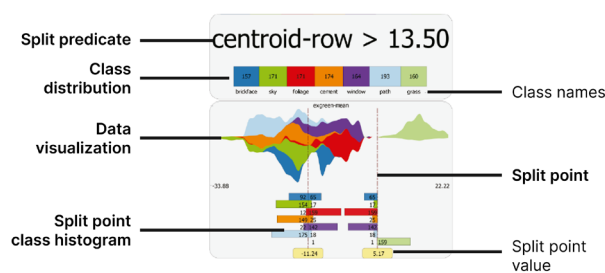


Figure 4.19 - Close-up of the node information

As can be seen in the Figure 4.18, the interface consists of a main view of the tree and a series of small windows around it that constitute views of other features. The main view contains a node-link diagram composed by nodes and links used to convey the node size. To visualize the distribution of classes, the large link is divided into bands, each colored according to the class and given the respective proportional width.

The nodes are represented by rectangles that contain relevant information inside, as represented in Figure 4.19. The node information incorporates (i) the split predicate displayed in plain text, (ii) the class distribution of the set of items in a determined node, (iii) a Streamgraph to visualize the attribute class values and distributions, (iv) the split points with class histograms with colored bars according to the classes to present the absolute values.

One of the small-scale windows provides a contextual overview of the main view of the tree and navigation support. The attribute view shows data visualization of each attribute for a selected node. In addition, there is also a data view that presents the data in a table. As well as in any other project of visualization, the interaction of the user with the is key for a better understanding of a decision tree. In this case, given that it is convenient to use different layouts depending on the operations, the authors implemented different options for the user to control the layout of the tree and the visualization of the nodes and animation techniques to guide the user throughout the transition. Besides these characteristics, the software incorporates close-up inspection via zoom-pan options and a separate window previously mentioned.

### 4.3.2. EMTree Results Viewer

EMTree was designed to help analysts build and understand complex decision trees by visualizing the partitioning of cases and visualizing model diagnostics, which in turn can help the analyst comprehend the predictions of a model and help the analyst assess the reliability, respectively (Barlow et al, 2005).

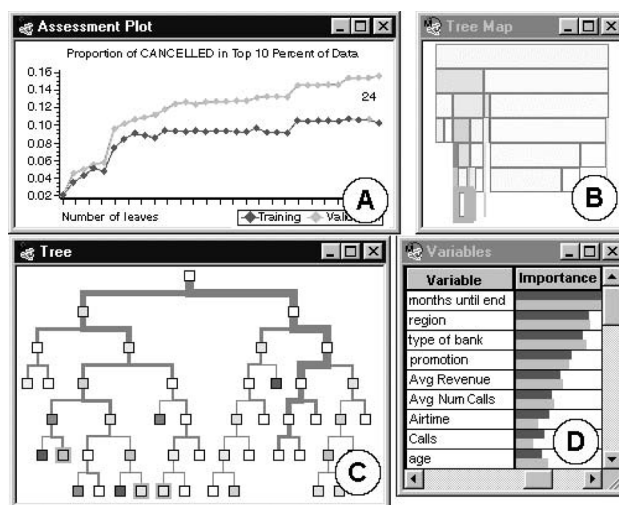
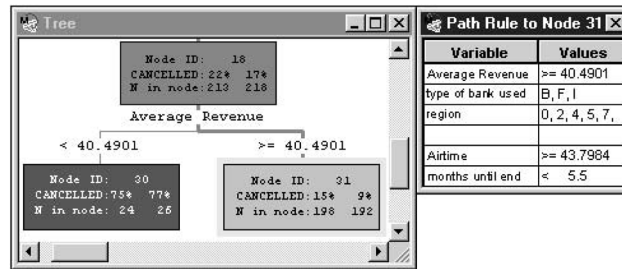


Figure 4.20 - Screenshots of the application taken from Barlow et al (2005)



**Figure 4.21** - Display of the rules as part of the tree (left) and in a separate window (right)

The application has fourteen linked views available to the user. In Figure 4.20, there are only four views presented Windows A, B, C, and D. Window A contains the assessment plot that shows the available subtrees and allows the user to see details about a subtree when selecting the related point in the plot which updates the other views. Window B displays a compact view of the tree topology, representing node size through width, and was designed to be used as a navigation tool for larger trees. Window C shows a traditional tree view and information about the model. Finally, window D contains a table that presents a list of variables and their relevance.

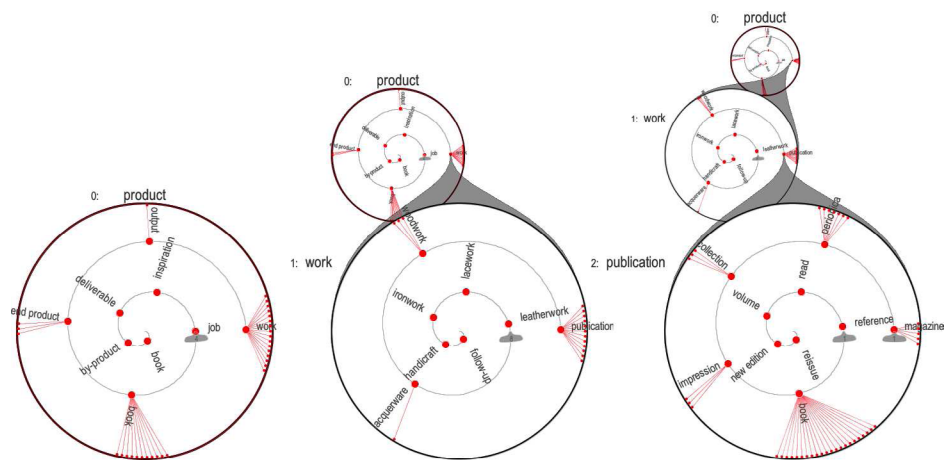
One intriguing characteristic of this application is the relationship between one window and the other. For example, if the user selects a variable in window D, the software highlights the nodes using that variable in windows B and C. To facilitate the process of finding a node deep in the tree, the authors established a relationship between windows B and C whereas by selecting a node in the compact view (window B), the main view (window C) moves that same node to the center of the window for a better visualization.

Other interesting functionalities to take into account are the use of line thickness in the branches proportional to the square root of the number of cases in each branch, the use of color to highlight the nodes with the main statistic of interest, and the use of two methods to display rules (Figure 4.21).

To confirm if the rules make sense and analyze if the model is reliable, the system uses Leaf Statistic Bar Chart that compares the validation data with the training data. For a model to be reliable, the bars representing the validation data would increase similarly to the training bars.

### 4.3.3. PaisleyTrees

PaisleyTrees presents an innovative approach to size-invariant tree visualization (Etemad et al, 2014). This technique merges node-of-interest (NOI) emphasis and tree-cut displays to facilitate rapid tree navigation, all without necessitating zooming or panning actions. This visualization method excels with trees of various depths and breadths, ensuring the clarity of displayed elements. To accomplish these benefits, the approach employs a hybrid layout inspired by traditional Paisley patterns, merging node-link, nested, and adjacency-based tree layout strategies, as demons-



**Figure 4.22** - Drilling into a PaisleyTree, K.Etemad, D.Baur, J.Brosz, S.Carpendale and F.F.Samavati, 2014

trated in Figure 4.22. This layout, which combines depth and breadth truncation, allows for efficient tree interaction.

At its core, this technique employs circles to symbolize nodes, arranged from bottom to top, with the NOI at the bottom, the root node at the top, and intermediary nodes in between. Circles and their contents progressively diminish in size higher up the spiral, indicating their distance from the NOI. The visualization encapsulates two levels of a node’s descendant sub-tree within each circle. Direct children are positioned as dots in a nested layout on a spiral within the circle, while grandchildren are situated on the circle’s perimeter using a node-link layout. To manage visual clutter, children are categorized into leaf and non-leaf groups. Leaf children appear on the spiral’s inner ring, while non-leaf children are positioned on the outer ring. A PaisleyTree begins with a single node circle representing the root node. Clicking on a non-leaf node triggers an expansion, updating the current NOI and providing a more detailed view of the tree’s structure.

To conclude, PaisleyTrees introduces an inventive and adaptable approach to tree visualization, merging node-of-interest focus and tree-cut displays for efficient navigation. Its innovative hybrid layout, inspired by Paisley patterns, optimizes space and interaction, while utilizing circles for node representation ensures clarity, resulting in a user-friendly design and applicability across diverse contexts.

#### 4.3.4. PansyTrees

PansyTree is a visualization technique for representing and comparing for both hierarchical structure and node attribute values in detail by merging multiple trees (Dong et al, 2020). It uses a tree metaphor design inspired by nature to allocate attributes into a flower-like icon by combining each node with the same name among three structures and compressing them into petal sets by linearized order. Distinguishing different structures by color encoding, it uses the petals’ height to

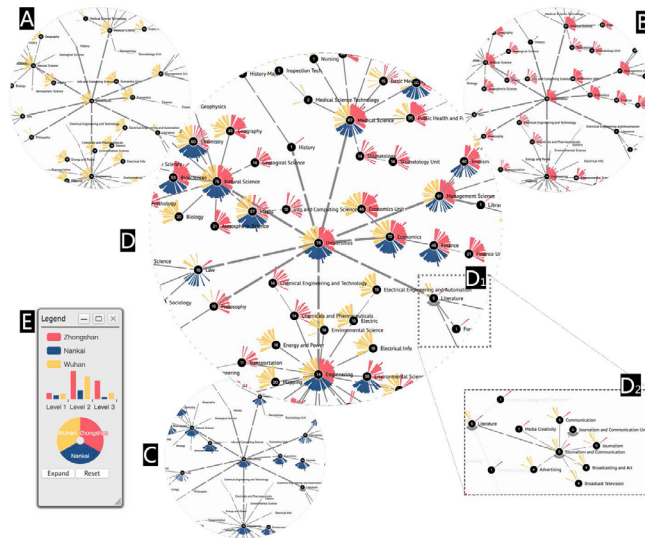


Figure 4.23 - PansyTrees merging three datasets from the Chinese National College Entrance Examination (CNCEE) results in 2017

Names	Elements	Descriptions
Flower center		Node in Pansy, named "node quick-view"
Petal/Petal set		Colored petals in Pansy, represents attribute value by their height
Sepal		Sepal in Pansy, represents no attribute values, but occupies hierarchies
Pansy		Pansy represents node element and its attribute values by colors

Names	Examples	Descriptions
Trunk		Links between root and its children
Branch		Reduced links' width level by level
Animated cursor		The flow speed of links, called "animated cursors" between parent and child described as: $V_{Link} \propto \frac{\ C_1 - C_2\ }{C_1}$

Figure 4.24 - Description of element design (a) and overview element (b)

map attribute values. Additionally, a force layout is added to interactively represent the relationships between each hierarchy. Figure 4.23 shows how the creators merge three different trees, highlighting the data in each with different colors.

To better understand data, we investigate the links with animation and different width to reveal relationships; further, introduce the concept of growth phases to correspond to stages of nodes with different petal quantity. The data from the three nodes are consolidated into a single entity by arranging their corresponding attributes in a clockwise sequence. The height of the petals indicates the extent of the positive value, and the petals in different colors obey the same located order (Figure 4.24). Besides all these functionalities, PansyTree implements several interactions to improve the design of the platform such as structural cue highlighting, conditional permutations to filter hierarchies and collapse/expand nodes to show details.

This project draws attention to the concept of merging three decision trees, making us consider various possibilities. For example, representing a random forest through a single tree that contains a merge of all the trees that make up the random forest. The exploration of this concept became one of the bases for the construction of our visualization tool.

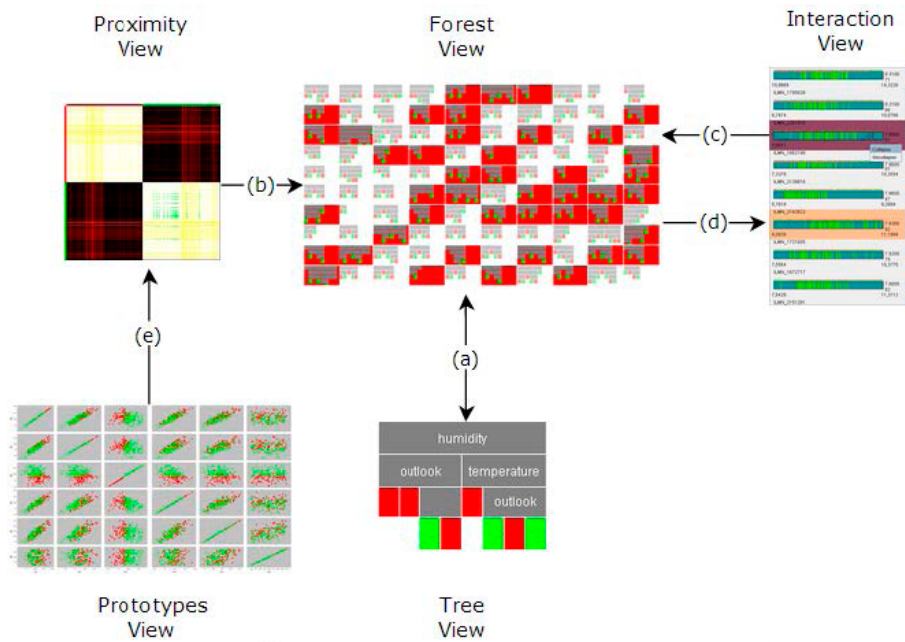


Figure 4.25 - Relations between the main panels of the ReFINE, Kuznetsova, 2014

### 4.3.5. ReFINE

ReFINE is a tool that aims to provide insights into the inner workings of random forests (Kuznetsova, 2014). The Random Forest Inspector is designed to help users understand how individual decisions are made within a random forest model, how different features contribute to predictions, and how the model behaves overall. This can be particularly valuable for gaining insights into model behavior, identifying sources of error, and improving model performance.

This tool enables users to engage with Random Forest functionalities, revealing their sources. The crucial link lies in establishing the connection between these functionalities and the constructed forest, aiding users in comprehending their derivation.

Although the main view is composed of the random forest, there are four more views including a proximity view, an interaction view, a prototypes view, and a tree view. Through the Interaction View, users can observe the decision-making sequence within the forest, the Proximity and Prototypes Views provide insight into the associations between diverse variables and the classification process, and the Tree View gives a detailed visualization of an individual tree represented by an icicle tree. All this functionalities can be seen in Figure 4.25.

## 4.4. Reflection and conclusion

Through the state of the art, we can get a small notion of the vast dimension of the field of data visualization and its processes and techniques in order to successfully develop a good visualization model for ML algorithms, always taking into account the context in which the problem is inserted.

It is worth noting that there is more research on decision tree visualization frameworks, as random forest visualization does not appear to be as explored in comparison. This demonstrates the potential that the tool to be developed for this dissertation has in terms of utility. Although the works presented in the previous chapter (Chapter 4.3) may not be entirely in accordance with the final object of the dissertation, they are related to it in the way that they contribute with small characteristics that we wish to implement in the practical project. Mainly regarding interactivity, in general, the projects provide extremely interesting functionalities to include in the project to be developed. For example, the case of the EMTree Results Viewer platform that presents two different views of the decision tree through the use of distinct tree-based visualization techniques. The presence of two types of visualization, the main one being the random forest and then the decision tree in detail, as is the case with the ReFINE platform, is another key functionality for the project to be developed.

It was also noted the existence of diverse techniques to visualize the information within each node of the decision tree and even the node itself as can be observed, for example, in the BaobabView proposal.

In general, all the works presented, despite having intriguing characteristics, left something to be desired in what concerns the aesthetic portion of the system, even taking into account that some have been developed more than a decade ago. In the development of the practical goal, we intend to use a more polished design, which somehow facilitates the visualization of information.





## 5. RaVi - Random Forest Visualizer

This chapter presents our solution for the visualization of a random forest model, developed after studying the main aspects of Random Forests, Decision Trees, and Data Visualization. Our tool—named RaVi, which stands for Random Forest Visualizer—is a web-based application composed of several panels, being the most important ones, the panels with the random forest and decision tree visualizations. In addition to these visualization models, we added other common visualization techniques to assist in data interpretation.

This section covers the objectives of the project developed in the context of this dissertation, the tasks and requirements outlined for the development of the platform, the work done on analyzing and structuring the data, the visualization models, and the implementation of the web application. Furthermore, we present a use case in which we describe the interaction with the application and the information that can be drawn from the visualization models. In addition to the use case, we also exhibit the results of user tests carried out at the end of the implementation.

### 5.1. Objectives

The main goal of this work is to use Data Visualization techniques to aid researchers and end-users in the analysis and improvement of ML models. With this in mind, we selected a use case where AI Models, more specifically tree-based models, are used to define which contact channel (phone call, SMS or e-mail) is more appropriate to use for future contacts, according to the clients' characteristics. Our purpose is to analyze the data obtained by the ML model and choose the most appropriate visualization techniques to transmit the data.

That said, we aimed to implement two different visualizations, one for the random forest and one for the decision trees, and develop a way of representing the most important data for both the researchers and other end users. Of this data, it was important for the user to be able to see which feature appears most often, the relationship between features, how many times a particular feature is used, what level it is at, and the samples in a particular node. In the next section, we present the tasks and requirements necessary to fulfill these objectives.

## 5.2. Requirements and tasks

Before implementation, it is fundamental to understand what functionalities we want to include in the application to give the best user experience possible according to the intended purpose of the application. For the tasks, we have the actions that we want the user to be able to perform, and for the requirements we present what we can implement so that the user can accomplish them. It should be noted that during the development of the project and after an evaluation of the previous tasks and requirements, some changes were made to be in accordance with the objectives for the realization of the project.

### Tasks

**T1:** The user must be able to access a visualization of the random forest taking into account all the trees

**T2:** The user must be able to manipulate the radial tree structure, i.e. move the tree, change its scale, and change the number of nodes displayed

**T3:** The user must be able to visualize the features that appear more often in each level

**T4:** The user must be allowed to inspect a decision tree individually

**T5:** The user must be able to select trees that contain a certain feature

**T6:** The user must have access to the node's details of a decision tree

**T7:** The user must be able to compare feature importances in each tree and in the random forest

**T8:** The user must visualize the relationship between features

**T9:** The user must be allowed to visualize how many times a certain features is used

### Requirements

**R1:** The tool must present a visualization of the random forest

**R2:** The tool must provide techniques such as zoom and filter to manipulate the radial tree

**R3:** The tool must show the features present in each level

**R4:** The tool must provide the visualization of individual trees

**R5:** The tool must show a set of trees with the same feature

**R6:** The tool must exhibit the information about each node

**R7:** The tool must show the importance of each feature

**R8:** The visualization of the radial tree must illustrate the relations between features

**R9:** The platform must indicate the number of times a feature is used

The requirements present characteristics that were strongly inspired by the works found during the state-of-the-art survey. In the following chapters, we take a look at the work carried out to fulfill the goals previously set.

### 5.3. Data analysis and structuring

Data analysis and structuring is a crucial step in developing visualization models. Proper analysis of data ensures that insights are derived, relationships are uncovered, and trends are identified. Moreover, well-structured data acts as the raw material that visualization tools mold into meaningful and interpretable representations. The analysis of the data guides the selection of appropriate visualization techniques, while structured data allows those techniques to convey insights accurately and persuasively. Ultimately, the synergy between data analysis and structuring underpins the creation of compelling visualizations that amplify understanding and inform decision-making (Gill, 2023; Yse, 2021).

The data given to us, allowed the access to information that helped us choose the most appropriate visualizations, including the size of the data set, the depth of each decision tree, and the number of features used. In addition, we were able to understand which information was essential for the visualization.

```
{
  "classes": [
    "email",
    "sms",
    "voice"
  ],
  "feature_names": [
    "cod_sexo_calc_bi",
    "media_chamadas_por_semana",
    "media_duracao_por_semana",
    "media_kb_por_semana",
    "media_sms_por_semana",
    "meses_antiguidade",
    "dsc_tipo equip",
    "dsc_tarifario",
    "bundle",
    "flag_ac",
    "flag_sdd"
  ],
  "feature_importances": [
  ]
}
```

Figure 5.1 - Original data structure: random forest properties

The file provided, which was transformed into a JSON (JavaScript Object Notation) file, had the structure shown in Figure 5.1 and Figure 5.2 where we can observe the 4 properties that constitute the random forest: “classes”, “feature\_names”, “feature\_importances”, and “estimators”.

