

**FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO**

# **Interface Design for Human-guided Explainable AI**

**João Rafael Gomes Varela**



Master in Informatics and Computing Engineering

Supervisor: Gonalo Reis Figueira

Co-Supervisor: Fbio Neves Moreira

July 31, 2022

# **Interface Design for Human-guided Explainable AI**

**João Rafael Gomes Varela**

Master in Informatics and Computing Engineering

July 31, 2022

# Abstract

Current Artificial Intelligence (AI) systems are capable of making decisions or performing tasks independently, without human intervention. However, these systems can sometimes behave unpredictably, making them harmful, particularly when responsible for making decisions that can affect our lives. Therefore, it is crucial to understand how these systems construct such choices to detect these problems and guarantee that they behave as expected. Nevertheless, a substantial part of today's AI systems uses complex algorithms, which are seen as "black-box" systems that are hard to interpret and hence trust.

The present work was developed within the scope of the European project TRUST-AI, which seeks to take a step towards the next generation of AI, resorting to explainable-by-design symbolic learning models. These models are transparent and produce results that can be interpreted and manipulated by humans, allowing the incorporation of experts' domain knowledge and reinforcing the models' trustworthiness.

In this dissertation, the focus is precisely on this human interaction. We developed the user interface of a web platform that supports the TRUST-AI goals. The interface allows users to create projects, upload datasets, select an algorithm that fits the problem (classification, regression, and prescription), configure it, and train it. Finally, it allows the visualization and interaction with the generated results, promoting the system's explainability and human-guided learning.

The design process followed a user-centered design methodology to create a smooth human-AI interaction and user experience. We started by performing a domain analysis where we studied the available literature about the problem and similar existing tools. Then, we elicited the requirements of the interface while keeping end-users actively involved. After, we started the interaction design phase, which involved researching and choosing appropriate methods to present the information, create mock-ups, validate them with the project's stakeholders, and implement them.

The developed prototype was evaluated through continuous evaluation sessions, a workshop, and a usability study where we had participants interacting with the prototype and providing quantitative and qualitative feedback. In this study, we assessed the workload of the main tasks supported by the platform using the NASA Task Load Index (TLX) questionnaire and the overall usability of the interface using the System Usability Scale (SUS) questionnaire.

We conclude from the conducted study that the participants generally showed a high level of perceived satisfaction regarding the prototype. The group average for the overall SUS score was 82.1, which is considered an "Excellent" rating based on standard SUS guidelines. This is very promising for this platform and could inspire other works in the XAI field. The results also allowed us to identify the interface's positive aspects and point to weaknesses and possible enhancements, which helped define a plan for future work.

**Keywords:** Explainable Artificial Intelligence, Genetic Programming, Human-Centered Artificial Intelligence, Human-Computer Interaction, User Interface Design

# Resumo

Os sistemas de Inteligência Artificial (AI) estão a tornar-se capazes de tomar decisões ou realizar tarefas de forma independente, sem qualquer tipo de intervenção humana. No entanto, por vezes esses sistemas comportam-se de forma imprevisível, tornando-os prejudiciais, principalmente quando são responsáveis por tomar decisões que afetem as nossas vidas. Torna-se, portanto, bastante importante perceber como é que esses sistemas constroem tais decisões, de forma a poder detetar este tipo de problemas e garantir que têm o comportamento esperado. Todavia, grande parte dos sistemas de AI atuais usam algoritmos com uma grande complexidade inerente que faz com que sejam considerados modelos “black-box”, difíceis de interpretar e, portanto, difíceis de confiar.

O objetivo do presente trabalho, passa por desenhar e desenvolver a interface de um sistema XAI, desenvolvido no âmbito do projeto europeu TRUST-AI, que visa dar um passo em direção à próxima geração de AI, recorrendo ao uso de modelos de aprendizagem simbólica explicáveis por design. Estes modelos são transparentes e produzem resultados que podem ser interpretáveis e moldáveis por humanos, permitindo incorporar o conhecimento de especialistas e reforçando a confiabilidade nos modelos.

Nesta dissertação, o foco é precisamente essa interação humana e, para isso, foi desenvolvida a interface de utilizador de uma plataforma web que suporta os objetivos do TRUST-AI. A interface permite aos utilizadores criar projetos, fazer upload dos seus próprios datasets, escolher um algoritmo que se adeque ao problema (classificação, regressão e prescrição), configurá-lo e treiná-lo. Por último, permite a visualização e interação com os resultados, promovendo a explicabilidade do sistema e uma aprendizagem guiada por humanos.

O processo de design seguiu uma metodologia centrada ao utilizador, de forma a criar uma experiência de utilizador e interação humano-AI satisfatórias. Começou-se por realizar uma análise de domínio onde foi estudada a literatura disponível sobre o problema e ferramentas similares já existentes. Depois, foram recolhidos os requisitos da interface, mantendo sempre utilizadores envolvidos. Após, deu-se início à fase de desenho de interação, que envolveu a pesquisa e escolha de métodos para apresentar a informação, criação de mock-ups, validação dos mesmos e implementação.