- **Classes:** represents the channels that can be used to contact the client. In our case, we have 3 channels (email, sms, voice) that correspond to 3 classes
- **Feature\_names:** the names of each feature mentioned in the data are defined
- **Feature\_importances:** represents the feature importance values of each feature for the random forest model
- **Estimators:** represents the decision trees generated by the RF model.

```

"estimators": [
  {
    "feature_importances": [
    ],
    "impurity": 8.25936855734962,
    "nodes": [
      {
        "node_id": 0,
        "depth": 0,
        "parent_node_id": null,
        "decision": "<=",
        "split_feature": 5,
        "split_threshold": 121.13974760153319,
        "child_nodes": {
          "left": 1,
          "right": 20
        },
        "values": [
          [
            402.0,
            407.0,
            385.0
          ]
        ],
        "impurity": 0.6664800832751137
      }
    ]
  }
]

```

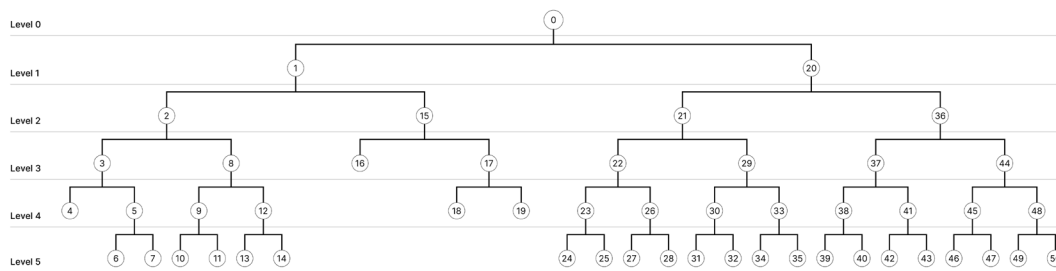
**Figure 5.2** - Original data structure: “estimator” properties

Overall, this JSON structure sets the groundwork for analyzing and visualizing data. It defines the classes, lists the features being considered, and stores the importance values of these features.

Each estimator contains 3 properties, “feature\_importances”, “impurity”, and “nodes”. The first property refers to the importance of each feature in the respective decision tree, the second reveals the impurity of the decision tree and the last consists of the definition of the nodes that make up the decision tree.

The property called “nodes” defines an object with several properties. Upon analyzing them, we considered that the most essential for the development of the visualization are “depth”, “split\_feature”, “child\_nodes”, “values”, and “impurity”. Through these properties, we can obtain information about each node such as, the level it is at, the number of samples by class, its impurity value, the feature responsible for further dividing the samples, its children and which way they branch.

Knowing the node level is crucial, given that the level at which a feature is used is a factor that contributes to the importance of that feature in the decision tree. The branching in a decision tree is essential for understanding the decision logic, interpreting the model’s behavior, making predictions, debugging issues, and integrating the tree into more complex machine-learning workflows (Yadav, 2019; Thorn, 2021). The impurity in a decision tree node is essential to guide the tree-building process by identifying where to split the data for better classification or prediction (Ansari, 2022).



**Figure 5.3 - Outline of the first decision tree of the data set**

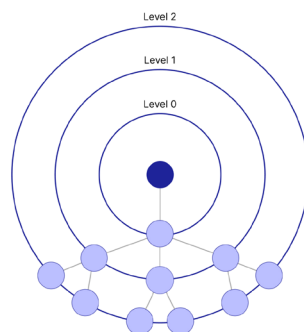
Once we analyzed the JSON file, we have come to ascertain that the aforementioned random forest consists of 500 decision trees. Each of these exhibits a maximum depth of 6 levels, ranging from 0 to 5. Additionally, throughout the random forest, there are 11 distinct split features, numbered from 0 to 10.

Furthermore, we were able to outline a decision tree as shown in Figure 5.3, referring to the first tree of the data set from the JSON file. By looking at the sketch of the decision tree, we conclude that the tree is built from left to right through the enumeration within each node.

After thoroughly examining the dataset, we opted for two different visualization techniques that we considered suitable for the type of data—one for the random forest and another for the decision tree—which will be explored in the next section. Since they are distinct approaches, it is required to structure the data differently for each to ensure that the chosen visualizations align well with the specificities of each algorithm.

### Random Forest

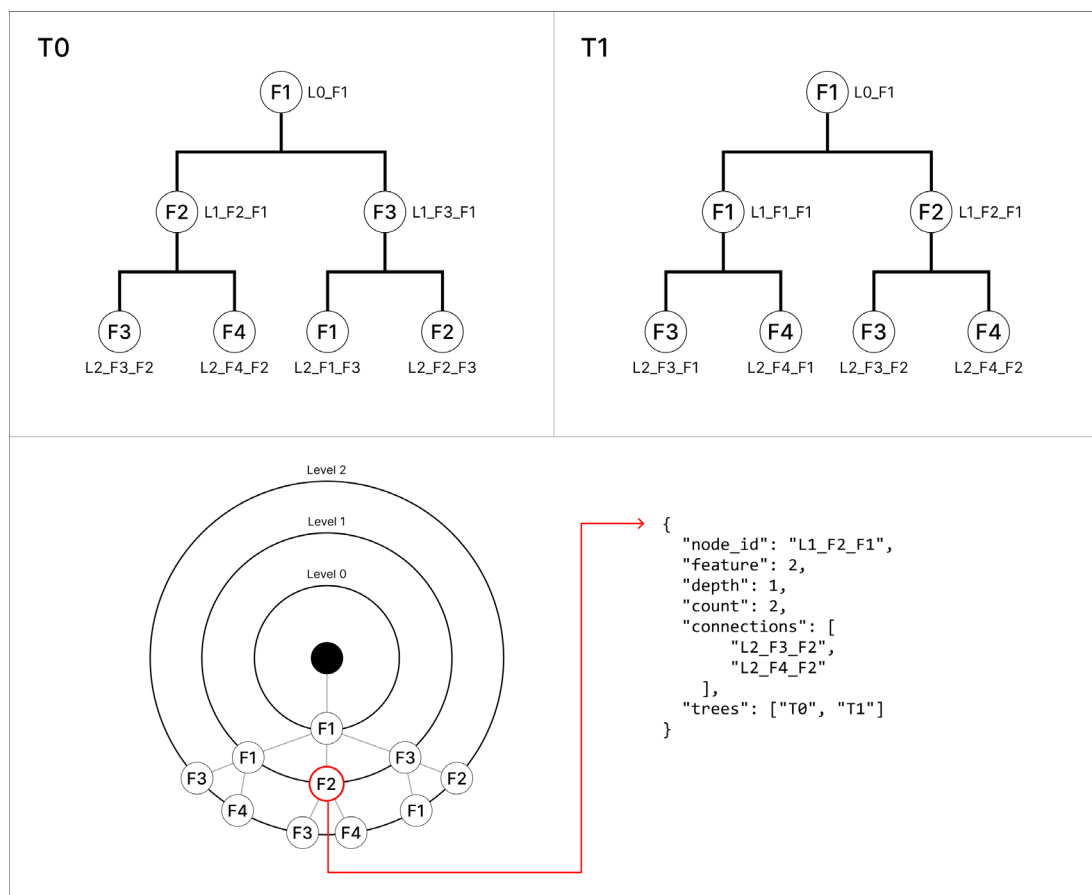
To represent the random forest in a single tree visualization, specifically a radial tree, we needed to organize the decision trees, that compose the ensemble, by feature and group them by the most important features. Each node in the radial tree, represented by a circle, represents a set of trees that have the feature indicated at the level at which that node is located. The dark blue circumferences of the radial tree (Figure 5.4) establish the levels that the decision trees have. With this visualization model we aim to show what features are represented in each level and the relationship between features, that is what features each one is linked to in the other levels.



**Figure 5.4 - Structure of a radial tree**

To build the radial tree we needed the following information for each node: the feature name, the level (i.e., depth) it is on, the connections, the number of times that feature is repeated, and in which trees this feature in this level occurs in. To do this, we assigned an id to each node and each decision tree. The id of the node is composed by the level, the feature, and the feature of its parent, as explained further.

Figure 5.5 shows an example on how to build a radial tree to represent a random forest composed of two Decision Trees (T0 and T1). . The node highlighted in red is identified by the ID “L1\_F2\_F1,” which translates to being at level 1, containing feature number 2, and its parent node containing feature number 1. Apart from the feature and level, represented as “feature” and “depth,” respectively, each node contains the “count” property, which defines the number of times the parameters of the node are repeated in the random forest. Additionally, there is the “trees” property that indicates the number of trees in the random forest where it repeats. In this case, it repeats twice in trees T0 and T1. Lastly, the “connections” property, as the name suggests, defines the connected nodes in the subsequent level.



**Figure 5.5** - Example of a random forest representation (below, left) composed by two decision trees (above) and the data structure of a node (below, right)

```

{
  "classes": [...],
  "feature_names": [...],
  "feature_importances": [...],
  "estimators": [
    {
      "feature_importances": [...],
      "impurity": 8.25936855734962,
      "nodes": [
        {
          "node_id": 0,
          "depth": 0,
          "parent_node_id": null,
          "decision": "<=",
          "split_feature": 5,
          "split_threshold": 121.13974760153319,
          "child_nodes": {
            "left": 1,
            "right": 20
          },
          "values": [
            [
              402.0,
              407.0,
              385.0
            ]
          ],
          "impurity": 0.6664800832751137
        }
      ]
    }
  ]
}

```

```

{
  "classes": [...],
  "feature_names": [...],
  "feature_importances": [...],
  "nodes": [
    {
      "node_id": "root",
      "connections": [
        "L0_F5",
        "L0_F6",
        "L0_F1",
        "L0_F0",
        "L0_F7",
        "L0_F4",
        "L0_F10",
        "L0_F9",
        "L0_F3",
        "L0_F8"
      ],
      "trees": [],
      "count": 0
    },
    {
      "node_id": "L0_F5",
      "feature": 5,
      "depth": 0,
      "count": 6,
      "connections": [
        "L1_F1_F5",
        "L1_F10_F5",
        "L1_F3_F5",
        "L1_F6_F5",
        "L1_F9_F5",
        "L1_F5_F5",
        "L1_F4_F5"
      ],
      "trees": ["T0", "T12", "T21", "T25", "T37", "T39"]
    },
    ...
  ]
}

```

**Figure 5.6** - Structuring of the data to visualize the Random Forest

Comparing the structure of the initial file, on the left side of Figure 5.6, with the structure of the radial tree, on the right side of Figure 5.6, the difference lies in the way the decision trees are arranged. While in the former the data is structured in such a way as to define several trees, in the latter the architecture refers to a single tree, containing, however, all essential information.

### Decision Tree

To build the visualization of the decision tree, the structuring of the data was simpler than in the case of the random forest. The visualization technique chosen, sunburst, contains a similar structure, but some minor changes have been made to make it easier to implement.

As you can see in Figure 5.7, the information we do not consider essential for visualization is not used. On the other hand, we have added the “side” property which defines which side of the branch the node belongs to, and the “size” property which defines the size of the leaf node to maintain the symmetry of the sunburst.

The biggest change to the structure of the data is related to the order in which the nodes are established in the construction of the tree. As previously mentioned, in the original data, the tree is built starting with the nodes on the left and working downwards. However, the sunburst implementation requires the tree to be built from the right-hand side.

```

"nodes": [
  {
    "node_id": 0,
    "depth": 0,
    "parent_node_id": null,
    "decision": "<=",
    "split_feature": 5,
    "split_threshold": 121.13974760153319,
    "child_nodes": {
      "left": 1,
      "right": 20
    },
    "values": [
      [
        402.0,
        407.0,
        385.0
      ]
    ],
    "impurity": 0.6664800832751137
  }
]

"depth": 0,
"feature": "5",
"impurity": 0.6664800832751137,
"samples": [[402.0, 407.0, 385.0]],
"children": [
  {
    "depth": 1,
    "side": "right",
    "feature": "10",
    "impurity": 0.6659739896606836,
    "samples": [[388.0, 384.0, 350.0]],
    "children": [...
      {
        "depth": 5,
        "impurity": 0.5950413223140496,
        "samples": [[2.0, 6.0, 3.0]],
        "size": 1
      }
    ]
  },
  {...}
]
}

```

Figure 5.7 - Structuring of the data to visualize the Decision Tree

In this section, we showcased the work carried out in data analysis to aid in the selection of visualizations and the structuring of data that assists in their implementation. In the upcoming chapters, we delve into the process of choosing the visualizations.

## 5.4. Visualization models

As stated before, analyzing data before selecting a visualization technique is of paramount importance as it lays the foundation for effective communication and insightful interpretation (Layer Software GmbH, 2023). It aids in identifying the most suitable visualization method that can effectively convey the intended message and highlight key insights. A well-chosen visualization not only enhances comprehension but also brings attention to relevant details that might be overlooked in raw data (Gill, 2023).

With due consideration of the points aforementioned, after a thorough analysis of the data, certain relevant factors become prominent. This involves presenting a large set of data effectively ensuring its accessibility to the audience. Simplicity and lucidity in interpretation are crucial to facilitate comprehension for a wide array of stakeholders. Moreover, wise utilization of space is essential, which entails exercising caution to prevent an overwhelming visualization.

Considering the noteworthy attributes showcased by the trees—encompassing six levels of depth inclusive of the root node, a predominantly broader structure rather than a deeper one, and an ensemble of eleven features within the random forest—our discerning assessment of visualization techniques discussed in Chapter 4 guided our decisions. As a result, we selected the radial tree visualization approach to represent the random forest, while opting for the sunburst visualization technique for the decision tree.



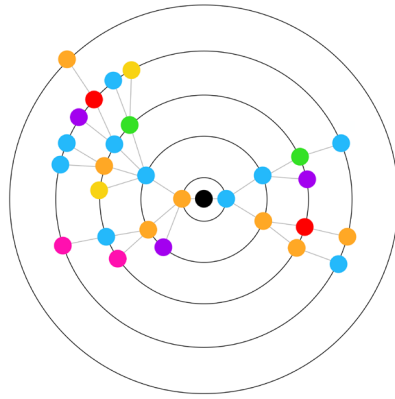


Figure 5.8 - Radial tree layout example

We opted for the radial tree (Figure 5.8) due to its efficient use of space and the evident perception of hierarchy it provides, offering a sense of balance and symmetry. Regarding the sunburst, which is a kind of space-filling version of the radial tree, it holds similar advantages to the radial tree as it also utilizes space effectively. In certain contexts, understanding it might be challenging, yet in this particular scenario, the decision trees are binary and not very deep, which contributes to the symmetry of the tree and consequently aids in its interpretation.

To maintain consistency between the two visualizations, we choose two models that use circles in its structure. Incorporating circles in the design, frequently involves designers tapping into the viewer's psychological responses. Circles symbolize concepts of unity, harmony, and inclusiveness. They represent a continuous form, a quality that renders circles a powerful yet safe, comforting, and serene design element. This necessity of maintaining consistency, was one of the reasons why we chose the sunburst visualization rather than other techniques we studied, as shown in Figure 5.9.

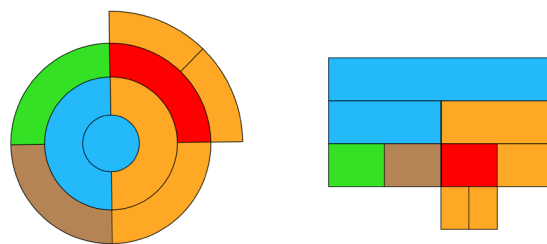


Figure 5.9 - Study sunburst vs icicle plot

“Circles are among the most ubiquitous symbols around the globe, used in countless variations since the birth of humankind. Associated with notions of unity, wholeness, and infinity, the circle has been an important visual metaphor in a wide array of systems of thought, from cartography and astronomy to physics and geometry. It was therefore inevitable that it eventually would be used to represent hierarchical structures.”

(Lima, 2014)

## 5.4.1. Studies for the visual structure

Following the selection of visualization techniques, a series of studies were conducted to define the visual marks and visual properties of the visualization. In this subsection, we present the chronicle of a set of studies conducted to test the visualizations and determine the most suitable visual properties. Furthermore, we engage in exploring alternative techniques beyond tree-based methods and revisit the work undertaken previously, detailing the subsequent modifications made.

To gain a more comprehensive understanding of the operational aspects of the sunburst and radial tree visualizations, we conducted brief studies employing a subset of simple decision trees extracted from the article on random forests ([https://uc-r.github.io/random\\_forests](https://uc-r.github.io/random_forests)), as illustrated in Figure 5.10.

The trees presented in Figure 5.10, although not identical, bear resemblance to the trees we will work with, in the sense that they are wider than they are deep and encompass diverse features. This facilitated the testing of different components. Consequently, after finalizing all visualization details with this sample of trees, the adaptation to the provided data became straightforward.

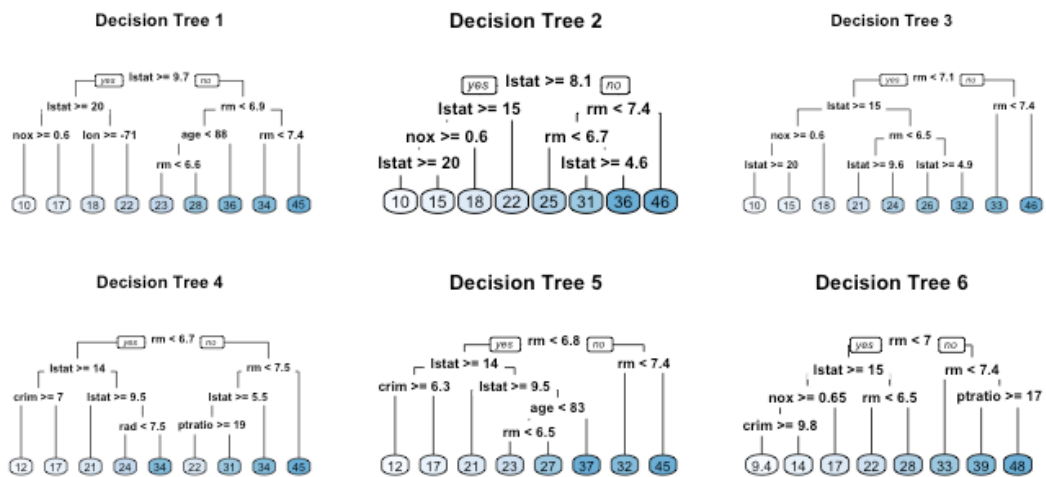


Figure 5.10 - Six decision trees based on different bootstrap samples

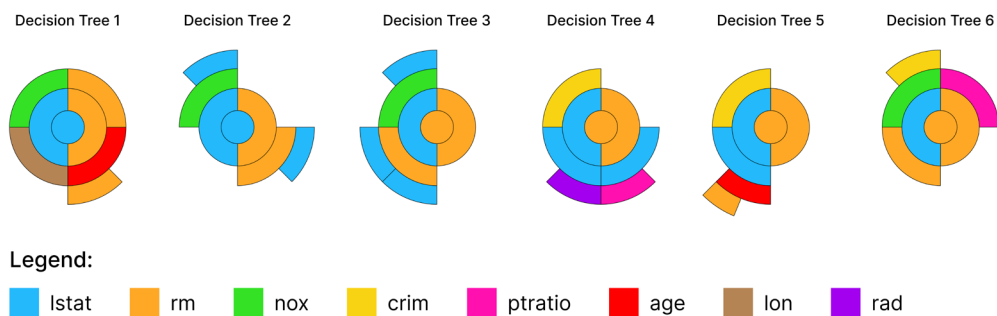
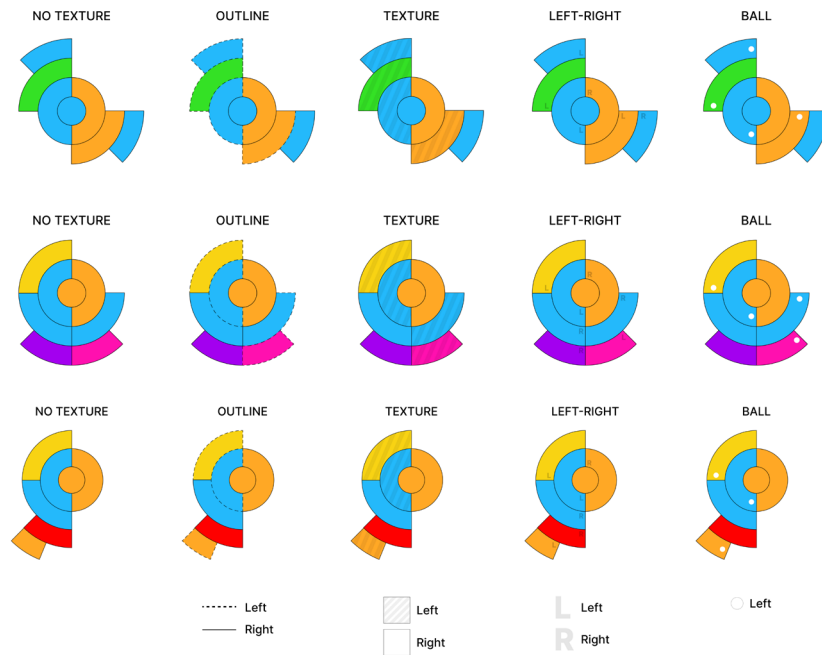


Figure 5.11 - Six decision trees transformed into sunbursts



**Figure 5.12** - Studies to distinguish the direction of tree branching in different tree structures

We started by assigning a distinct color to each of the eight features present in the random forest. Subsequently, we transitioned to transforming the decision trees into sunbursts, as depicted in Figure 5.11. This process aided not only in grasping the functioning of the visualization but also in identifying areas that required improvement.