O protótipo desenvolvido foi avaliado através de sessões de avaliação contínuas, um workshop e um estudo de usabilidade onde os participantes interagiram com o protótipo e forneceram feedback quantitativo e qualitativo. Neste estudo, avaliamos a carga de trabalho das principais tarefas suportadas pela plataforma, utilizando o questionário NASA Task Load Index (TLX), e a usabilidade geral do protótipo, utilizando o questionário System Usability Scale (SUS).

Do estudo conduzido conclui-se que, de forma geral, os participantes mostraram um alto nível de satisfação em relação à interface. A pontuação SUS média correspondeu a 82.1, o que é considerado “Excelente” conforme as diretrizes padrão do SUS. Isto é bastante promissor para a plataforma e pode inspirar outros avanços na área de XAI. Os resultados também permitiram identificar aspetos positivos da interface e apontar para defeitos e possíveis melhorias, que ajudaram a



definir um plano para trabalhos futuros.

**Keywords:** Inteligência Artificial Explicável, Programação Genética, Inteligência Artificial Centrada no Humano, Interação Humano Computador, Design de Interfaces do Utilizador

# Acknowledgements

This thesis marks the end of a five-year adventure where a lot of work was done. Maybe more than what was needed. But most importantly, this adventure allowed me to connect with so many wonderful people and experience things I never thought I would. And for that, I am truly grateful.

I want to start by thanking my supervisor, Prof. Gonçalo Figueira, and my co-supervisor, Prof. Fábio Moreira. Both did a fantastic job at guiding me through this long challenge and pushing me to go further. It was an absolute joy to work alongside them. I wish them the best of luck in the future and the TRUST-AI project.

I am also deeply grateful to Prof. Jácome Cunha. Even though he was not officially supervising my work, he showed immense support throughout the semester and was extremely helpful by sharing vital technical knowledge for my thesis.

I would like to acknowledge all the collaborators of the TRUST-AI project that I had the opportunity to work with. Moreover, I would like to express my gratitude to the ones that took some of their time to participate in the usability study.

A special thanks to my family, especially my dear mother, who always encouraged me to pursue my dreams, and above all, to be happy.

This endeavor would not have been possible without the support of my close friends, to whom I am deeply indebted. Thank you for all the memories, crying, and laughter. It has been a pleasure to share my life with you, and I can only hope it continues this way.

Finally, words cannot express my gratitude to some of the best people I will ever know. I could not have undertaken this journey without the support of Enzo, Inês, Miguel, Pedro and Tiago. Thank you for always being there for me. Thank you for believing in me. Thank you for making me feel loved. Thank you for being you! You are lovely people.

João Rafael Gomes Varela

*“Build a man a fire, and he’ll be warm for a day.  
Set a man on fire, and he’ll be warm for the rest of his life.”*

Terry Pratchett

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Context . . . . .	1
1.2	Motivation . . . . .	1
1.3	Goals . . . . .	2
1.3.1	TRUST-AI Project . . . . .	2
1.3.2	Interface Design . . . . .	3
1.3.3	Social Impact . . . . .	3
1.4	Summary . . . . .	4
1.5	Structure . . . . .	4
<b>2</b>	<b>Literature Review</b>	<b>5</b>
2.1	Explainable Artificial Intelligence . . . . .	5
2.1.1	Goals . . . . .	7
2.1.2	Artificial Intelligence Systems . . . . .	8
2.1.3	Ensuring Explainability . . . . .	9
2.1.4	Explainability Techniques . . . . .	10
2.1.5	Genetic Programming . . . . .	11
2.2	Interaction Design . . . . .	13
2.2.1	Usability . . . . .	14
2.2.2	User Experience . . . . .	16
2.2.3	User-centered Design . . . . .	18
2.2.4	Design Research . . . . .	20
2.2.5	Interaction Design Patterns . . . . .	22
2.2.6	Interaction Design for Explainable Artificial Intelligence . . . . .	24
2.3	Similar Platforms . . . . .	25
2.3.1	HeuristicLab . . . . .	26
2.3.2	Eureqa . . . . .	28
2.3.3	TuringBot . . . . .	29
2.3.4	Summary . . . . .	31
<b>3</b>	<b>Problem and Proposed Approach</b>	<b>34</b>
3.1	Problem Description . . . . .	34
3.2	Solution Approach and Methodology . . . . .	36
3.2.1	Domain Analysis . . . . .	36
3.2.2	Requirements Elicitation . . . . .	37
3.2.3	Interaction Design . . . . .	37
3.2.4	Evaluation . . . . .	37
3.3	Summary . . . . .	37