### Sunburst – Branching of the decision trees

While analyzing the models, the need arose to enhance the differentiation between nodes branching to the left and those branching to the right. To address this, various types of visual marks and properties such as, outlining, texture, text, and shapes were tested, as outlined in Figure 5.12. To determine the most effective and visually appealing manner of distinguishing the direction of the tree branching, a shared questionnaire (Appendix A) was conducted among participants.

The questionnaire in question was completed by a total of twenty-seven participants from the fields of Design and Multimedia, Computer Engineering, Data Science, and Biomedical Engineering. Within this group, twenty-one are students, four are researchers, one is a professor, and one is a data scientist.

The questionnaire comprises the presentation of the trees from Figure 5.12, followed by the question: “In your opinion, which of the approaches below is most suitable for distinguishing the left side from the right side of the various branches of the tree?” This question was posed for each of the trees, yielding the results shown in Figure 5.13.

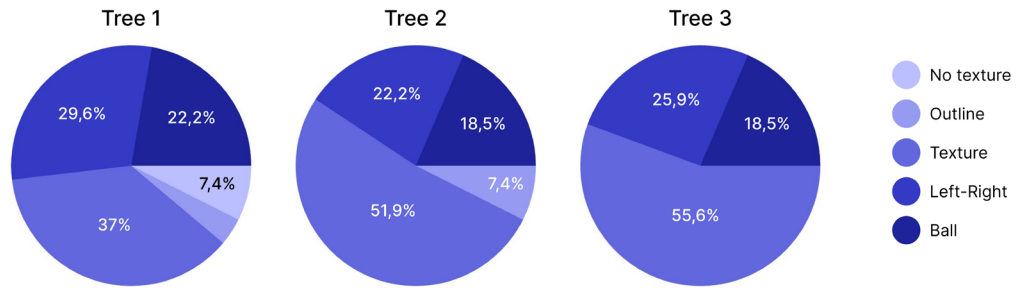


Figure 5.13 - Results of the “Decision Tree Visualization” questionnaire

As evident from Figure 5.13, it can be inferred that the method users found most effective is the application of texture to the node, as they argued it to be the most intuitive and consistent choice. Despite the “Ball” option not being the favorite, it was one of the most voted solutions, thus it was later employed for a different function in the visualization.

### Classes and impurity

Once the use of texture was established, the matter of values to be presented in the leaf nodes was addressed, specifically, the best way to represent them in the sunburst. While the example used in Figure 5.9 did not involve classes, we defined that the different values in the leaf nodes represented classes—to align more closely with the decision trees in our work. The assignment was structured as follows:

- Class 1: values from 0 to 10
- Class 2: values from 11 to 20
- Class 3: values from 21 to 30
- Class 4: values from 31 to 40
- Class 5: values from 41 to 50

Based on this distribution, we conducted several tests in the visualization, exploring the use of text (A), size variation (B), colors (C), or width (D) of the nodes, as illustrated in Figure 5.14.

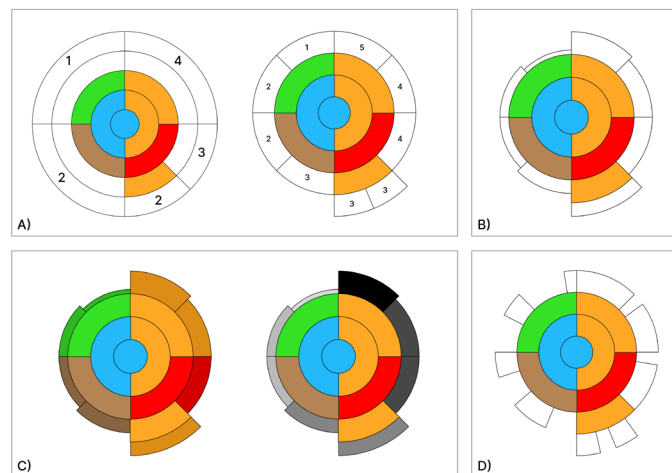
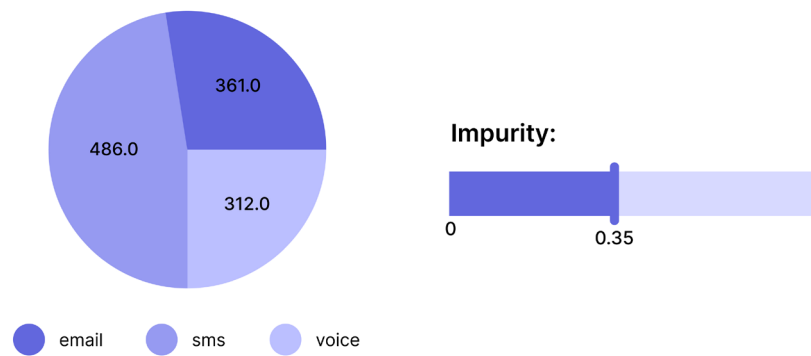
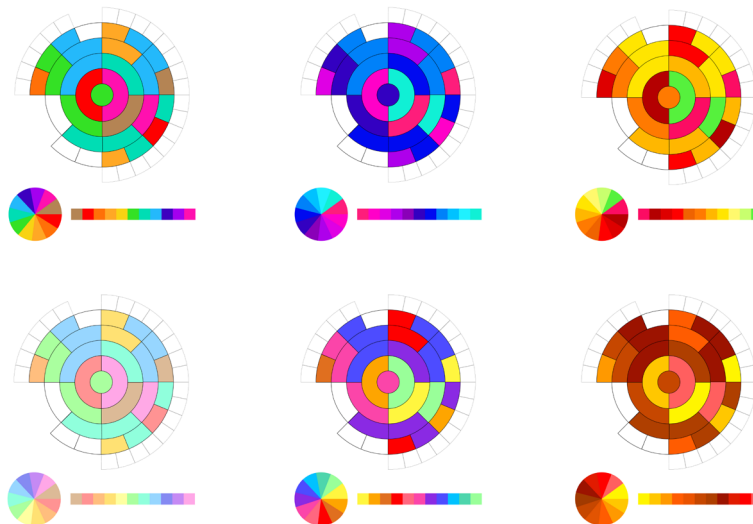


Figure 5.14 - Leaf-nodes visual tests



**Figure 5.15** - Study of the pie chart visualization technique for class presentation (left) and the visualization technique for presenting the “impurity” property (right)



**Figure 5.16** - Study of colours to be assigned to each feature

In the case of the data we are working with, we intend to present not just one class but all three classes - email, sms, and voice - within each node. Given this particular detail, the options depicted in Figure 5.14 render the visualization illegible and challenging to interpret. Consequently, we chose to implement a panel that would encompass all the information for each node, including the classes. This approach allows us to comprehend the distribution of samples along the nodes of the decision tree from the root down to the leaf nodes. To display the values of the three classes, we determined that a pie chart would be employed, exemplified in Figure 5.15 (left).

In addition to the classes, presenting the “impurity” values for each node is essential. With this in mind, the visualization chosen for this purpose is depicted in Figure 5.15 (right). This technique involves a single-bar chart ranging from zero (0) to one (1), which fills according to the impurity value of the selected node.



Figure 5.17 - Colour Palette

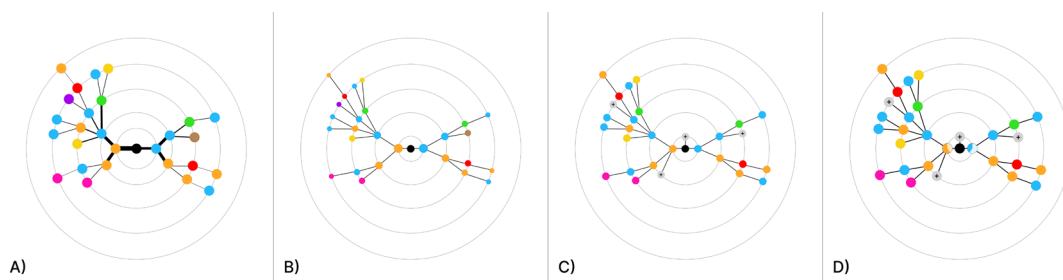


Figure 5.18 - Tests for the random forest visualization

### Feature Colors

Upon finalizing the visualizations, a brief study of color coding was conducted to associate each feature with a distinct color. This task posed a challenge given the presence of eleven features, meaning 11 different colors. It was imperative to consider the contrast between them and whether they harmoniously worked together to maintain visual appeal and user interpretability. To address this, we opted to employ a sketch of one of the decision trees created from the provided data. Figure 5.16 illustrates the conducted studies, while Figure 5.17 showcases the definitive colors assigned to each feature.

### Radial Tree

The studies conducted for the random forest aim to comprehend the optimal arrangement of nodes in the radial tree. Given that the objective is to illustrate the most significant features at each level, finding a suitable visual solution is crucial. To achieve this, we developed the following alternatives, as presented in Figure 5.18.

In option A, we maintain a consistent circle size for all nodes and adjust the thickness of branches based on how strong the connections between features are. In option B, we alter the node size according to the level it resides in. In alternative C, the approach involves maintaining both node size and branch thickness equal, but hiding less significant nodes in gray circles. . Finally, for hypothesis D, it is similar to the previous one, except nodes are replaced with pie charts to provide an indication of how many trees the feature is present in.

However, the legibility of nodes is at stake, especially since the random forest we are working with contains five hundred (500) trees instead of just six (6). As a result, options A and D were discarded; in the former, the thickness of branches with less important connections can become imperceptible, and in the latter, observing the pie charts as they move away from the root node becomes difficult. Due to the potential confusion of nodes used to add more nodes in solution C when utilizing the five hundred trees, option B emerges as the most viable. Despite the selection, it is noteworthy that with a random forest containing such a high number of decision trees, a functionality must be implemented to filter the most important features for visualization and a zoom technique to emphasize nodes of greater interest.

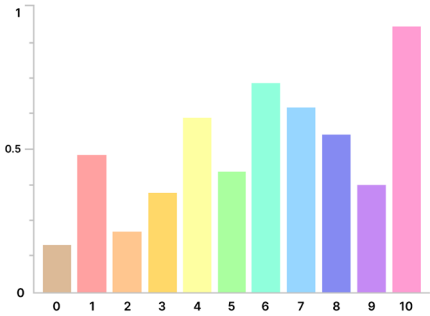


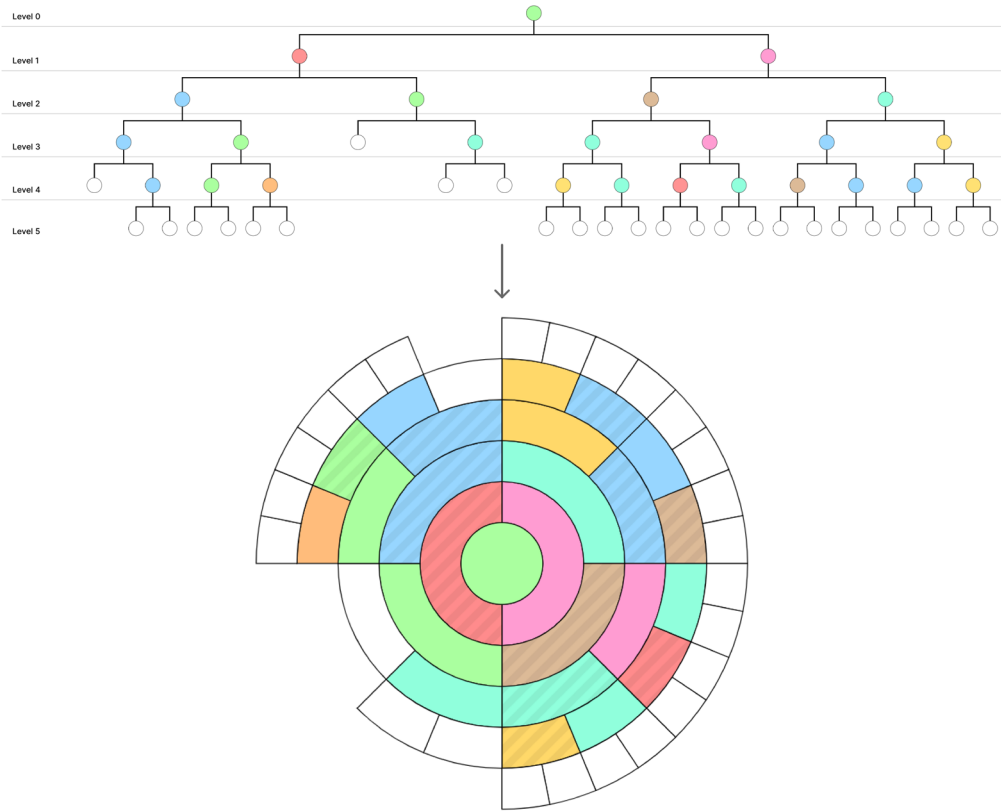
Figure 5.19 - Feature importance graph

**Feature importances**

Feature importances are values presented not only in the random forest but also in each decision tree. Thus, we opted for a visualization technique employed in both tree-based visualizations. This technique involves a bar chart that utilizes the colors of features, each represented by a corresponding bar, as illustrated in Figure 5.19. In this manner, it becomes possible to define the most and least important features while simultaneously comparing them.

**5.4.2. Final visualization models**

In this section, we introduce the finalized visualization models using the provided data and how they are transformed into the corresponding visualizations for the random forest and the decision tree. Once the studies for the visualization models were completed using a dataset composed of only six trees, we proceeded with the transformation of the provided data for the intended visualizations.



**Figure 5.20** - Conversion of decision tree data to sunburst



**Figure 5.21** - Visualization of different decision trees

Initially, we performed the conversion without code implementation to observe how the visual properties behaved, enabling us to confidently proceed to model development. We commenced by converting a single decision tree into a sunburst (Figure 5.20). The outcome showcases a contrast between colors, distinguishing leaf nodes from the other nodes. The texture differentiation was well achieved, and overall, we find the visualization appealing. Post-implementation, we observed how the visualization model behaved with distinct decision trees (Figure 5.21).

Secondly, we conducted a simulation of how the random forest appears with two trees extracted from the data we were working with. The radial tree, represented in Figure 5.22, consists of five black circumferences representing the levels of the decision trees, from the root down to the level preceding the leaf nodes. The radial tree's root corresponds to the entire dataset, and the nodes represent a set of decision trees.



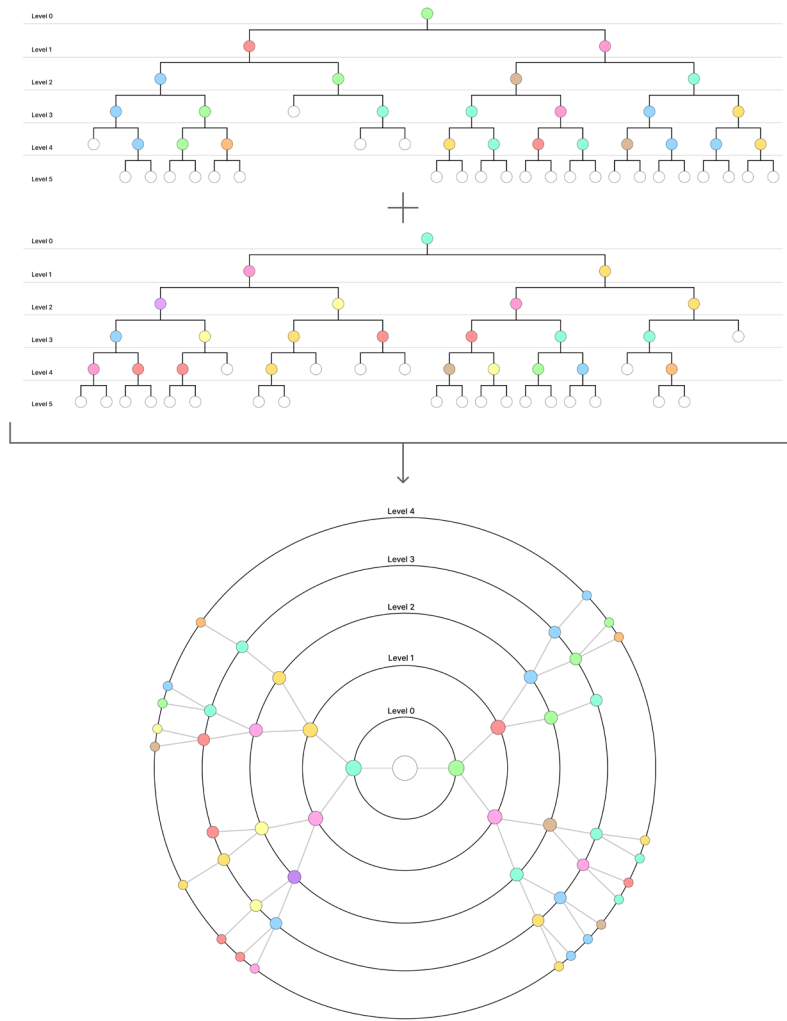


Figure 5.22 - Conversion of two decision trees into a radial tree

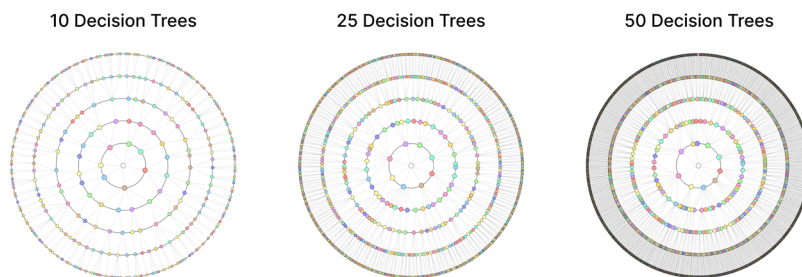


Figure 5.23 - Radial trees with different number of trees

Once implemented, it is possible to observe the behavior of the radial tree with varying numbers of trees (Figure 5.23). As the number of trees increases, legibility decreases, justifying the use of techniques such as zoom and filtering to counteract this issue, which are explained in the subsequent chapter.

## 5.5. Web application development

Despite the primary focus of the project being the visualization models, the solution also involves the development of the platform where these are embedded. The development of the web application, which incorporates all the elements mentioned in Chapter 5.4—colors, visualizations, charts—enables testing the interactions between the visualizations and the user, as well as understanding whether the data is being accurately conveyed.

The next sections focus on the steps taken for the development of the web application, from its prototype to its implementation. In the section regarding the prototype, we present the user flow followed by the medium and high-fidelity prototypes of the pages to be implemented. Next, we briefly introduce the languages used to programme the web application and provide the structure of the code developed during the implementation.

### 5.5.1. Prototype design

In this section, we address the process of the interface design, from previous concepts to the final prototype. Firstly, we start by describing the application flow so we can understand the structure of the application. Secondly, we present the first tests performed for the design of the application through medium-fidelity, followed by the high-fidelity prototypes which were then implemented. The prototypes displayed in this chapter were created in Figma.

A medium-fidelity prototype, serves as an intermediate step in the design process, helping to bridge the gap between the initial concept exploration and the more advanced stages of design and development (high-fidelity). These prototypes offer individuals a clearer impression of the potential appearance of the solution and provide flexibility for altering the course and experimenting with various choices during the design process (Dam et al, 2023). For this part, we used wireframes that serve as a visual guide, outlining the fundamental structure of the digital product. They offer insights into their interconnections and overall arrangement by illustrating the relationships between various elements (What is a Wireframe & Its Role in the Design Process, (n.d.)).

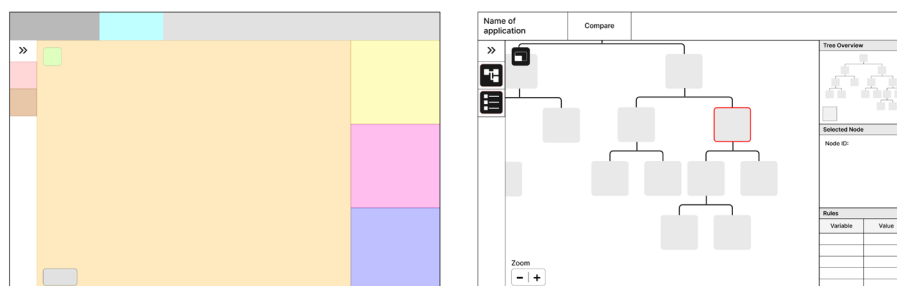


Figure 5.24 - Wireframes for the intermediate delivery

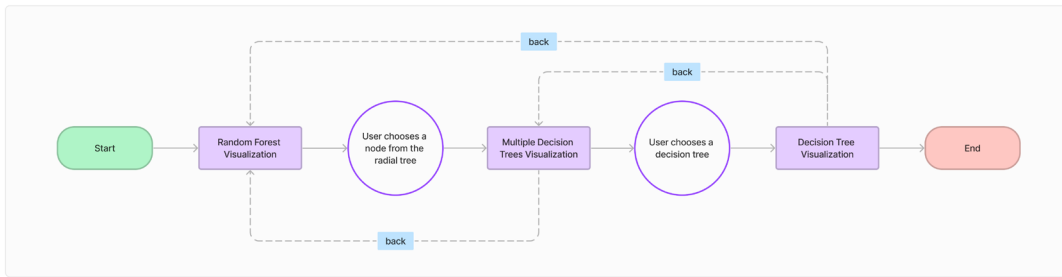


Figure 5.25 - Application user flow



Figure 5.26 - Updated Wireframes

For the intermediate delivery, we created wireframes with the aim of meeting the previously established requirements (Figure 5.24). However, due to changes in the tasks and requirements, new wireframes were developed that align with the intended goals for the development of the platform.

The first wireframes were considered inaccurate and did not fulfill one of the main objectives of visualizing the random forest. That said, after a review of the requirements, we developed a brief application flow (Figure 5.25) which explains the succinct structure of the application and assists in the development of new wireframes (Figure 5.26).

The flow of the application is quite simple, which makes it easy for the user to navigate. The application consists of three pages, and the user accesses the next page by interacting with the visualization models. As you can see from the wireframes in Figure 5.26, the first page is for visualizing the random forest, the second contains the representation of multiple decision trees and the third page is for visualizing a single decision tree. To see the wireframes on a larger scale, see Appendix B.

After stipulating the components that make up each page, we proceeded to make a high-fidelity prototype using mockups (Appendix C), which will be presented further. High-fidelity prototypes are developed to thoroughly assess the entire range of interactions within the finished solution, examining it from functional, visual, and experiential perspectives. This approach offers a considerably more authentic representation of the potential final product, enabling you to make precise refinements and conduct comprehensive experience evaluations in the later stages (Dam et al, 2023). In this stage, we developed mockups, a graphical depiction that simulates the ultimate visual presentation of a design, portraying genuine design components like colors, typography, and images to provide a tangible sense of the final aesthetic (What is a Mockup and its Role in Design?, (n.d.)).

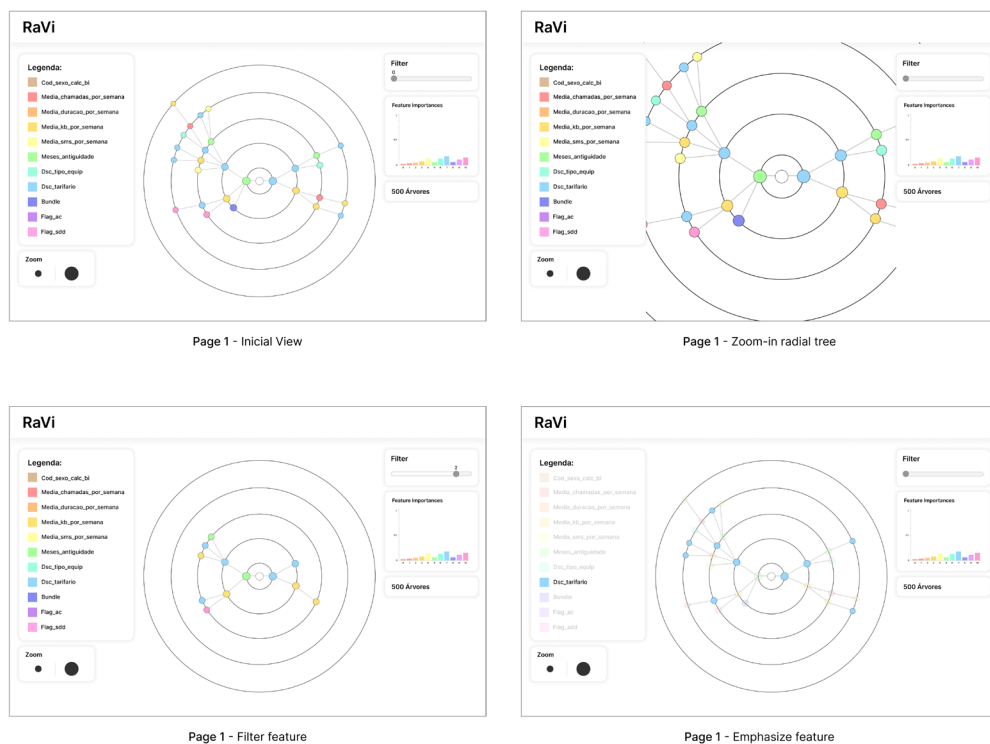


Figure 5.27 - “Radial Tree Visualization” screens

### Radial Tree Visualization

The page depicted in Figure 5.27 is composed of seven components: navigation bar, legend, zoom, filter, feature importances chart, and tree count panel. The navigation bar is consistent across all pages and features a breadcrumb trail menu to enable easy navigation backward while also orienting the user within the application. The legend serves not only as a translator for the color codes of the features but also as a type of filter. When a user clicks on a feature in the legend, the nodes in the radial tree with the same color are highlighted, as demonstrated in the bottom right screen of the figure. The third component enables users to zoom in and out on the radial tree to gain a better view of the model. The main component of the page is the visualization of the random forest, represented by an interactive radial tree. This page also includes a filter which, as the name suggests, filters the data displayed in the

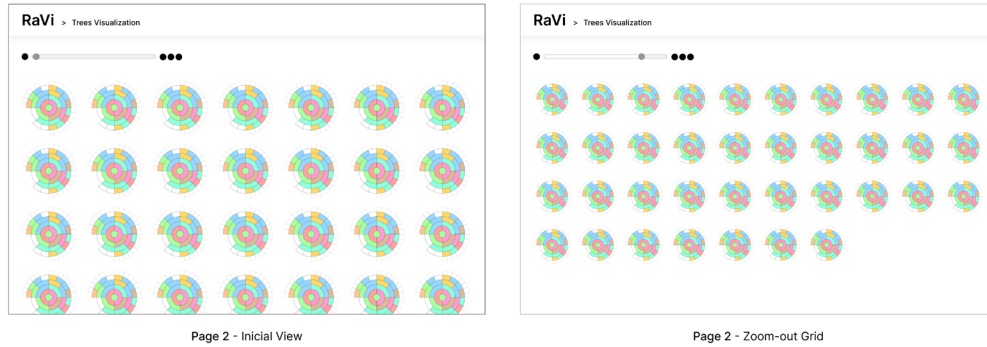


Figure 5.28 - “Multiple Trees Visualization” screens



Figure 5.29 - “Sunburst Visualization” screens

tree according to the “count” property mentioned in Chapter 5.3. In the example shown in Figure 5.27, the filter is set to “2”, meaning that only nodes with a count equal to or greater than two are displayed in the radial tree. Additionally, the page contains a panel that depicts the feature importances relative to the random forest and another panel displaying the value indicating the quantity of trees comprising the random forest.

### Multiple Trees Visualization

When the user clicks on a node in the radial tree on the first page, they are directed to the page presented in the screens of Figure 5.28. This page presents the visualization of various decision trees and a slide bar that controls the dynamic grid containing the trees. The set of decision trees displayed in the grid correspond to the trees that

adhere to the characteristics of the node the user clicked on. For instance, if the user clicks on the node with feature 5 at level 0 of the radial tree, the subsequent decision trees shown are all the trees from the random forest that include feature 5 at level 0.

### Sunburst Visualization

From the page with multiple trees, the user is directed to a page featuring the visualization of just one decision tree, as shown in Figure 5.29. This transition occurs when the user clicks on one of the trees in the grid, displaying the selected tree. The third page contains two legends, the decision tree visualization model, a chart panel, and a panel with node information. On this page, in addition to the legend related to features and their assigned colors, an additional legend has been introduced to define the usage of texture in the visualization. Once again, the main component of the page is the visualization itself, represented by a sunburst. Each time a user clicks on a node in the sunburst, information about that node appears in a panel on the right, accompanied by a circle that overlays the node for identification. The panel for the node contains information such as its level, the name of the feature used, a pie chart depicting class distribution, and a single bar chart indicating the “impurity” value. Regarding the pie chart, if the user wishes to observe the values corresponding to each class, they can interact with the chart by hovering over each slice. Similar to the first page, this page also includes a bar chart displaying the feature importances, albeit in this case, it refers to the feature importances of the represented decision tree. From this page, the user can navigate back to the previous page or directly return to the random forest.

In summary, in this subsection we presented the medium and high-fidelity prototypes developed for the web application. The medium-fidelity prototype suffered considerable changes since the intermediate delivery. After finishing the design of the high-fidelity prototypes we implemented the aforementioned, as we will discuss in the next section.

## 5.5.2. Application implementation

The implementation of the web application RaVi was mostly developed using D3.js (Data-Driven Documents), a free and open-source JavaScript library for data visualization. Its low-level, standards-based approach offers unparalleled flexibility in creating dynamic and data-driven graphics (What is D3?, (n.d.)). In addition to the D3.js library, HTML, CSS, and JavaScript languages were also used for creating the rest of the application along with the visualization templates. HTML is the foundation of web development and the language used to structure content. It uses various elements, represented by tags, to define the structure of a web page. CSS is used to control the presentation and visual styling of web pages. It is a stylesheet language that defines how HTML elements should appear on the screen or other media. JavaScript is a dynamic programming language that adds interactivity and behavior to web pages. In summary, D3 is responsible for the data-driven visualizations, HTML structures the content, CSS styles the presentation, and JavaScript adds interactivity and dynamic behavior to web pages. These four technologies work together to create the present web application. The D3.js framework was chosen in advance as a

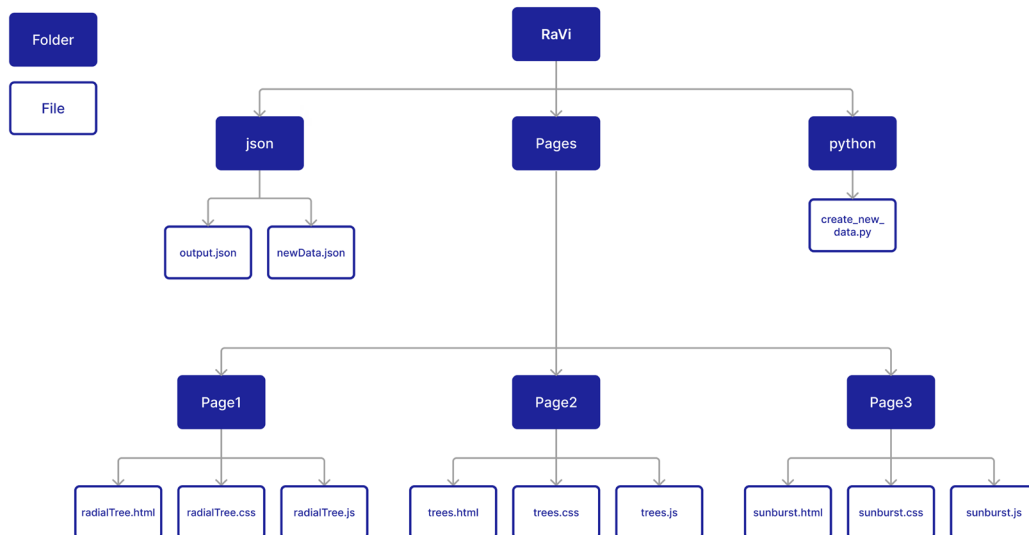


Figure 5.30 - Code structure

requirement for the development of the project in question. All the code developed during the implementation of the application is in a private GitHub repository which can be accessed through the following link <https://github.com/EdyS12/RaVi.git>.

The developed code is divided into three distinct folders, namely “json,” “Pages,” and “python,” as depicted in Figure 5.30. Within the “json” folder, there are two JSON-format files, which correspond to the data used for creating the visualizations. The “output.json” file contains the original data without any structural modifications, while the “newData.json” file contains data structured specifically for the development of the random forest visualization—radial tree.

The main folder referred to as “Pages” contains the code developed for the web application’s pages and is, in turn, composed of three subfolders, each corresponding to a different page: (i) the “Page 1” folder contains the code for the random forest visualization; (ii) the “Page 2” folder contains the page featuring the visualization of multiple decision trees; and (iii) the “Page 3” folder is related to the application page where the user accesses the visualization of a decision tree—sunburst. The subfolders for each page all follow the same structure, meaning each subfolder contains three files—an HTML, a CSS, and a JavaScript file. The HTML files are used to define the structure of each page, organize content into various elements and establish the logical hierarchy of the page’s content, they include links to the CSS and JavaScript files. To style the HTML elements, the CSS files are used to define the visual presentation and layout of the components. Finally, the JavaScript files are responsible not only for the interactivity and functionalities implemented in the application but also contain the code developed in D3.js for creating the visualizations on each page.

The last folder called “python” contains a single Python file. This file contains the code that converts the structure of the initial “output.json” data into the necessary structure to draw the radial tree. This file also allows changing the number of trees from the random forest that we want to be revealed in the visualization.

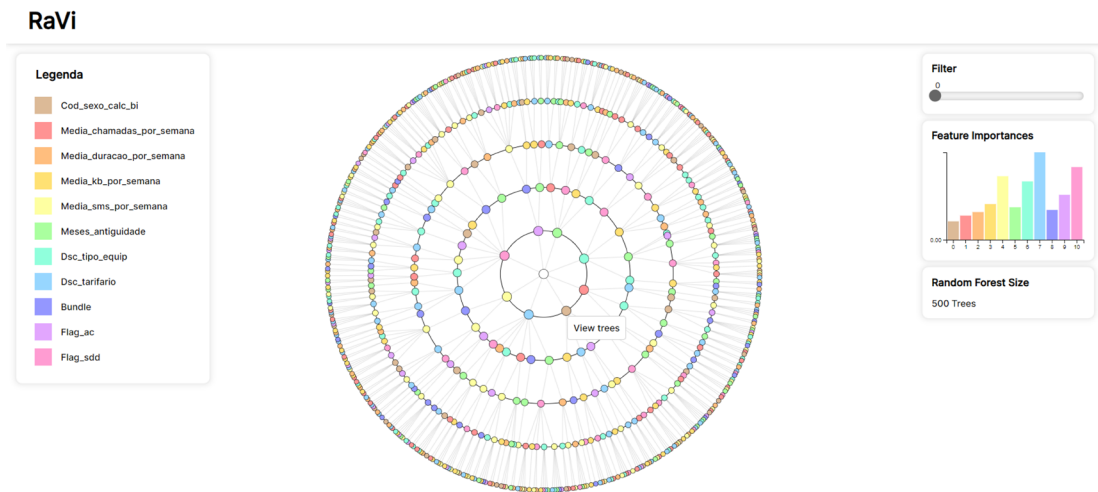


Figure 5.31 - “Radial Tree Visualization” implemented Screen

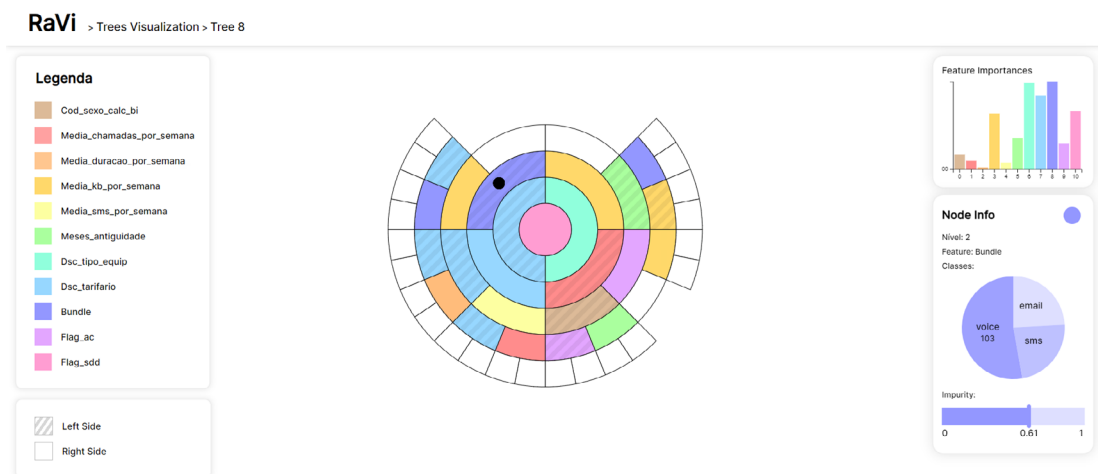


Figure 5.32 - “Sunburst Visualization” implemented Screen

During the implementation of the radial tree, we searched for similar solutions from other public works. For the radial tree, we based our work on an open-source code (PenBox Level 2 (forked) - CodeSandbox, 2023), which already included a zoom-in and zoom-out function controlled by mouse controls, rather than a button as initially envisioned in the high-fidelity prototype. Consequently, we discarded the button implementation as can be seen in Figure 5.31.

For the sunburst (Figure 5.32), we followed the structure of a tutorial from GitHub (Sunburst Tutorial, n.d.). However, we made some changes depending on the data we were working with, including adding the root node and the interaction that involves displaying the information for each node and identifying it. After the implementation of all functionalities, we move on to the user testing phase, which will be discussed in Chapter 5.7.



## 5.6. Use cases

A use case is a term employed in software development, product design, and other related fields to depict how a system can be utilized to accomplish distinct objectives or assignments. It delineates the interplay between users or entities and the system to attain a precise result (Daly, 2023). In this section, we present three use cases in which we describe the interaction with the application from the random forest visualization to the decision tree visualization and what information is extracted from the models and the interaction. To see the images of the screens implemented on a larger scale, see Appendix D.

### Case 1: Feature relationships and feature most used feature in a given level

The user, a data scientist that we assume to be familiar with random forests, decision trees, and their components, utilizes the RaVi platform to visualize data. The goal is to obtain detailed information about the features of the random forest. This information is important since it provides a better understanding regarding whose attributes better characterize the different types of users in the data set, according to the classes.

After entering the RaVi platform, the user comes across a radial tree visualization representing the random forest (Figure 5.33, above, left). The color of the node is associated with the color of its feature, and the color respective to each feature can be checked in a legend on the left-hand side of the screen. To have a better look at a specific portion of the structure, the user zooms in on the radial tree, enabling a comprehensive view of the relation between features. From then on, the user can see that the feature “Dsc\_tipo\_equip” (feature 6) at level 0 is connected to seven features—features 0, 3, 5, 6, 7, 9 and 10—at level 1 throughout the random forest (Figure 5.33, above, right).