<b>4</b>	<b>Requirements Elicitation</b>	<b>39</b>
4.1	Methodology . . . . .	39
4.2	Non-functional Requirements . . . . .	39
4.3	Functional Requirements . . . . .	40
4.4	Use Cases . . . . .	41
4.5	Sitemap . . . . .	43
4.6	Summary . . . . .	45
<b>5</b>	<b>Interaction Design</b>	<b>46</b>
5.1	Methodology . . . . .	46
5.2	Screens and Features . . . . .	47
5.2.1	Authentication Page . . . . .	47
5.2.2	Projects Page . . . . .	47
5.2.3	Datasets Section . . . . .	48
5.2.4	Sessions Section . . . . .	49
5.2.5	Bookmarked Section . . . . .	49
5.2.6	Training Section . . . . .	49
5.2.7	Filter Section . . . . .	51
5.2.8	Evaluation Section . . . . .	52
5.3	Storyboards . . . . .	53
5.3.1	Sign In and Sign Up . . . . .	53
5.3.2	Project Creation . . . . .	54
5.3.3	Session Setup . . . . .	54
5.3.4	Training . . . . .	55
5.3.5	Filter Solutions . . . . .	55
5.3.6	Compare Sessions . . . . .	57
5.3.7	Highlight and Save Solution . . . . .	57
5.3.8	Evaluate Solution . . . . .	58
5.3.9	Rearrange Screen . . . . .	58
5.3.10	Edit Expression . . . . .	59
5.4	Architecture and Implementation . . . . .	64
5.4.1	Technologies . . . . .	66
5.4.2	Implementation Details . . . . .	67
5.5	Summary . . . . .	70
<b>6</b>	<b>Evaluation</b>	<b>71</b>
6.1	Continuous Evaluation . . . . .	71
6.2	Usability Study . . . . .	72
6.2.1	Procedure . . . . .	72
6.2.2	Pilot Sessions . . . . .	74
6.2.3	Main Sessions . . . . .	78
6.2.4	Threads to Validity . . . . .	89
6.2.5	Conclusions . . . . .	90
6.3	Workshop . . . . .	91
6.4	Summary . . . . .	92
<b>7</b>	<b>Conclusions and Future Work</b>	<b>93</b>
7.1	Conclusions . . . . .	93
7.2	Future Work . . . . .	94

<b>References</b>	<b>95</b>
<b>A Content Inventory</b>	<b>99</b>
<b>B Usability Study Information</b>	<b>103</b>
<b>C Usability Study - TRUST Features</b>	<b>105</b>
<b>D Usability Study Interviewer Script</b>	<b>109</b>
<b>E Usability Study Guide - Pilot Sessions</b>	<b>112</b>
<b>F Pilot Sessions - Questionnaire Answers</b>	<b>114</b>
<b>G Usability Study Guide - Main Sessions</b>	<b>117</b>
<b>H Main Sessions - Questionnaire Answers</b>	<b>120</b>
<b>I TRUST-AI - Workshop Presentation</b>	<b>125</b>
<b>J TRUST-AI - Workshop Exercise Guide</b>	<b>131</b>

# List of Figures

2.1	Explainable artificial intelligence system diagram. . . . .	6
2.2	Performance and explainability trade-off for different artificial intelligence techniques (Gunning and Aha, 2019). . . . .	7
2.3	Performance and explainability trade-off with XAI techniques (Gunning and Aha, 2019). . . . .	12
2.4	Performance and explainability trade-off with genetic programming. . . . .	13
2.5	Different disciplines of interaction design and user experience. . . . .	14
2.6	Different facets to the understanding of users' interaction with technology (Hasenzahl and Tractinsky, 2006) . . . . .	17
2.7	User-centered design process for interactive systems (ISO 9241-210:2019) . . . .	19
2.8	Heuristiclub UI - Enter Data Section . . . . .	26
2.9	Heuristiclub UI - Set Model Parameters Section . . . . .	26
2.10	Heuristiclub UI - Training Section . . . . .	27
2.11	Heuristiclub UI - Results Section . . . . .	27
2.12	Eureqa UI - Enter Data Section . . . . .	29
2.13	Eureqa UI - Set Model Parameters Section . . . . .	30
2.14	Eureqa UI - Training and Results Section . . . . .	31
2.15	TuringBot UI - Set Model Parameters, Train and Visualize Results Section . . . .	32
2.16	TuringBot UI - Visualize and Filter Logs Section . . . . .	32
3.1	Human-guided XAI Framework . . . . .	34
3.2	TRUST Platform Users . . . . .	35
3.3	Flow diagram of the proposed methodology for UCD . . . . .	36
4.1	TRUST - High-level Sitemap . . . . .	44
5.1	TRUST - Authentication Page (Sign In) . . . . .	47
5.2	TRUST - Authentication Page (Sign Up) . . . . .	48
5.3	TRUST - Projects Page . . . . .	48
5.4	TRUST - Datasets Section . . . . .	49
5.5	TRUST - Sessions Section . . . . .	50
5.6	TRUST - Bookmarked Section . . . . .	50
5.7	TRUST - Training Section . . . . .	51
5.8	TRUST - Filter Section . . . . .	52
5.9	TRUST - Evaluation Section . . . . .	53
5.10	Sign In & Sign Up storyboard . . . . .	54
5.11	Project Creation storyboard . . . . .	55
5.12	Session Setup storyboard - Dataset Uploading . . . . .	56
5.13	Session Setup storyboard - Session Creation . . . . .	56

5.14	Training storyboard . . . . .	57
5.15	Filter Solutions storyboard . . . . .	58
5.16	Compare Sessions storyboard . . . . .	59
5.17	Highlight & Save Solution storyboard . . . . .	60
5.18	Evaluate Solution storyboard . . . . .	61
5.19	Rearrange Screen storyboard - Add New Window . . . . .	61
5.20	Rearrange Screen storyboard - Drag & Close Window . . . . .	62
5.21	Edit Expression storyboard . . . . .	63
5.22	TRUST - Frontend High Level Architecture . . . . .	65
5.23	TRUST - Representation of a Genetic Programming Solution . . . . .	67
5.24	TRUST - Editor Operators & Functions List . . . . .	69
5.25	TRUST - Editor Error Message . . . . .	70
6.1	Pilot Sessions - Tasks NASA TLX Scores Comparisons . . . . .	76
6.2	Pilot Sessions - SUS Score . . . . .	77
6.3	Participants Characterization - Age Distribution . . . . .	79
6.4	Participants Characterization - Academic Background . . . . .	79
6.5	Participants Characterization - Academic Level . . . . .	80
6.6	Participants Characterization - Comfortability using Artificial Intelligence Algorithms and Techniques . . . . .	80
6.7	Participants Characterization - Comfortability using eXplainable Artificial Intelligence Algorithms and Techniques . . . . .	81
6.8	Participants Characterization - Comfortability using Genetic Programming Algorithms . . . . .	81
6.9	Participants Characterization - Months of Experience using GP Algorithms . . . . .	82
6.10	Interview Structure . . . . .	83
6.11	Main Session - Task 1 (Training Visualization) TLX Scores . . . . .	84
6.12	Main Session - Task 2 (Edit & Evaluate) TLX Scores . . . . .	85
6.13	Main Session - Task 3 (Do It All) TLX Scores . . . . .	85
6.14	Main Session - Tasks NASA TLX Scores Comparisons . . . . .	86



# List of Tables

2.1	Techniques to involve users in the design process (adapted from Sharp et al. (2007)).	20
4.1	TRUST - Key Functional Requirements . . . . .	40
4.2	TRUST - User Stories . . . . .	41
6.1	SUS Scores Interpretation . . . . .	75
6.2	Main Sessions - Tasks Average Time Comparison . . . . .	86
6.3	Main Sessions - Summary of SUS questionnaire results . . . . .	87
F.1	Pilot Sessions - Participants Characterization . . . . .	114
F.2	Pilot Sessions - Task 1 Nasa TLX Scores . . . . .	115
F.3	Pilot Sessions - Task 2 Nasa TLX Scores . . . . .	115
F.4	Pilot Sessions - Task 3 Nasa TLX Scores . . . . .	115
F.5	Pilot Sessions - Task 4 Nasa TLX Scores . . . . .	115
F.6	Pilot Sessions - SUS Results . . . . .	116
H.1	Main Sessions - Participants Characterization . . . . .	121
H.2	Main Sessions - Task 1 Nasa TLX Scores . . . . .	122
H.3	Main Sessions - Task 2 Nasa TLX Scores . . . . .	122
H.4	Main Sessions - Task 3 Nasa TLX Scores . . . . .	123
H.5	Main Sessions - SUS Results . . . . .	124

# Abbreviations

TRUST	Transparent, Reliable and Unbiased Smart Tool
AI	Artificial Intelligence
XAI	eXplainable Artificial Intelligence
GP	Genetic Programming
HCI	Human-Computer Interaction
UCD	User-centered Design
HCD	Human-centered Design
UI	User Interface
UX	User Experience
XUI	Explanation User Interface
SUS	System Usability Scale
TLX	Task Load Index
MWL	Mental Workload

# Chapter 1

## Introduction

This chapter introduces the context of this dissertation work and the problem we address in Section 1.1. We follow by explaining the motivation behind this work in Section 1.2 and its main goals in Section 1.3. The chapter is summarized in Section 1.4, and it finishes with a brief description of the structure of this document’s remainder in Section 1.5.

### 1.1 Context

The concept of Artificial Intelligence (AI) was born in the 1950s when Alan Turing hypothesized its ability to overcome human intelligence. However, with the technological capabilities at the time, such a thing was impossible (Kile, 2013). With recent advancements in technologies, a better theoretical understanding of AI, and the availability of large amounts of data, AI has become an integral part of society and our lives, impacting a variety of sectors such as marketing, healthcare, art, military, education, and so forth (Nadikattu, 2016).