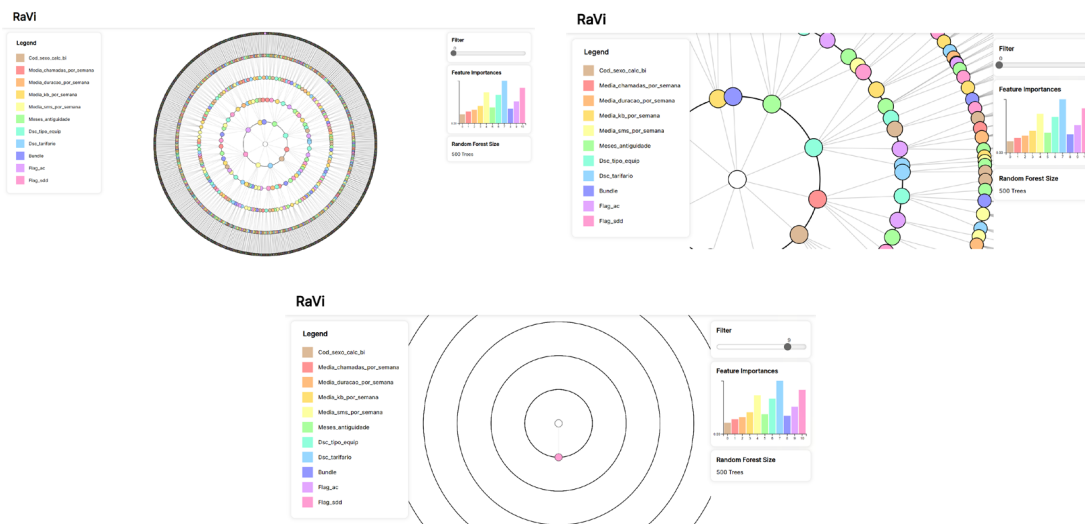


Figure 5.33 - Radial tree screens

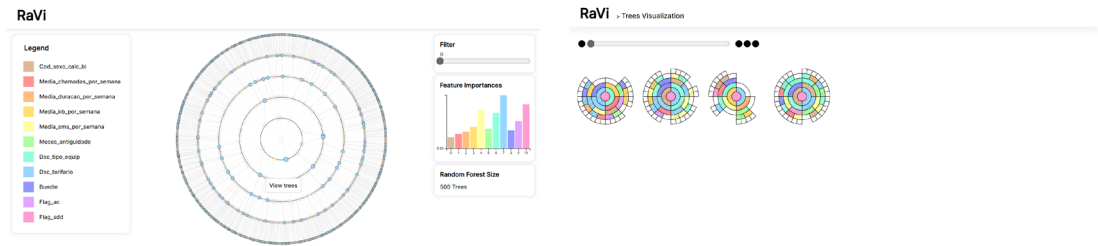


Figure 5.34 - Radial tree and multiple trees screens

In addition to the relationships between features, the user wants to find the most used feature at level 0. There are several features at this level, so the user uses the filter in the top right-hand corner of the screen to filter the tree until the most used is left (Figure 5.33, below). After filtering, the user concludes that the most used feature at level 0 is the feature “Flag\_sdd” (feature 10). This feature is also one of the features with the highest feature importance value.

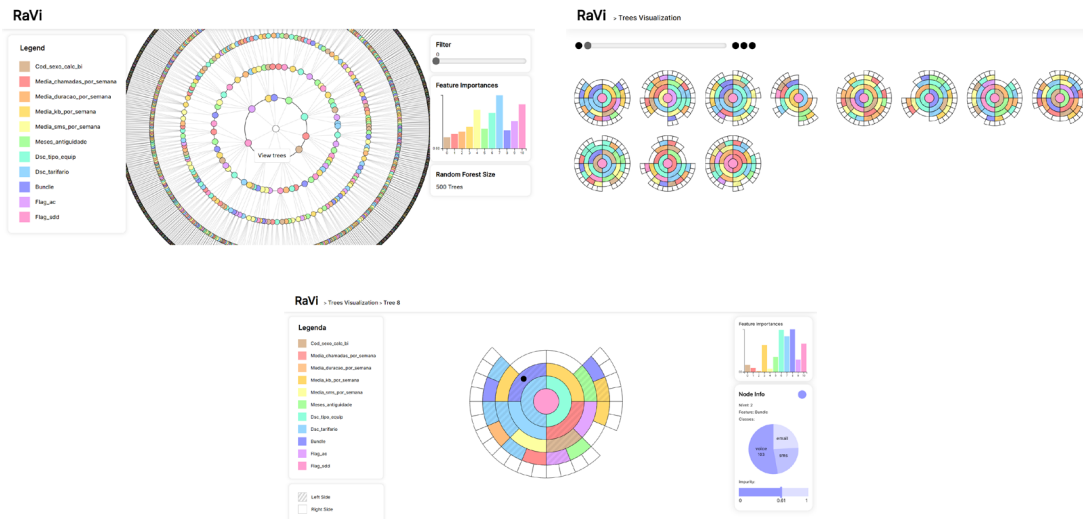
### Case 2: Depth of the feature and how many times it is used

This use case outlines the interaction of a marketer with the RaVi platform to gain insights into a random forest model. The marketer, who lacks prior knowledge of random forests and decision trees, aims to understand how many times a feature is used, in which levels it appears, and in which trees. With this valuable information, the marketer can adapt or create specific marketing strategies or campaigns to target customers falling into these specific categories based on these characteristics.

After entering the RaVi platform, the user comes across a radial tree visualization representing the random forest (Figure 5.33, above, left). However, the user is only interested in highlighting the nodes with the color of the feature “Dsc\_tarifario” to see how many times they appear in the random forest, thus providing insights into the significance of the feature within the model. To highlight these nodes, the user clicks on the legend on the left-hand side of the screen, above the feature they want to highlight (Figure 5.34, left). The user then zooms in on the radial tree and clicks on one of the highlighted nodes on level 1 of the radial tree to find out how many trees contain the features at those levels. This action leads the marketer to a new page, where a compilation of trees adhering to the specified rules is presented: decision trees with the feature “Flag\_sdd” at level 1 (Figure 5.34, right). This can reveal variations in feature relationships and help identify specific conditions or thresholds associated with each feature.

### Case 3: Node of decision tree information

The user, a data scientist, utilizes the RaVi platform to visualize data. The goal is to examine how many samples are in each node at a given level. This information is important when developing the models since it tells how discriminative a certain feature is. For instance, if we have a feature that splits the data, but the number of



**Figure 5.35** - Radial tree (above, left), multiple trees (above, right), and sunburst screens (bellow)

samples in each class is more or less the same at a certain level, it is an indication that we might need more discriminative features and/or deeper models.

The process begins with the user accessing the RaVi framework. Once on the platform, the user engages with this visualization by selecting a particular node of interest within the radial tree (Figure 5.35, above, left). This action triggers a redirection to a different page, where an array of decision trees is presented (Figure 5.35, above, right). Among these options, the user clicks on the first available tree which redirects the user to a dedicated page showcasing the chosen decision tree, displayed as a sunburst. On the sunburst interface, the user continues to interact with the visualization by clicking on the desired node. In response, the platform presents the user with comprehensive details regarding the selected node (Figure 5.35, bellow). This information includes the node's level within the tree hierarchy (Level 2), the feature responsible for the node's split ("Bundle"), a pie chart representing the class distribution within the node, and the node's impurity. By hovering the cursor over each slice of the pie chart, the number of samples per class is shown. Figure 5.35, bellow), emphasizes the number of samples from the class "voice". After looking at the pie chart, the user can perceive that more than half of the samples are attributed to the class "voice", which reveals the importance of such feature (.e., bundle) in this classification.

## 5.7. Validation

This section covers the tests conducted with users to validate our solution. This includes the materials and methods used, the results obtained, and their analysis. Due to the time and period in which the tests were carried out, it was not possible to test the interactivity between the users and the tool, as it would have been necessary to carry out the tests personally, which was not possible. That said, we solely had the

opportunity to carry out tests on the visualizations which, nevertheless, helps us to realize what can be improved in them.

For the execution of the tests, we conducted a questionnaire that has three main components. Firstly, we provided context to the participants in case they are not familiar with the concepts of random forests and decision trees. After presenting how the decision trees are represented in the visualizations, we asked specific questions about each of the pages of the web application. Lastly, we included more general questions about the developed application. Despite the questionnaire being in Portuguese, the questions and answers were translated for the dissertation document.

The questionnaire in question was developed using Google Forms (Appendix E) and is divided into eight sections. The first section is an introduction to the form, informing the participant about its theme, briefly explaining the concepts, and stating its purpose. The second section consists of a brief outline of how decision trees were transformed into the used visualization models—sunburst and radial tree—, providing context for participants with less knowledge about the subject. The third section is related to participant information, where they were asked about their profession and their professional/educational area. Sections from the fourth to the sixth contain specific questions about the characteristics of each page of the web application, to which the participant responds based on what they observe in the images related to the pages. The fourth section, concerning Page 1 of the application, begins with a brief explanation of the visualization model used for the random forest, followed by the following questions:

- Q1:** Can you find the “Media\_chamadas\_por\_semana” feature at level 0?
- Q2:** Can you verify how many decision trees compose the random forest?
- Q3:** Can you identify the feature with the highest feature importance?
- Q4:** Can you easily distinguish which feature should be emphasized?
- Q5:** Can you verify which value is used to filter the radial tree?

The fifth section pertains to Page 2 of the application, which, just like in the previous section, provides a brief explanation of how the visualization model works; in this case, for a decision tree. The questions posed to the participant were as follows:

- Q1:** Can you find where to zoom in on the decision tree grid?
- Q2:** Can you identify the common feature for all trees at level 0?

In the sixth section, we provide images from Page 3 of the tool, followed by the following questions:

- Q1:** Can you identify the selected node?
- Q2:** Can you find the features with the lowest feature importance?
- Q3:** Can you determine the id of the presented tree?
- Q4:** Can you find the impurity value of the selected node?
- Q5:** Can you find the voice class value of the selected node?

The last two sections are dedicated to general questions about the web application. The seventh section presents open-ended questions, while the questions in the eighth section are answered using a semantic differential scale. The semantic differential scale is a type of rating scale found in questionnaires for people to provide ratings

for a product. This scale involves presenting participants with a range of options for rating, with each option consisting of a pair of opposite adjectives (Semantic differential scale, example, and question types, n.d.). Regarding the solution developed, we wanted the participant to evaluate the application in terms of creativity, innovation, professionalism, usefulness, and ease of learning. Finally, in the last section, the participants were encouraged to answer the following open-ended questions:

**Q1:** Did you have difficulty finding any requested information? If yes, what and what would you change?

**Q2:** In your opinion, is the information presented explicitly? If not, what would you change?

**Q3:** Do you think the visualization models are easy to interpret? Please provide justification.

**Q4:** In your opinion, do the colors of the features have sufficient contrast to distinguish them? If not, which ones would you change?

**Q5:** Did you find the application and visualizations easy to use for someone who has no knowledge of random forests and decision trees? If not, what would you change?

### 5.7.1. Results and analysis

The questionnaire was filled out by a total of twenty-seven (27) individuals from the fields of Design and Multimedia, Computer Engineering, Data Science, and Bioinformatics. Out of this total, twenty-one (21) are students, two (2) are researchers, two (2) are data scientists, one (1) is a designer, and one (1) is a developer.

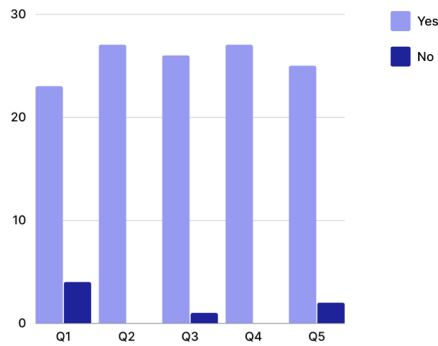
#### Section 4: Page 1 - Random Forest

The questions presented in section 4, as previously mentioned, are related to the first page of the web application, where the visualization of the random forest in the form of a radial tree is located. In this, and the next two sections, the answers are in the form of “Yes” or “No” to understand if the participant could interpret the visualization. As can be seen from the graph in Figure 5.36 (left), in general, most participants were able to interpret the visualization and the other features of the application, and the question they had the most difficulty with was the first one, i.e., “Can you find the “Media\_chamadas\_por\_semana” feature at level 0?”. We deduce that the reason for this difficulty may be that the participants did not fully understand how the levels of the radial tree work.

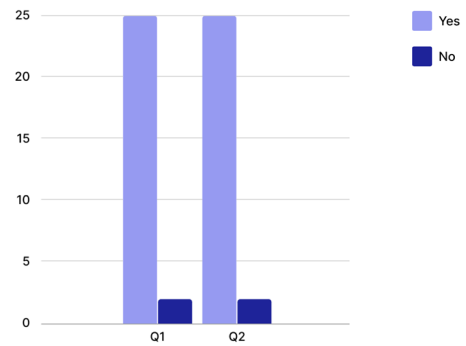
#### Section 5: Page 2 - Decision Trees

The questions presented in section 5, refer to the second page of the web application, which shows a grid with various decision trees. As can be seen from the graph in Figure 5.36 (right), in both questions, most participants were able to interpret what was presented, with only two people answering “No” to each question.

Section 4: Page 1 - Random Forest

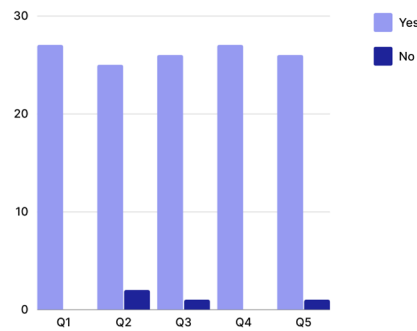


Section 5: Page 2 - Decision Trees



**Figure 5.36** - Section 4: Number of “Yes” and “No” answers per question (left); Section 5: Number of “Yes” and “No” answers per question (right)

Section 6: Page 3 - Sunburst



**Figure 5.37** - Section 6: Number of “Yes” and “No” answers per question

### Section 6: Page 3 - Sunburst

The questions presented in section 6, are associated with the third page of the web application, where solely one decision tree is visualized. Figure 5.37 shows the results obtained in this section, which prove to be favorable, as a large proportion of participants answered “Yes” to all the questions. The question where the participants showed a slight difficulty was the second question, with one of the participants stating that his difficulty was due to the difficulty in distinguishing the colors of the features.

### Section 7: Open-ended questions

#### Question 1

The first question of the seventh section was asked to understand if the participant had difficulty identifying the requested information, which, according to the answers given, most respondents did not have difficulty with. However, some struggled with distinguishing similar colors, especially in the feature importance graph. Overall,

participants found the interface satisfactory but provided suggestions for enhancing visual clarity and ease of use. The zoom functionality was occasionally unclear, especially on the second page, and a few participants suggested clearer icons. A common request was to improve color contrast for better readability, which is a subject that is further explored in one of the following questions. Additionally, one of the participants pointed to the difficulty in understanding what the filter on the radial tree page affected and suggested the implementation of a caption.

### Question 2

The majority of respondents found the information to be explicitly presented and easy to understand. Regarding the colors, some participants suggested adding options for visualizations suitable for colorblind individuals, such as using numbers or textures while other participants showed concerns about distinguishing colors if the number of features increases. Other participants suggested a better identification of the IDs for the sunburst trees, but overall, deemed the information as well-exposed and organized. A few participants also noted that the design was clean and aided in locating information quickly, although they found it beneficial to add explanations of technical terms for the participants with less knowledge of the subject. One of the most relevant suggestions was to improve the representation of how many trees belong to a radial tree node.

### Question 3

The majority of participants found the visualization models to be easy to interpret. They appreciated the clear and intuitive design, including the use of colors, textures, and shapes. Some participants noted that the radial layout of the sunburst trees allowed for a compact yet informative representation. Legends accompanying the visualizations were generally well-organized, aiding in understanding. However, a few participants mentioned potential confusion with too many simultaneous representations in the radial tree. While most found the models straightforward, some acknowledged that people without a background in the field might require a bit more time to understand them. The use of different visual cues like colors and textures for distinguishing elements was widely appreciated, though a few participants suggested minor enhancements such as clearer labeling regarding the filters. Overall, respondents found the models easy to understand, particularly after a brief explanation, and appreciated the available options such as the zoom feature for examining details.

### Question 4

A large portion of the participants believed that the feature colors had sufficient contrast to distinguish them. Some participants, however, pointed out specific color combinations that were less perceptible, especially when colors were close in hue. While many found the colors distinguishable, a few suggested improvements, such as using stronger colors or incorporating a dark mode option. The overall consensus was that the color choices were generally effective, but as the number of features increased or colors became more similar, distinguishing them could become challenging. A few participants also highlighted, once again, potential difficulties for individuals with color vision deficiencies and suggested offering adapted themes for better accessibility. In general, participants appreciated the color choices but noted room for improvement in certain cases.

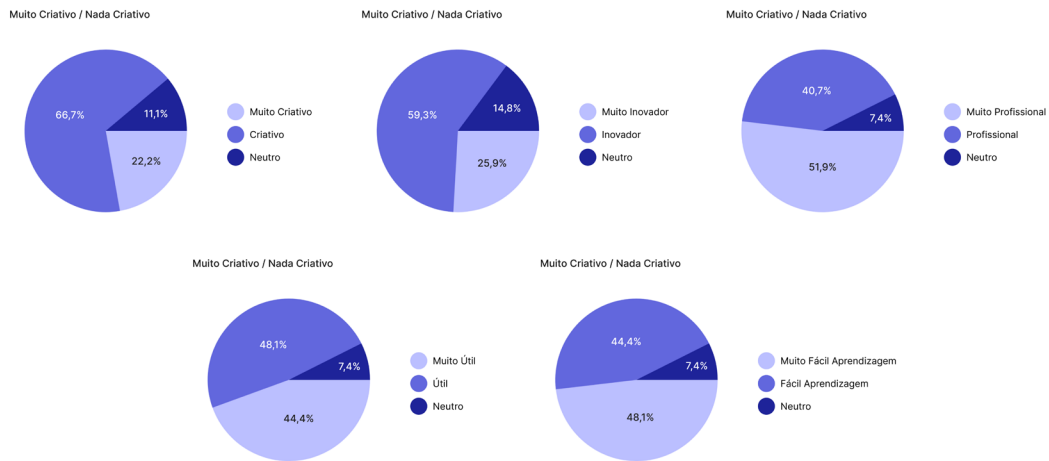


Figure 5.38 - Section 8: Semantic Differential Scale Results

### Question 5

Most participants found the application and visualizations to be user-friendly even for individuals without prior knowledge. It was noted that the descriptions and legends accompanying the visualizations aided in understanding, making the interface intuitive. A few participants mentioned that while the application is easy to navigate, interpreting the data might be challenging for those without background knowledge. Suggestions included adding more explanatory text, buttons for additional information, and explanations for specific terms used. Overall, participants felt that the application was well-designed, with explanations at the start providing helpful context, and many found it accessible even for those with minimal knowledge of decision trees and random forests.

### Section 8: Semantic Differential Scale

Lastly, in the last section, as mentioned previously, we used the semantic differential scale to evaluate a few features of the web application. Consonant to the results shown in Figure 5.38, the balance was very positive. According to the questionnaire participants, it proves to be creative, innovative, very professional, useful, and very easy to learn.

Although the ideal would have been to test the participant's interaction with the tool in person, the tests carried out were an asset in noting the strengths and weaknesses of the visualization and other features of the platform. Overall, we consider the results of the questionnaire to be very positive and the suggestions to be very pertinent and constructive. After analyzing the responses, it should be noted that there are several factors to take into consideration and there are some aspects to be improved, not only in terms of the design of the visualization but also in terms of explaining them to participants who do not have as much knowledge about random forests and decision trees, such as a marketing professional, who is part of our target audience. All these issues will be taken into consideration for future work on improving the application.



## 6. Conclusion

Over the last few years we have witnessed significant transformation in several fields, mainly propelled by the use of Artificial Intelligence (AI). The Marketing field is no exception, where it is imperative to create contemporary and competitive solutions that align with the new patterns of interaction and customer engagement within the digital realm. By doing so, we can effectively address the distinctive requirements and obstacles faced by marketers.

To tackle the challenges mentioned above and to simplify the assessment and categorization of diverse customer profiles, Machine Learning (ML) models have been used. However, these models tend to be opaque in their operations, namely regarding the way they make decisions/predictions. Due to their intricate nature, understanding how and why a particular model generates a specific outcome can be a daunting task. To facilitate the examination of these models for researchers and other end users, Data Visualization (DV) techniques have been put into practice.

The main goal of this work was to develop a tool to help visualize ML models to better understand how they work and interpret the data resulting from these models. This work is part of a real-world project that aims to develop intelligent tools for the analysis and construction of computational models to identify the most convenient channel to contact each customer.

We began by developing a work plan after choosing the methodology to be followed for developing the proposed solution. We then proceeded to briefly research the concepts of Machine Learning, random forests, and decision trees to have an understanding of the study areas in question. Furthermore, in relation to the state of the art, we explored techniques, design principles, and tools that constitute the forefront of Data Visualization and visualization in ML. Once the essential knowledge had been acquired, we moved on to establish the concept for the development of the tool called RaVi, a web application for visualizing machine learning models. Some of the application's features were inspired by different existing platforms, but others were idealized according to the data initially provided. After analyzing the data, we proceeded with various studies on the tree-based visualizations to be chosen until we arrived at an ideal result. After choosing the visualizations, we moved on to creating prototypes of the proposed solution and programming it using D3.js. Finally, after implementing the web application, we carried out tests with a total of twenty-seven (27) participants from different areas, to obtain feedback from the public on the effectiveness of the visualizations developed during the dissertation period.

## 6.1. Limitations and future work

The implementation of the tool so far has helped us to consolidate the idealized concept for solving the problem we were presented with, and we consider it a good starting point. As previously mentioned, it was not possible to carry out tests of user interaction with the platform in person for this goal, which does not allow us to fully evaluate the experience of the web application. In the future, we hope to be able to carry out a new round of tests with more participants from our target audience.

Although the testing situation is not ideal, it is worth noting that, in any case, we have obtained very positive results from the first round. Through the results, we can see that the solution works, but there is also room for improvement according to some of the suggestions that have been made. One of the next steps for this project will be to improve the platform according to these results. Some of the improvements to be made include intensifying the contrast between some of the feature colors, clarifying the legend of the "feature importances" graph, clarifying the function of the random forest filter, and making the zoom of both the random forest and the various trees more explicit. In addition, one of the participants mentioned the importance of adding some initial explanations for users who are less familiar with the concepts of random forests and decision trees, as was done in the questionnaire itself. Through, for example, information buttons, we can not only inform the user about some key concepts but also let them know how the visualization models work, which influences the interpretability of the application.

For this dissertation, we worked with a specific dataset and developed the tool from there. Given that this application is being designed to be used by an entity, we agree that it is essential to ensure that in the future it can be exploited with any type of data that the user wishes to analyze. By changing the type of data, there may be a need to change certain characteristics depending on the data, such as the colors/theme of the application, the visualization of the decision tree, or even the information to be displayed. These choices would become the responsibility of the user. In our opinion, this is a tool with great potential and once it is finalized it will be very useful for the project it is part of.

## 6.2. Reflection

At the start of developing this project, I faced some difficulties due to a lack of knowledge about the area of Data Visualization. As I researched more on the subject, I began to understand the importance that the role of visualization plays not only in the area of ML but also in many other areas. It is through data visualization that we can convey important information to the public in a different, creative, and visually appealing way. By taking part in this project, I have not only put into practice the skills I have acquired so far in the Design and Multimedia course, but I have also acquired new competencies such as learning a new programming language and knowledge about an area of design that, for me, was still unexplored.

With the development of the RaVi web application, we believe we have presented

an innovative solution for visualizing random forests. Although there is still work to be done to improve the tool, we are satisfied with the work carried out during the dissertation period, as we have fulfilled most of the requirements set for this goal.



## 7. References

- Ansari, W. (2022). Understanding the maths behind the Gini impurity method for decision tree split. Analytics India Magazine. <https://analyticsindiamag.com/understanding-the-maths-behind-the-gini-impurity-method-for-decision-tree-split/>
- Barlow, T., & Neville, P. (2005). Case study: visualization for decision tree analysis in data mining. EEE Symposium on Information Visualization, 2001. INFOVIS 2001. [Preprint]. <https://doi.org/10.1109/infvis.2001.963292>
- Berinato, S. (2020, September 14). Visualizations that really work. Harvard Business Review. <https://hbr.org/2016/06/visualizations-that-really-work>
- Biau, G., & Scornet, E. (2016). A random forest guided tour. *Test*, 25(2), 197–227. <https://doi.org/10.1007/s11749-016-0481-7>
- Bishop, C. M. (2006). Pattern recognition and machine learning. Springer Verlag.
- Brush, K., & Burns, E. (2022). data visualization. Business Analytics. <https://www.techtarget.com/searchbusinessanalytics/definition/data-visualization>
- Card, S. K., Mackinlay, J. D., & Shneiderman, B. (1999). Readings in Information Visualization: Using Vision to Think. <http://luizrodrigues.com/artigos-tcc/Information%20Visualization/Readings%20in%20Information%20Visualization%2C%20Using%20vision%20to%20think.pdf>
- Chen, L. (2021, December 7). Basic ensemble learning (Random Forest, AdaBoost, Gradient boosting)- step by step explained. Medium. <https://towardsdatascience.com/basic-ensemble-learning-random-forest-adaboost-gradient-boosting-step-by-step-explained-95d49d1e2725>
- Create Bar Chart using D3. (n.d.). <https://www.tutorialsteacher.com/d3js/create-bar-chart-using-d3js>
- Daly, N. (2023). What is a use case? Blog Wrike. <https://www.wrike.com/blog/what-is-a-use-case/#How-to-write-a-use-case-for-a-project>
- Dam, R. F., & Siang, T. Y. (2023, August 29). What kind of prototype should you create? The Interaction Design Foundation. <https://www.interaction-design.org/literature/article/what-kind-of-prototype-should-you-create>
- Data Visualization Definition and Examples | Microsoft Visio. (n.d.). <https://www.microsoft.com/en-us/microsoft-365/visio/data-visualization>
- Data Visualization in Machine Learning - JavatPoint. (n.d.). [www.javatpoint.com. https://www.javatpoint.com/data-visualization-in-machine-learning](https://www.javatpoint.com/data-visualization-in-machine-learning)
- Dong, Y., Fauth, A., Huang, M. L., Chen, Y., & Liang, J. (2020). PansyTree: Merging Multiple Hierarchies. IEEE Pacific Visualization Symposium. <https://doi.org/10.1109/pacificvis48177.2020.1007>
- Doshi-Velez, F., & Kim, B. (2017). Towards A Rigorous Science of Interpretable Machine Learning. arXiv (Cornell University). <https://doi.org/10.48550/arxiv.1702.08608>
- Etemad, K., Baur, D., Brosz, J., Carpendale, S., & Samavati, F. (2014). PaisleyTrees:

- A Size-Invariant Tree Visualization. EAI Endorsed Transactions on Creative Technologies, 1(1), e2. <https://doi.org/10.4108/ct.1.1.e2>
- Gill. (2023, April 4). Data Visualization Best Practices: How to create Effective visuals for your data | Amphy blog. Amphy Blog. <https://blog.amphy.com/data-visualization-best-practices/>
- Graphical integrity. (n.d.). <https://jcsites.juniata.edu/faculty/rhodes/ida/graphicalIntRedes.html>
- Holtz, Y. (n.d.). Basic pie chart in d3.js. [https://d3-graph-gallery.com/graph/pie\\_basic.html](https://d3-graph-gallery.com/graph/pie_basic.html)
- How to design an information visualization. (2023, August 29). The Interaction Design Foundation. <https://www.interaction-design.org/literature/article/how-to-design-an-information-visualization>
- Hulstaert, L. (2018, July 9). Interpreting machine learning models - Towards Data Science. Medium. <https://towardsdatascience.com/interpretability-in-machine-learning-70c30694a05f>
- Jha, A. (2023). Visualizing Machine Learning Models: How to guide and tools. neptune.ai. <https://neptune.ai/blog/visualizing-machine-learning-models>
- Kapil, A. R. (2022, October 1). Advantages and disadvantages of decision tree in machine learning. Blogs & Updates on Data Science, Business Analytics, AI Machine Learning. [https://www.analytixlabs.co.in/blog/decision-tree-algorithm/#Advantages\\_and\\_Disadvantages\\_of\\_Decision\\_Tree](https://www.analytixlabs.co.in/blog/decision-tree-algorithm/#Advantages_and_Disadvantages_of_Decision_Tree)
- Kuznetsova, N. (2014, August). Random forest visualization. Eindhoven University of Technology Research Portal. <https://research.tue.nl/en/studentTheses/random-forest-visualization>
- Layer Software GmbH. (2023). What is data visualization? Process, tools, best practices. Layer Blog. <https://blog.golayer.io/business/data-visualization>
- Lima, M. (2014). The Book of Trees: Visualizing Branches of Knowledge. Princeton Architectural Press.
- Maças, C., Polisciuc, E. (2021) “Material teórico da disciplina de Visualização de Informação,” 1\_VI\_Process\_Principles, Mestrado em Design e Multimédia
- Melo, C. (2023). Redes Neurais Multicamadas com Python e Keras. Sigmoidal. <https://sigmoidal.ai/redes-neurais-python-keras-2/>
- Molnar, C. (2020). Interpretable Machine learning. Lulu.com.
- Munzner, T. (2009). A nested model for visualization design and validation. IEEE Transactions on Visualization and Computer Graphics, 15(6), 921–928. <https://doi.org/10.1109/tvcg.2009.111>
- Pandey, P. (2023, May 23). Visualizing Decision Trees with Pybaobabdt - Towards Data Science. Medium. <https://towardsdatascience.com/visualizing-decision-trees-with-pybaobabdt-f8eb5b3d0d17>
- PenBox Level 2 (forked) - CodeSandbox. (2023, April 18). CodeSandbox. <https://codesandbox.io/s/penbox-level-2-forked-cz4ifi?file=/src/index.js:39169-39616>
- Rudin, C., & Radin, J. (2019). Why Are We Using Black Box Models in AI When We Don't Need To? A Lesson From An Explainable AI Competition. Harvard Data Science Review, 1(2). <https://doi.org/10.1162/99608f92.5a8a3a3d>
- Sakkaf, Y. (2021, December 14). Decision Trees: ID3 algorithm explained | Towards data science. Medium. <https://towardsdatascience.com/decision-trees-for-classification-id3-algorithm-explained-89df76e72df1>

- Scalco, F. (2021). Visualização de Dados em Processos de Machine Learning [Trabalho de Conclusão de Curso]. Universidade de Caxias do Sul.
- Schilling, J. (2022). A design process for Dataviz. Nightingale. <https://nightingaledvs.com/a-design-process-for-dataviz/>
- Semantic differential scale, example, and question types | QuestionPro. (n.d.). <https://www.questionpro.com/semantic-differential-scale.html>
- Somvanshi, M., & Chavan, P. (2016). A review of machine learning techniques using decision tree and support vector machine. 2016 International Conference on Computing Communication Control and Automation (ICCUBEA) [Preprint]. <https://doi.org/10.1109/iccubea.2016.7860040>
- Spring, M. 2. (2019, June 10). Chapter 2 Fundamentals | A Reader on Data Visualization. [https://mschermann.github.io/data\\_viz\\_reader/fundamentals.html#design-principles](https://mschermann.github.io/data_viz_reader/fundamentals.html#design-principles)
- Sunburst Tutorial (d3 v4), Part 1. (n.d.). Gist. <https://gist.github.com/denjn5/e1cdbbe586ac31747b4a304f8f86efa5>
- Teoh, S. T., & Ma, K. (2003). PaintingClass. Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining [Preprint]. <https://doi.org/10.1145/956750.956837>
- Thorn, J. (2021, December 13). Decision Trees explained - towards data science. Medium. <https://towardsdatascience.com/decision-trees-explained-3ec41632ceb6>
- Tufte, E. R. (2001). The visual display of quantitative information.
- Van Den Elzen, S., & Van Wijk, J. J. (2011). BaobabView: Interactive construction and analysis of decision trees. 2011 IEEE Conference on Visual Analytics Science and Technology (VAST) [Preprint]. <https://doi.org/10.1109/vast.2011.6102453>
- Ward, M. O., Grinstein, G., & Keim, D. A. (2015). Interactive Data Visualization: Foundations, Techniques, and Applications, second edition. [https://openlibrary.org/books/OL28550879M/Interactive\\_Data\\_Visualization](https://openlibrary.org/books/OL28550879M/Interactive_Data_Visualization)
- What is a Decision Tree | IBM. (n.d.). <https://www.ibm.com/topics/decision-trees>
- What is a Mockup and its Role in Design? (n.d.). <https://miro.com/>. <https://miro.com/mockup/what-is-a-mockup/>
- What is a Wireframe & Its Role in the Design Process | Miro. (n.d.). <https://miro.com/>. <https://miro.com/wireframe/what-is-a-wireframe/#wireframe-vs-mockup-what%E2%80%99s-the-difference?>
- What is D3? | D3 by Observable. (n.d.). <https://d3js.org/what-is-d3>
- What is Data Visualization? | IBM. (n.d.). <https://www.ibm.com/topics/data-visualization>
- What is Random Forest? | IBM. (n.d.). <https://www.ibm.com/topics/random-forest>
- Yadav, P. (2019, September 23). Decision tree in Machine Learning - towards data science. Medium. <https://towardsdatascience.com/decision-tree-in-machine-learning-e380942a4c96>
- Yiu, T. (2021, December 10). Understanding random Forest - towards data science. Medium. <https://towardsdatascience.com/understanding-random-forest-58381e0602d2>
- Yse, D. L. (2021, December 14). How to Prepare your Data - Towards Data Science. Medium. <https://towardsdatascience.com/the-basics-of-data-prep-7bb5f3af77ac>





# Appendix A

## Visualization of Decision Trees questionnaire

### Visualização de Árvores de Decisão

Este formulário está a ser realizado no âmbito da dissertação "Visualização de Algoritmos de Machine Learning" para o Mestrado em Design e Multimédia.

O principal objectivo desta dissertação é o desenvolvimento de uma aplicação web que, através de métodos de visualização de informação, auxilie a análise e melhoria de modelos de aprendizagem máquina. A técnica de visualização utilizada para a análise de árvores de decisão consiste em *sunbursts* (um exemplo deste tipo de modelos pode ser visto na imagem abaixo).

Através deste inquérito, pretende-se encontrar a maneira mais eficaz e visualmente apelativa de distinguir o sentido da ramificação da árvore.

**Nota:** Ao submeter este formulário, está a consentir partilhar as informações solicitadas, sendo que estas serão usadas apenas para o propósito acima descrito.

eduarda.edsilva.silva@gmail.com [Mudar de conta](#)

🔒 Não partilhado

Imagem retirada do livro "The Book of Trees: Visualizing Branches of Knowledge" de Manuel Lima (2014)

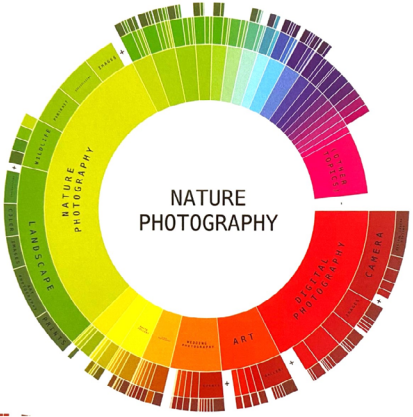


Figure A.1 - Visualization of Decision trees questionnaire: introduction



Figure A.2 - Visualization of Decision trees questionnaire: visualization models

The screenshot shows a web interface for user information collection. The title is "Visualização de Árvores de Decisão". Below the title, there is a user profile section with the email "eduarda.edysilva.silva@gmail.com" and a "Mudar de conta" link. A "Não partilhado" status is also visible. A red asterisk indicates a mandatory question: "\* Indica uma pergunta obrigatória".

The main content area is titled "Informação do utilizador" and contains two questions, each with radio button options:

Qual a sua área de trabalho/estudo? \*

- Design e Multimédia
- Informática
- Ciência de Dados
- Outra: \_\_\_\_\_

Qual a sua profissão? \*

- Estudante
- Professor(a)
- Investigador(a)
- Outra: \_\_\_\_\_

Figure A.3 - Visualization of Decision trees questionnaire: user information

**Visualização de Árvores de Decisão**

eduarda.edysilva.silva@gmail.com [Mudar de conta](#)

🔒 Não compartilhado

\* Indica uma pergunta obrigatória

**Visualização da Árvore**

Na sua opinião, qual das abordagens abaixo é a mais indicada para distinguir o lado esquerdo do lado direito das várias ramificações da árvore?

\*

NO TEXTURE

OUTLINE

..... Left  
—— Right

TEXTURE

▨ Left  
□ Right

LEFT-RIGHT

L Left  
R Right

BALL

Left

No texture

Outline

Texture

Left-Right

Ball

---

\*

NO TEXTURE

OUTLINE

..... Left  
—— Right

TEXTURE

▨ Left  
□ Right

LEFT-RIGHT

L Left  
R Right

BALL

Left

No texture

Outline

Texture

Left-Right

Ball

---

\*

NO TEXTURE

OUTLINE

..... Left  
—— Right

TEXTURE

▨ Left  
□ Right

LEFT-RIGHT

L Left  
R Right

BALL

Left

No texture

Outline

Texture

Left-Right

Ball

Figure A.4 - Visualization of Decision trees questionnaire: tree visualizations

The image shows a Google Forms interface for a questionnaire titled "Visualização de Árvores de Decisão". At the top, the user's email "eduarda.edysilva.silva@gmail.com" is displayed with a "Mudar de conta" link and a cloud icon. Below this, it indicates "Não compartilhado". The main section is titled "Comentários" and contains the question "Gostaria de fazer alguma observação acerca da visualização?". A text input field is labeled "A sua resposta". At the bottom of the form, there are three buttons: "Anterior", "Enviar", and "Limpar formulário". Below the form, a disclaimer states "Nunca envie palavras-passe através dos Google Forms." and provides links for "Denunciar abuso", "Termos de Utilização", and "Política de privacidade". The Google Forms logo is at the bottom center.