AI systems have become so sophisticated that they can perform tasks independently, without human intervention, replacing many humans that would previously perform those tasks, creating new opportunities and, ultimately, reshaping our society. However, these systems are far from perfect and can behave unpredictably, making them harmful, particularly when responsible for making decisions that can affect our lives. This results in an emerging need to understand how these systems construct such choices to detect these problems and guarantee that they behave as expected (tax, 2020).

### 1.2 Motivation

Recently, we have witnessed the creation and utilization of systems based on algorithms or models with immense complexity, such as Deep Neural Networks, Random Forests, Support Vector Machines, etc. Although they have shown to be successful even when applied to complex problems, their inherent complexity results in a lack of transparency, making them “black box” models. Such

models can not explain why they have made a decision or prediction, making it hard to interpret them and, hence, trust (Schoonderwoerd et al., 2021).

There are scenarios where it is extremely important to hold someone or something accountable for a particular decision. In Doran et al. (2017), the authors formulate some of these scenarios:

“How can we hold accountable Artificial Intelligence systems that make decisions on possibly unethical grounds, e.g., when they predict a person’s weight and health by their social media images or the world region they are from as part of a downstream determination about their future, like when they will quit their job, commit a crime, or could be radicalized into terrorism” (Doran et al., 2017)

Many systems like these are out there without people even realizing it, but it is not necessarily sensible to trust them. Humans tend to be skeptical about depending on systems that are not directly interpretable, especially when responsible for making security, health, and economic-related decisions. For that reason, there has been an increasing demand for transparent and ethical AI (Goodman and Flaxman, 2016). This is why the field of eXplainable Artificial Intelligence (XAI) has emerged, which aims to make AI systems more transparent and trustworthy, allowing humans to question them and receive humanly understandable explanations (Hagras, 2018). There is a trade-off between a system’s performance and its transparency. However, an improvement in a system’s transparency can also contribute to better performance as it can help detect and reduce its deficiencies for three main reasons: it helps ensure impartial decision-making by detecting biases; it helps create a more robust system as it highlights potential adversarial perturbations (inputs that can “fool” a system; it ensures that only relevant variables contribute to infer the output guaranteeing a truthful causality (tax, 2020).

## 1.3 Goals

### 1.3.1 TRUST-AI Project

The present work aims to design and implement the user interface of the “Transparent, Reliable and Unbiased Smart Tool” (TRUST), developed within the TRUST-AI project, which is funded by the European Union’s Horizon 2020 research and innovation program, proposed and led by INESC TEC (Instituto de Engenharia de Sistemas e Computadores, Tecnologia e Ciência). TRUST is expected to be a web application that allows its users to create projects, upload datasets, select a model that fits the problem (classification, regression, and prescription), train and adjust it, and visualize the results graphically and textually. Although some applications like this exist, TRUST will focus itself on three main components:

- **Using explainable models to solve problems**, namely symbolic learning models generated by **genetic programming** techniques, which are transparent and produce results that are easily interpreted and manipulated by humans;

- **Generate human understandable results** that can give enough confidence for the user to trust the model or adjust it. This can be done by providing the users with explanations such as why a given solution was chosen and why another solution was not;
- **Promote human-machine interaction**, which is backed up by the previous two, by allowing its users to understand and interact with the models, ultimately resulting in a human-guided AI. This is the main focus of the dissertation.

TRUST is envisioned to be transparent, reliable, and prevent undesirable biases, making it a powerful tool with many applications and capable of disrupting multiple sectors, where human control is essential.

### 1.3.2 Interface Design

Although XAI models, such as symbolic learning models, are necessary to build a transparent system capable of generating explanations that show the rationale behind each decision, they are not everything that is needed. To be truly effective in gaining human trust, graphical interfaces play an essential role that can “make or break” the project’s success, as they are responsible for direct communication with the end-user (Liao et al., 2020).

Within the scope of this dissertation, we aim to build a user interface responsible for making the generated solutions approachable and easily understandable. The interface should allow humans to provide feedback and interact with the system, contributing to a human-guided AI. Even though XAI, as a field, has been growing rapidly, there is little to no research or shared practices on how to design user-friendly explainable AI applications (Liao et al., 2020). For these reasons, user interaction methods that can enhance the human-friendliness of AI models will be investigated. This includes narratives and visuals that can make solutions more easily understandable and the capacity to adapt the interface, and models on the fly, according to human evaluation.

The user interface should communicate and fetch information from a backend server that provides the necessary services. This backend component is being developed in parallel within the scope of another master’s dissertation, which is also part of the TRUST-AI project.