Figure A.5 - Visualization of Decision trees questionnaire: comments

# Appendix B

## Web application wireframes

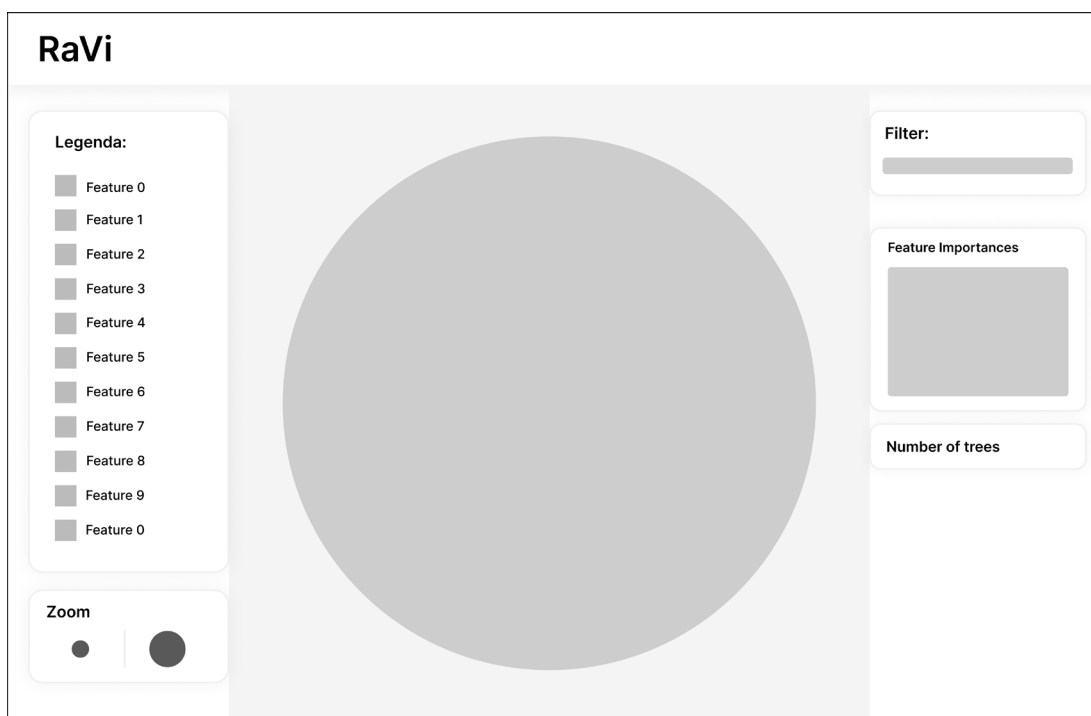


Figure B.1 - Web application wireframes: random forest visualization screen

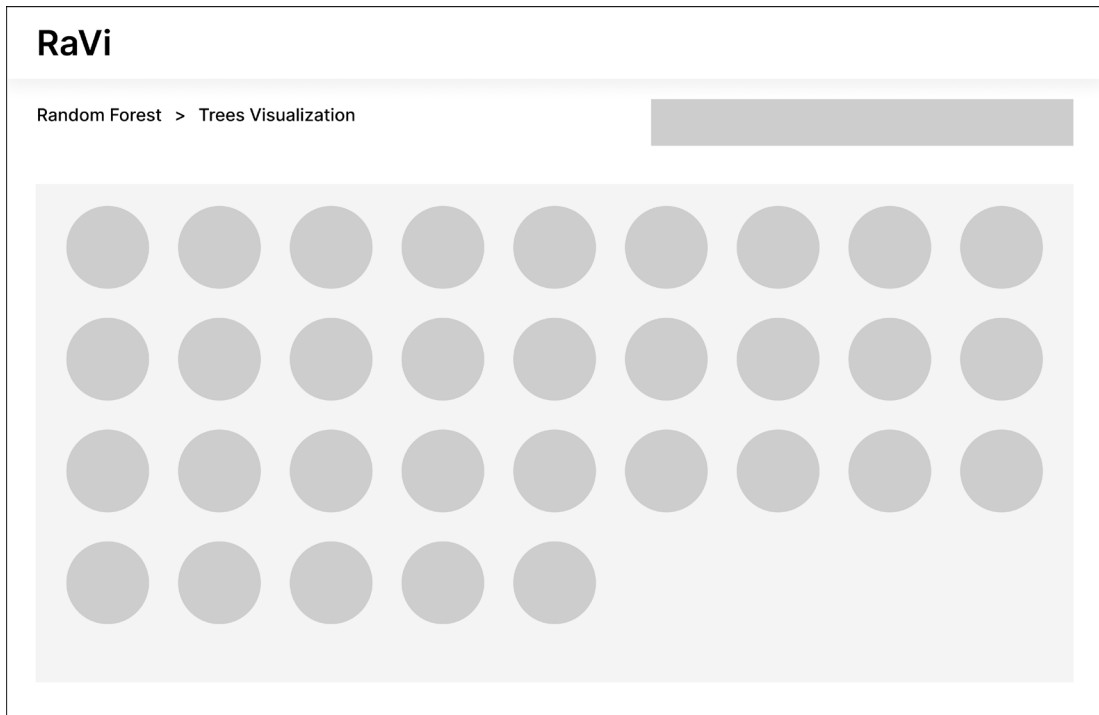


Figure B.2 - Web application wireframes: multiple trees visualization screen

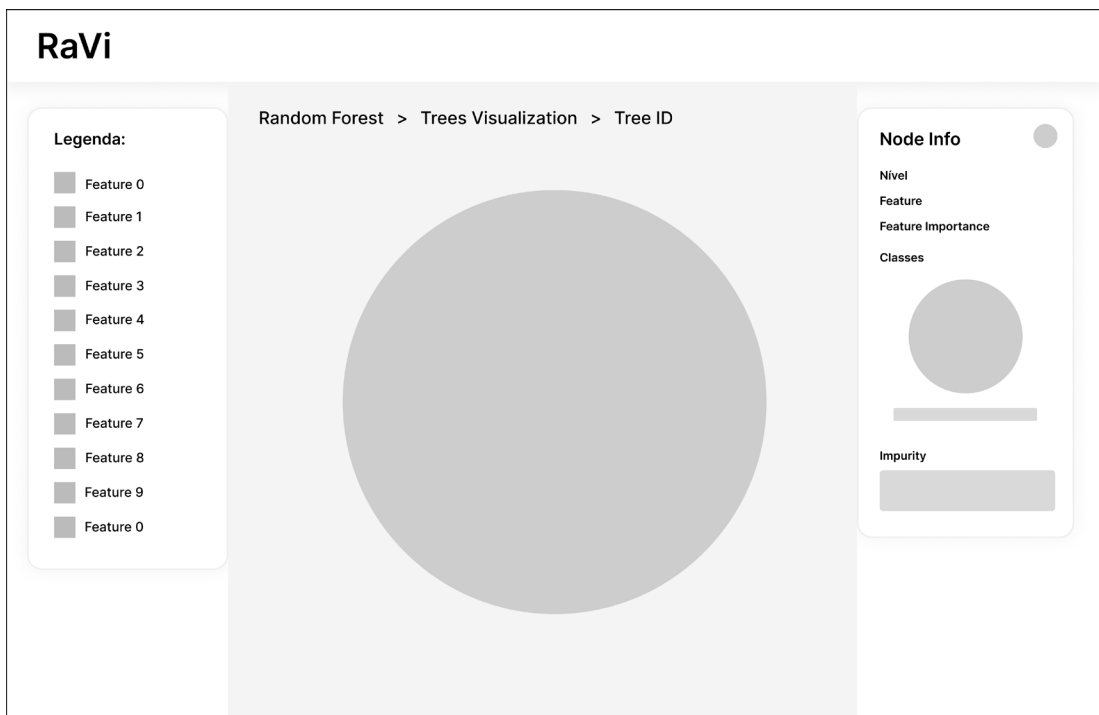


Figure B.3 - Web application wireframes: decision visualization screen

# Appendix C

## Web application mockups

### RaVi



Figure C.1 - Web application mockups: radial tree visualization screen

# RaVi



Figure C.2 - Web application mockups: radial tree filter screen

# RaVi

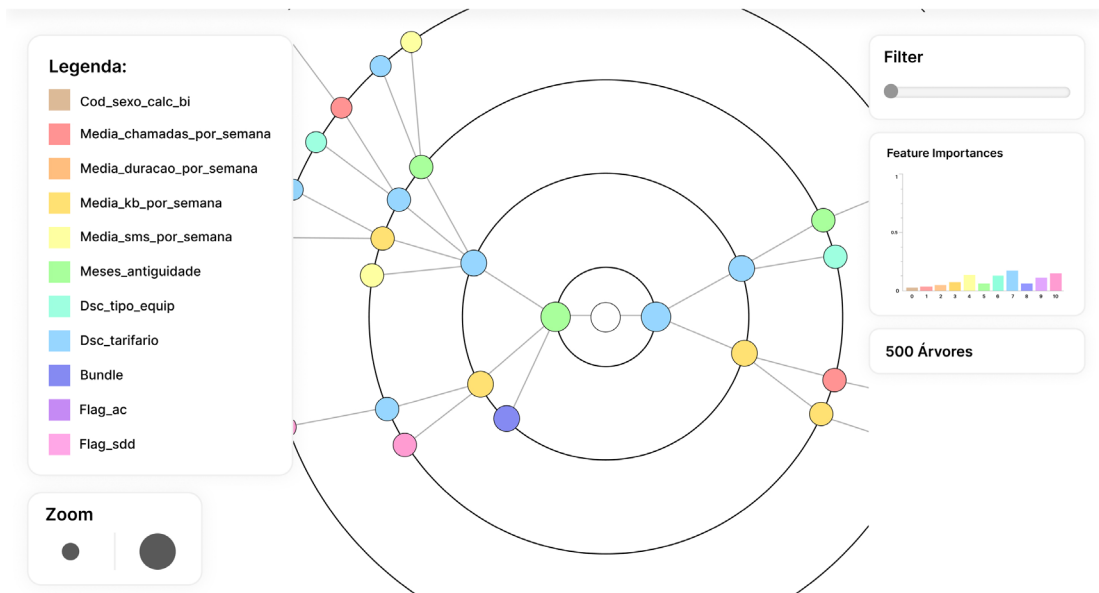


Figure C.3 - Web application mockups: radial tree zoom screen



# RaVi

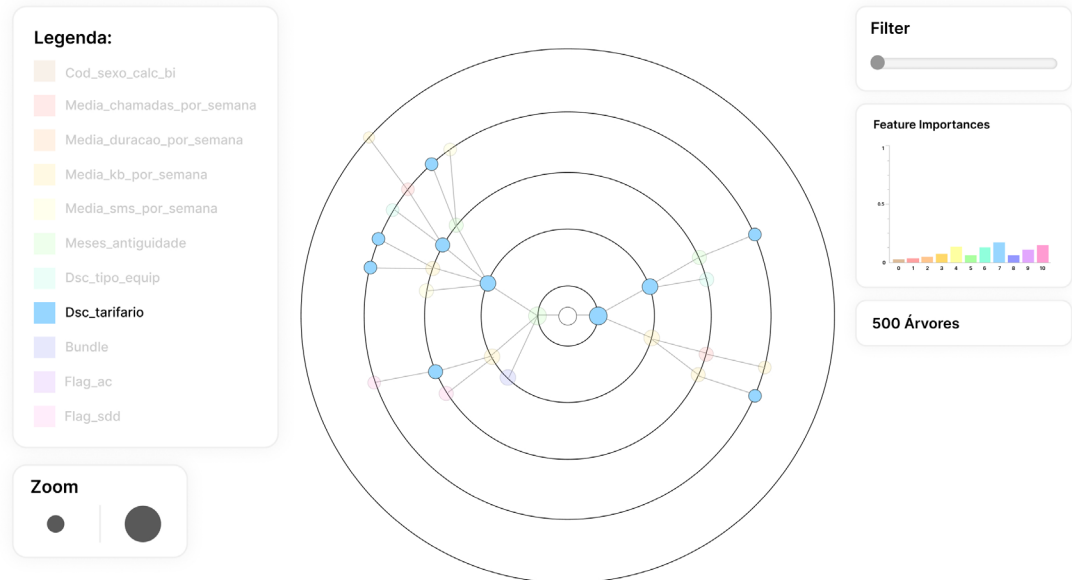


Figure C.4 - Web application mockups: radial tree legend filter screen

# RaVi

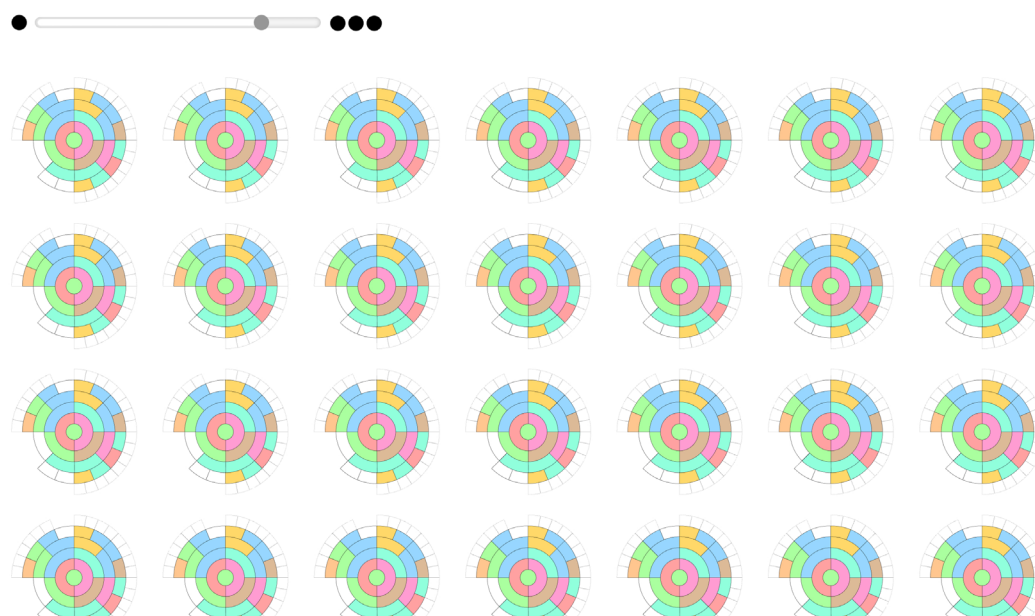


Figure C.5 - Web application mockups: multiple tree visualization screen

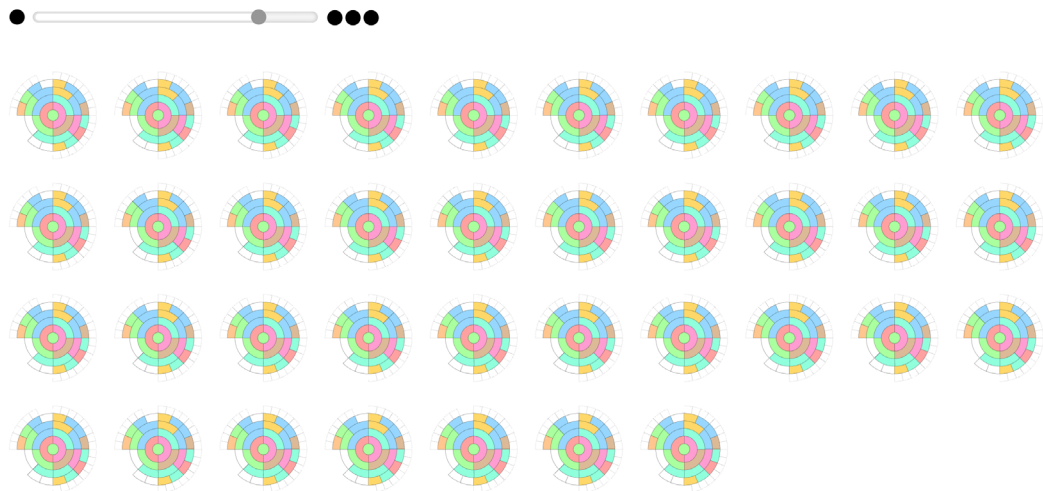


Figure C.6 - Web application mockups: multiple trees visualization zoom screen

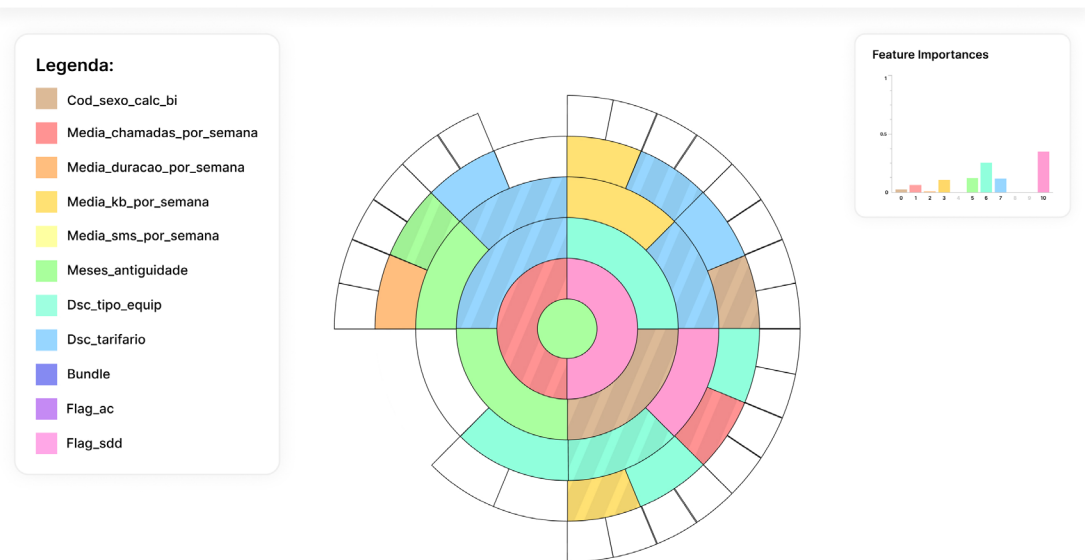


Figure C.7 - Web application mockups: sunburst visualization screen

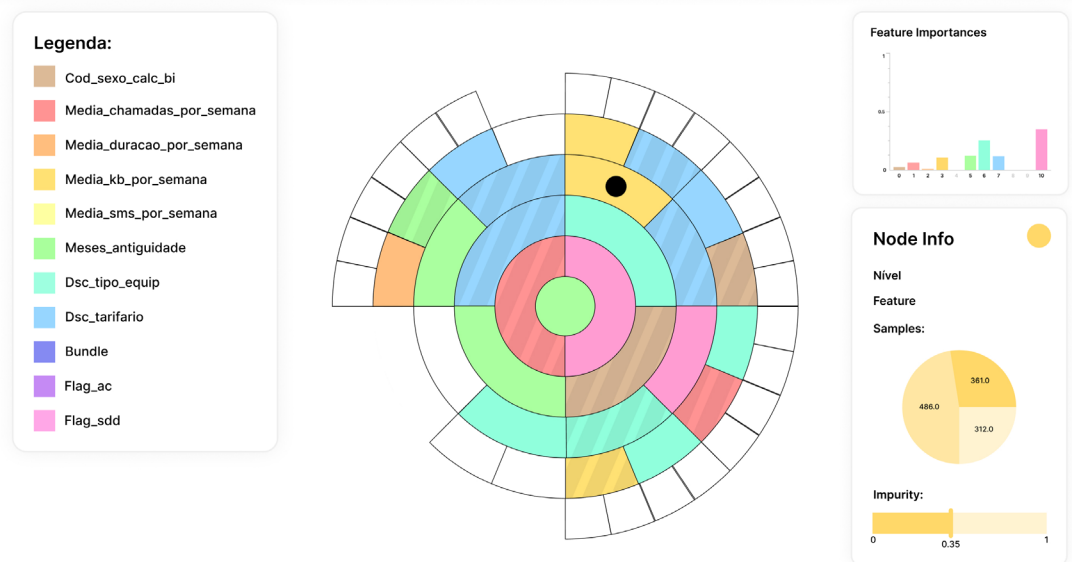


Figure C.8 - Web application mockups: node information screen



# Appendix D

## Use cases implemented screens

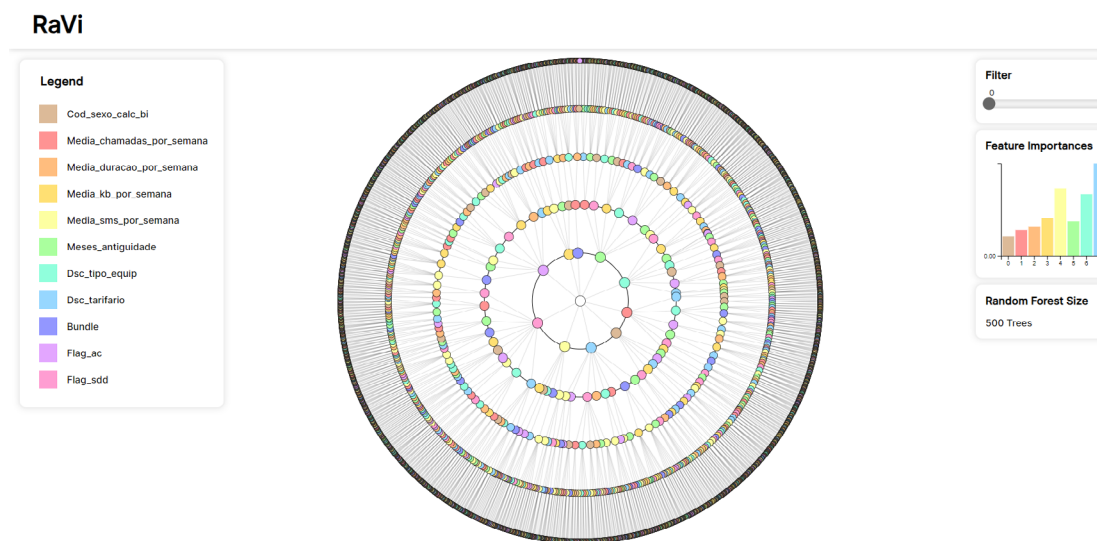


Figure D.1 - Implemented screens: radial tree visualization

## RaVi

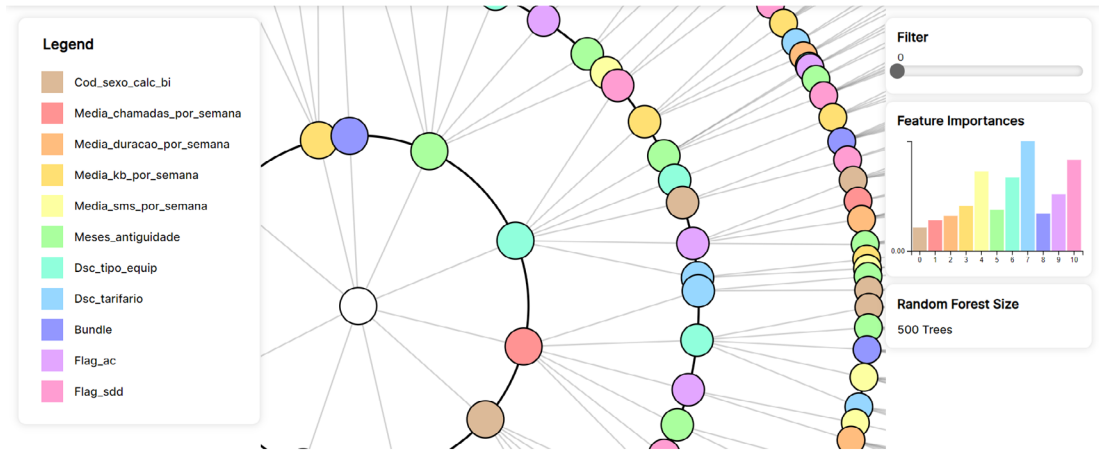


Figure D.2 - Implemented screens: radial tree visualization zoom

## RaVi

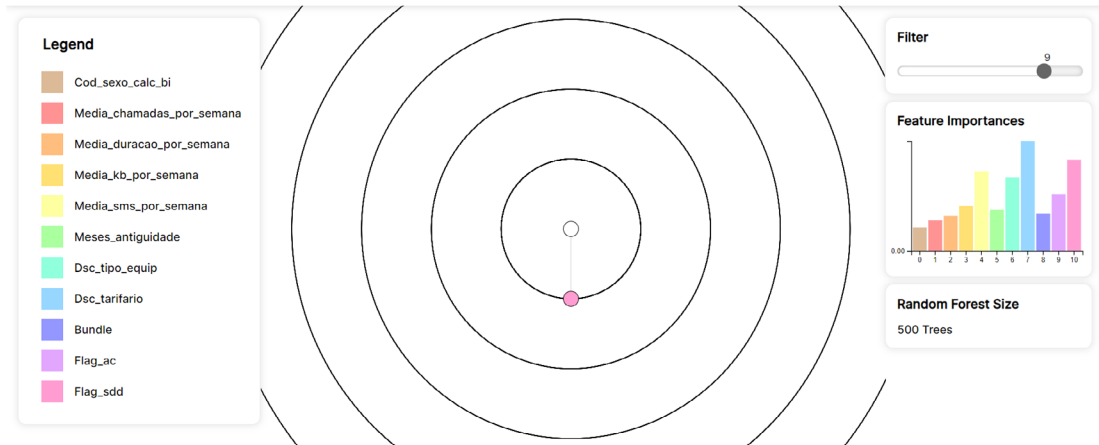


Figure D.3 - Implemented screens: radial tree visualization filter

## RaVi

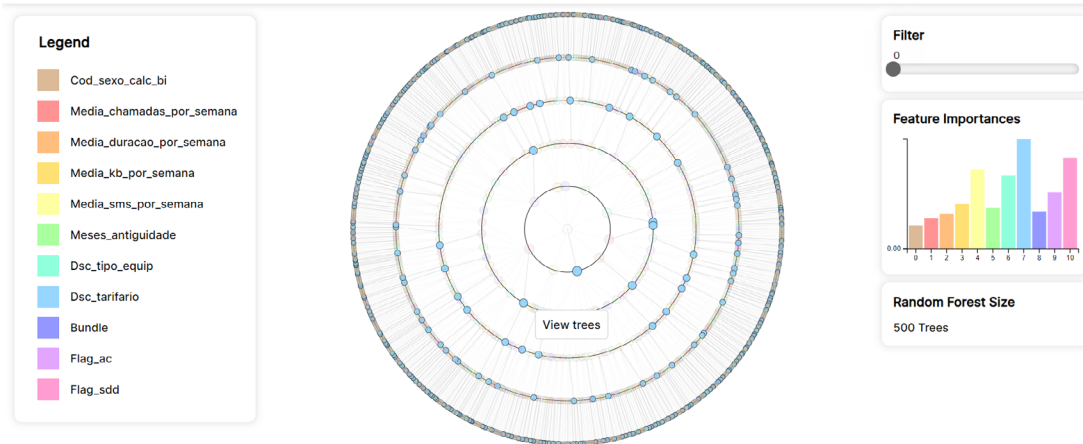


Figure D.4 - Implemented screens: radial tree legend filter

## RaVi

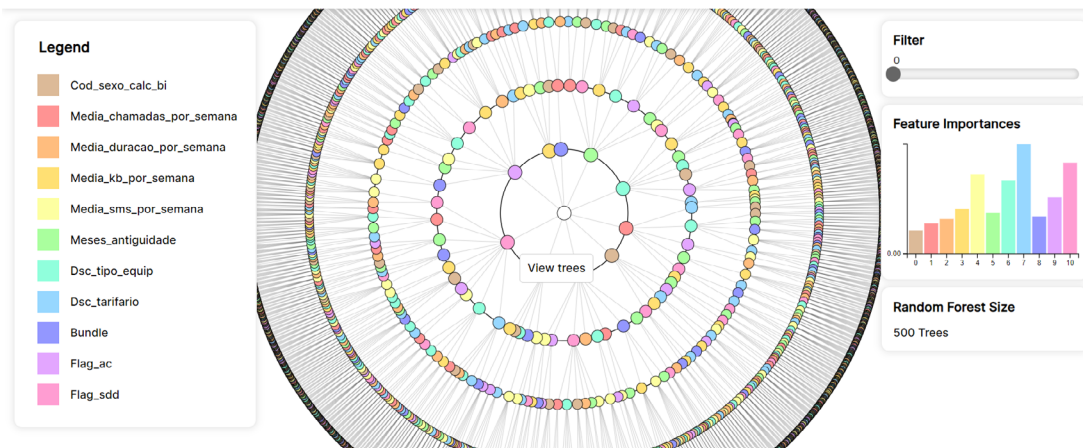


Figure D.5 - Implemented screens: radial tree visualization cursor hover

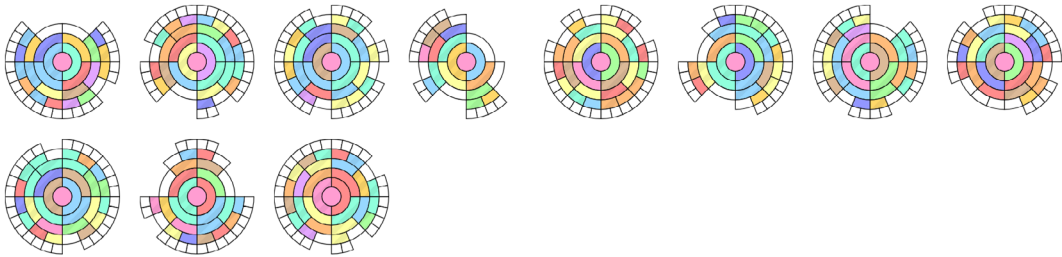


Figure D.6 - Implemented screens: multiple tree visualization

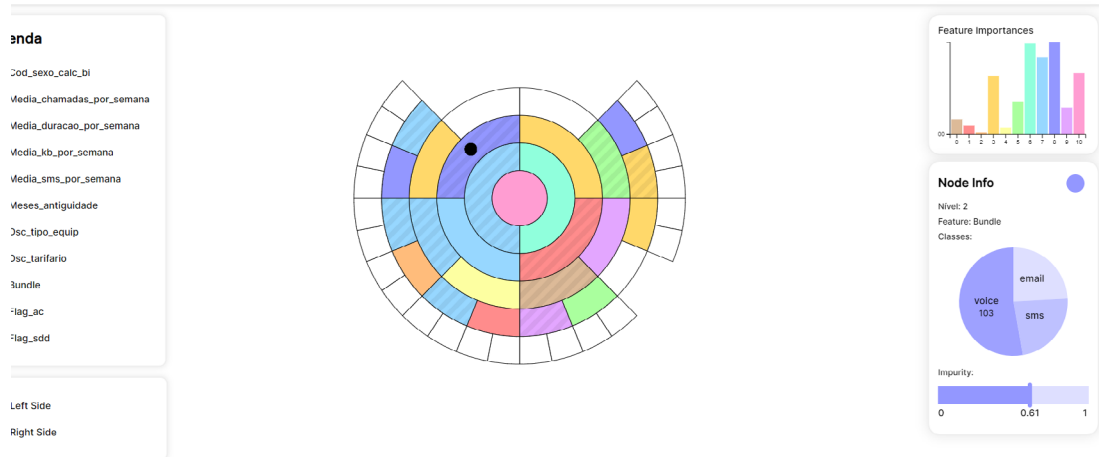
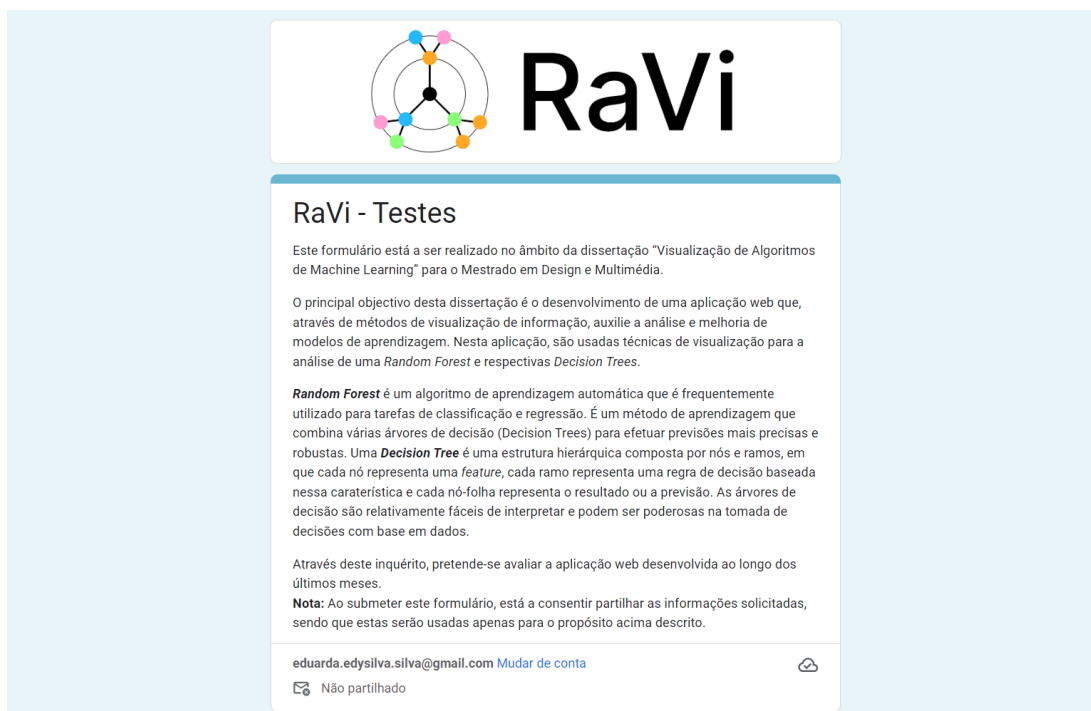


Figure D.7 - Implemented screens: sunburst visualization



# Appendix E

## RaVi validation questionnaire



The image shows a screenshot of a web-based questionnaire titled "RaVi - Testes". At the top, there is a logo for "RaVi" which consists of a network diagram with nodes and edges, followed by the text "RaVi" in a large, bold font. Below the logo, the title "RaVi - Testes" is displayed. The main content of the questionnaire is in Portuguese and includes the following text:

Este formulário está a ser realizado no âmbito da dissertação "Visualização de Algoritmos de Machine Learning" para o Mestrado em Design e Multimédia.

O principal objectivo desta dissertação é o desenvolvimento de uma aplicação web que, através de métodos de visualização de informação, auxilie a análise e melhoria de modelos de aprendizagem. Nesta aplicação, são usadas técnicas de visualização para a análise de uma *Random Forest* e respectivas *Decision Trees*.

**Random Forest** é um algoritmo de aprendizagem automática que é frequentemente utilizado para tarefas de classificação e regressão. É um método de aprendizagem que combina várias árvores de decisão (*Decision Trees*) para efetuar previsões mais precisas e robustas. Uma **Decision Tree** é uma estrutura hierárquica composta por nós e ramos, em que cada nó representa uma *feature*, cada ramo representa uma regra de decisão baseada nessa característica e cada nó-folha representa o resultado ou a previsão. As árvores de decisão são relativamente fáceis de interpretar e podem ser poderosas na tomada de decisões com base em dados.

Através deste inquérito, pretende-se avaliar a aplicação web desenvolvida ao longo dos últimos meses.

**Nota:** Ao submeter este formulário, está a consentir partilhar as informações solicitadas, sendo que estas serão usadas apenas para o propósito acima descrito.

At the bottom of the questionnaire, there is a footer containing the email address "eduarda.edysilva.silva@gmail.com" with a link "Mudar de conta" and a cloud icon. Below that, there is a lock icon and the text "Não partilhado".

Figure E.1 - RaVi validation questionnaire: introduction section



## RaVi - Testes

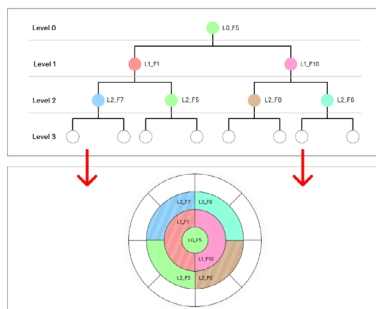
eduarda.edsilva.silva@gmail.com [Mudar de conta](#)  
Não partilhado

### Transformação das Decision Trees para os Modelos de Visualização

Para uma melhor compreensão de como os modelos de visualização foram criados, são apresentados dois esquemas como exemplo.