### 1.3.3 Social Impact

TRUST carries the mission of accelerating the convergence of human and machine by providing a tool that can solve various classification, regression, and prescriptive problems with a high level of explainability, guaranteeing that machines behave in agreement with the expectations of human experts in terms of factors such as ethics, bias or stability of the models. This will help establish a relationship of trust between humans and AI systems, making the latter a better fit for applications that can impact our lives.

## 1.4 Summary

The current dissertation provides meaningful contributions by providing a systematic literature review on topics such as Explainable Artificial Intelligence and Interaction Design, with the goal of uncovering the most appropriate and generalizable design methods to develop a user-centered XAI system.

The knowledge gathered with the literature review will be applied to develop the TRUST tool: a transparent XAI system that can bridge the gap between humans and machines, allowing them to collaborate in discovering new and explainable solutions for regression, classification, and prescriptive problems.

## 1.5 Structure

This document is composed of seven more chapters. Chapter 1 contextualizes the work developed and provides a rationale for it, its goals, and the methodology followed during the dissertation.

Chapter 2 covers the fundamentals and analysis of relevant topics within the thesis' scope, such as Explainable AI and Interaction Design. It also includes the analysis of similar existing tools.

Chapter 3 is where the problem is analyzed, as well as the possible difficulties and challenges. It follows with a proposed approach, methodology, and plan to implement the solution.

Chapter 4 is where the methodology for gathering requirements is explained and where the functional and non-functional requirements are described. Then, the use-cases and sitemap to satisfy these requirements are detailed.

Chapter 5 starts by describing the strategy followed for the Interaction Design phase. It follows by presenting the developed results, its main capabilities, and architecture.

Chapter 6 explains the process of evaluating the developed prototype. It describes the continuous evaluation that was performed during the design process and, finally, details the usability study that was carried out after the Interaction Design phase.

Finally, Chapter 7 details and discusses the contributions, results, and challenges of the dissertation, as well as some prospects for future work on the topic.

## Chapter 2

# Literature Review

This chapter provides a deep-dive analysis of topics essential to understanding the context of the dissertation and successfully achieving the proposed goals. Firstly, in Section 2.1, a review of the current knowledge about Explainable Artificial Intelligence. Then, Section 2.2 reviews the topic of Interaction Design. Finally, Section 2.3 presents an analysis and evaluation of similar platforms so that we can learn from their strengths and work on reinforcing their weaknesses.

### 2.1 Explainable Artificial Intelligence

Artificial intelligence (AI) holds the potential for improving both private and public life. A key component of data science is the automated discovery of patterns and structures in massive amounts of data. Currently, it drives applications in diverse areas such as medicine, law, and finance. More present than ever in our daily life, AI systems have grown to become so sophisticated that some stopped requiring any human intervention in their design, making them capable of performing decisions independently (tax, 2020).

Although AI systems' performances keep improving, they have become so complex that most of the time, we simply can not understand what the rationale behind their decision-making process is. This became especially true with the increase in popularity and employment of "black-box" machine learning models, which are complex systems whose inner workings are extremely hard or even impossible for humans to understand (Doran et al., 2017). This lack of transparency becomes especially important when these systems are responsible for making decisions in sensitive areas that can impact work opportunities, prison sentences, and even our health. In this context, it becomes hard to trust AI systems and much more to hold them accountable.

This raises the question and challenge: how can we understand and trust the decisions these systems suggest. To achieve a high level of trustworthiness and evaluation of a machine's ethical and moral standards, explanations should be provided as an output of any AI system (Belle and Papantonis, 2021). Such explanations should provide human-understandable insights into

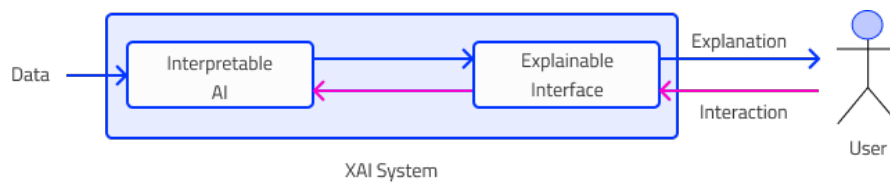


Figure 2.1: Explainable artificial intelligence system diagram.

the rationale that an AI uses to draw conclusions, giving stakeholders an alternative to “blindly” accepting the system’s decisions.

This is where the field of *eXplainable Artificial Intelligence* (XAI) comes into play. Although there is some lack of consensus when it comes to defining explainability, XAI is generally defined as an intersection between AI, social science, and human-computer interaction (HCI) (Vilone and Longo, 2020), whose goal is to “create a suite of machine learning techniques that enables human users to understand, appropriately trust, and effectively manage the emerging generation of artificially intelligent partners” (Gunning and Aha, 2019).