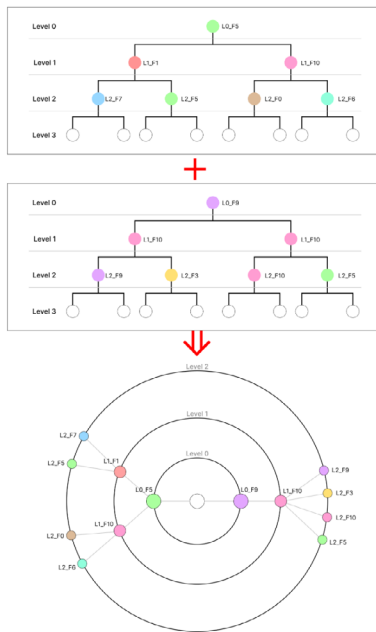
#### Decision Tree -> Sunburst

Cada nó da árvore, é representado no *sunburst* com um arco, sendo que o nó-raiz da árvore (L0\_F5) está posicionado no círculo interno do *sunburst* e os restantes nós dos diferentes níveis vão sendo colocados no exterior do *sunburst*. Os nós-folha (a branco) são também coloridos a branco no exterior do *sunburst*. Os nós à esquerda de cada divisão na árvore são identificados no *sunburst* através de uma textura.



#### 2 Decision Trees -> Radial Tree (Random Forest)

Neste exemplo, mostramos como duas árvores—resultantes do random forest—podem ser representadas na *Radial Tree*. Este modelo pretende ser um sumário de todas as regras das árvores do modelo.



[Anterior](#)

[Seguinte](#)

[Limpar formulário](#)


Nunca envie palavras-passe através dos Google Forms.

Este conteúdo não foi criado nem aprovado pelo Google. [Denunciar abuso](#) [Termos de Utilização](#) [Política de privacidade](#)

Google Formulários



Figure E.2 - RaVi validation questionnaire: models transformation section



# RaVi

## RaVi - Testes

eduarda.edysilva.silva@gmail.com [Mudar de conta](#)

Não partilhado

\* Indica uma pergunta obrigatória

### Informação do utilizador

Qual a sua profissão? \*

- Estudante
- Professor(a)
- Investigador(a)
- Outra: \_\_\_\_\_

Qual a sua área de trabalho/estudo? \*

- Design e Multimédia
- Informática
- Ciência de Dados
- Marketing
- Outra: \_\_\_\_\_

Figure E.3 - RaVi validation questionnaire: user information section





# RaVi

## RaVi - Testes

eduarda.edysilva.silva@gmail.com [Mudar de conta](#)

Não compartilhado

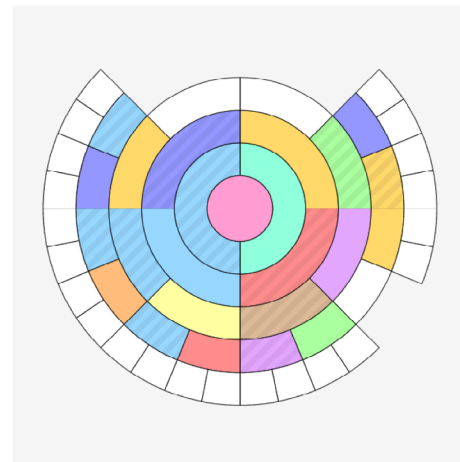
\* Indica uma pergunta obrigatória

### Página 2 - Decision Trees

Observe as imagens e leia os textos com atenção e responda às seguintes questões.

#### Decision Tree

Para a *decision tree*, foi escolhido o modelo de visualização denominado de *sunburst*. O *sunburst*, que representa uma árvore, começa com um nó-raiz central e as restantes filas, que expandem-se para fora a partir do meio, recorrendo a uma sequência de anéis segmentados. Esses anéis contêm as mesmas cores correspondentes às *features*, com exceção dos nós da última fila (*leaf nodes*) pois estes já não se dividem mais. Para além disso, foi adicionada uma textura aos nós para poder distinguir o lado esquerdo do lado direito, sendo que a textura está presente nos nós do lado esquerdo.



Esta imagem representa a página da aplicação web relativa à visualização de múltiplas *Decision Trees* através de uma grelha dinâmica.



Consegue encontrar onde fazer zoom na grelha de *decision trees*? \*


- Sim  
 Não

Consegue identificar a feature comum a todas as árvores no nível 0? \*

- Sim  
 Não

Figure E.5 - RaVi validation questionnaire: multiple tree section





# RaVi

## RaVi - Testes

eduarda.edysilva.silva@gmail.com [Mudar de conta](#)

Não partilhado

\* Indica uma pergunta obrigatória

### Questões de resposta aberta

De uma forma breve, responda às seguintes questões.

Teve dificuldade a encontrar alguma informação solicitada? Se sim, qual e o que mudaria? \*

A sua resposta

Na sua opinião, a informação está exposta de forma explícita? Se não, o que mudaria? \*

A sua resposta

Pensa que os modelos de visualização são de fácil interpretação? Justifique. \*

A sua resposta


Na sua opinião, as cores das *features* têm contraste suficiente para as distinguir? Se não, quais mudaria? \*

A sua resposta

Achou a aplicação e as visualizações de simples uso para quem não tem conhecimento sobre *random forests* e árvores de decisão? Se não, o que mudaria? \*

A sua resposta

Figure E.7 - RaVi validation questionnaire: open-ended answer section



# RaVi

---

## RaVi - Testes

[eduarda.edysilva.silva@gmail.com](mailto:eduarda.edysilva.silva@gmail.com)
[Mudar de conta](#)
🔒

🔒 Não partilhado

\* Indica uma pergunta obrigatória

---

### Semantic Differential Scale

Selecione em cada conjunto de palavras, o que corresponde melhor ao que sentiu durante a interação com a aplicação RaVi.

---

\*

- Muito Criativo
- Criativo
- Neutro
- Pouco Criativo
- Nada Criativo

\*

- Muito Inovador
- Inovador
- Neutro
- Pouco Inovador
- Básico

\*

- Muito Profissional
- Profissional
- Neutro
- Pouco Profissional
- Antiprofissional

\*

- Muito Útil
- Útil
- Neutro
- Pouco Útil
- Inútil

\*

- Muito Fácil Aprendizagem
- Fácil Aprendizagem
- Neutro
- Difícil Aprendizagem
- Muito Difícil Aprendizagem

Figure E.8 - RaVi validation questionnaire: semantic differential scale section