The concept of explainability is frequently replaced with the notion of interpretability, understandability, and even transparency, though there are subtle differences between these concepts as noted in (tax, 2020):

- **Understandability** - capacity to make the model understandable to end-users. That is, the characteristics of a model to make a human understand how it works;
- **Interpretability** - ability to provide or bring out the meaning of something in understandable terms to a human;
- **Explainability** - the extent to which a person can understand and comprehend the reasoning behind a decision made by a model, usually associated with the notion of explanations as an interface between humans and a decisions maker;
- **Transparency** - the ability for a model to be humanly understandable by itself.

As illustrated in Figure 2.1 an XAI system should be able to generate explanations and describe the reasoning behind its decisions and predictions. The user interacts with the explainable interface to send questions to the interpretable model and receive explanations for the predictions. The interpretable model works with the data to come up with explanations or predictions for the user’s query.

As stated, explainability is one of the main barriers for AI to be applied in sensitive areas. In general, humans are reticent to adopt systems that they can’t understand and hence trust (Zhu et al., 2018). This, together with the increasing demand for ethical and responsible AI (Goodman and Flaxman, 2016), which stands for the “right to explanation” and vigorous efforts to prevent unethical biased models, has led to a recent uprising of the XAI field.



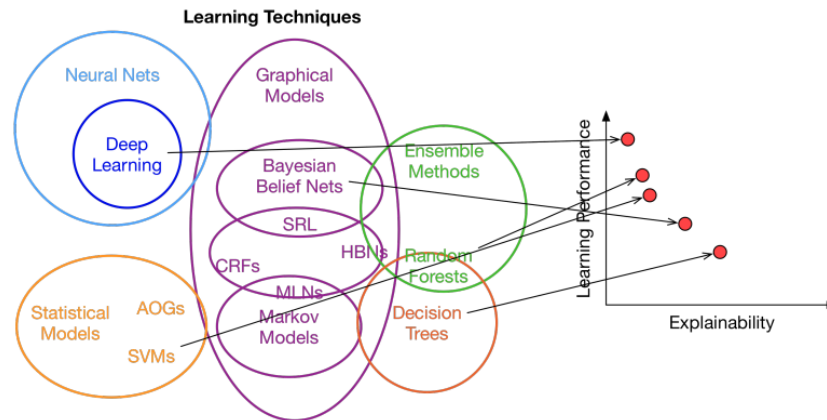


Figure 2.2: Performance and explainability trade-off for different artificial intelligence techniques (Gunning and Aha, 2019).

Although there is usually a trade-off between the model's performance and its transparency, as seen in Figure 2.2, meaning that more opaque models (such as black-box models) result in better performance, that is not always the case. An interpretable system can lead its designer to improve its results for three different reasons (Došilović et al., 2018):

- interpretability helps detect and consequently correct bias;
- interpretability leads to more robust systems as it facilitates the detection of adversarial perturbations;
- interpretability helps ensure that the model reasoning is based on true causation on truthful causality by guaranteeing that only meaningful variables are used to infer the output.

In short, although such a trade-off exists, it can almost be compensated by the advantages that the interpretability of a system brings to the table. Current XAI techniques also aim to create models that are more understandable while still being high-performing.

### 2.1.1 Goals

There is no mathematical definition of XAI's goals, and that is also a topic where there is not complete agreement within the XAI community. However, we can say that based on literature (tax, 2020; Vilone and Longo, 2020; Doran et al., 2017), there are some generally agreed goals for the field. These goals are triggered mainly by user necessities and are typically directed at supporting transparency and improving the user's decision-making capacity by providing decision-oriented explanations with factual information to support it (Wang et al., 2019):

- **Trustworthiness** - although hard to quantify, inducing trust in the end-user is one of the primary goals of XAI. Trustworthiness might be defined as the confidence that the system will act as expected when facing a given problem;

- **Causality** - showcasing the causal links between the involved variables can be essential so that users can verify which variables are being given more importance when deducing the output and if they are, for instance, resulting in biases;
- **Transferability** - understanding the inner works of a model eases the task of clarifying the boundaries that could have an impact on a model, allowing users to reuse the same knowledge for different problems and even improve it;
- **Confidence** - there should be information about how robust and reliable a system is. An explainable model should include information regarding the working regime's confidence;
- **Fairness** - from a social standpoint, explainability can help guarantee fairness in AI systems since it allows users to visualize the relations affecting the results, highlighting biases that could cause the system to be unfair or unethical;
- **Accessibility** - Explainability makes AI easier to understand, enabling users to participate more actively in the development of AI models. Depending on the level of explainability, it can become possible for non-technical or non-expert users to deal with those models;
- **Interactivity** - users should be able to interact with their models by tweaking and manipulating them so as to ensure a successful model that works as expected.

## 2.1.2 Artificial Intelligence Systems

When it comes to explainability, we can consider a spectrum where on one side, we have a fully opaque system, and on the other one, a fully transparent one.

### 2.1.2.1 Opaque Systems

Opaque (“black-box”) systems are analogous to an oracle that makes predictions based on input but does not explain how or why those predictions were generated (Doran et al., 2017). In systems like these, the mechanisms responsible for mapping inputs to outputs are way too complex for a human to understand. In other words, these systems lack transparency and, therefore, aren't understandable by themselves. However, there are *post-hoc* explanations that can be applied to those models to make them more understandable.

Common *post-hoc* explanations include model approximation using local simpler models, visualized as saliency maps or in the form of decision trees. These methods produce explanations that appear to be similar to those produced by transparency-based methods. It is critical to properly indicate to consumers or ethicists that they are *post-hoc* approximations if they are supplied (Abrás et al., 2004).

Although we have seen an increase in usage of these types of techniques, they are not that successful in making AI systems trustworthy, as the generated simplified models lose a lot of the original complexity (making them fundamentally different) and do not allow users to understand how to improve the original model.

### 2.1.2.2 Transparent Systems

Transparent systems are systems where a user can see and investigate and comprehend how inputs are translated into outputs. These systems can be achieved by combining three different dimensions (Belle and Papantonis, 2021; Vilone and Longo, 2020; tax, 2020):

- **Simulatability** - a model's ability to be simulated or reasoned by a human. This category, of course, only includes models that are simple and compact. A user must be able to fully comprehend the structure and function of a model before it can be simulated;
- **Decomposability** - the ability to break down a model into parts (input, parameters, and calculations) and explain each one. This characteristic empowers the ability to understand, interpret and explain a model's behavior;
- **Algorithmic Transparency** - the ability to comprehend the model's procedure for generating its results. In general, the sole criteria for a model to belong in this category is that it can be inspected through mathematical analysis by the user.

Depending on their complexity, a transparent system can be auto explicable and easy to inspect, interpret, trust, validate, and hence trust.

### 2.1.3 Ensuring Explainability

There have been attempts to create broad procedural standards for implementing and explaining AI systems. Among them, it was suggested in Leslie (2019) that explainability should be incorporated and considered taking into account four important dimensions of practical AI design:

- **Contextual factors** - when developing an approach to interpretability, it is important to examine the potential consequences as well as domain-specific requirements. This includes a deep understanding of the goal of the AI in question, the level of detail of the explanations required by its intended audience, the performance, and the existing technology that can be used;
- **Interpretable techniques** - transparent or explainable by design models should be preferred when possible, as they lead to a much more natural and higher level of explainability. It is important to thoroughly study the problem at hand to choose the appropriate models capable of meeting the system's requirements. It is recommended to think about explainability before the performance, which endorses the consideration of standard interpretable models before thinking about more complex opaque models;
- **Responsible design** - ethics, fairness, and safety implications should all be considered while building AI systems. To this end, it is proposed that a full articulation and evaluation of the relevant explanatory strategies, as well as an examination of their coverage, is carried out. It is also important to verify that it meets the requirements, and, finally, the definition of an interpretability action plan that pushes a high-standard explanation delivery strategy;

- **Human input** - when designing explainable systems, it is encouraged to rethink interpretability taking into account the cognitive skills, capacities, and limitations of humans. After all, they are the ones explainability aims to aid. To this end, the expertise of the audience should be involved in all the system's design phases.

Although not rigid, following the above guidelines will contribute to methodologically building explainable and responsible AI (tax, 2020).

On a deeper level, when it comes to designing the actual explanations, the consideration of the following explanation requirements is suggested (Vilone and Longo, 2020; Adadi and Berrada, 2018):

- **explain to justify** - the decisions made by the model should be explained to boost the model's justifiability;
- **explain to control** - explanations should improve a model's transparency, allowing it to be debugged, manipulated, and have its defects identified;
- **explain to improve** - explanations should help users improve the performance of the model by guiding them to decisions that help do so;
- **explain to discover** - extraction of new knowledge, such as correlations and patterns in data, should be supported with explanations.

In general, explanations should build confidence and trust in the model's accuracy, denote what is wrong, and guide the user in the right direction (Doran et al., 2017).

#### 2.1.4 Explainability Techniques

We can consider two main categories for interpretability techniques in AI (Abrás et al., 2004):

- **Transparency-based** - constructing self-explanatory models, such as decision trees, symbolic expressions, ruled-based or linear models, that can be easily understandable by themselves;
- **Post-hoc** - creating secondary interpretable models that aid the interpretation of the original black-box model.

For each of these techniques, various types of explanations to better understand models have been explored, and on a general level, they all fall into one of the following categories (Doran et al., 2017):

- **Text explanations** - using symbols, such as natural language text, to create explainable representations. Other examples include propositional symbols, which define abstract ideas that capture high-level processes and explain the model's behavior;

- **Visual explanation** - attempt to create visuals that aid in the understanding of a model. Despite certain inherent obstacles (such as our incapacity to grasp more than three dimensions), the proposed methodologies can aid in the discovery of information about the decision boundary or how features interact with one another;
- **Local explanations** - attempt to describe how a model works in a certain area of interest. As a result, the ensuing explanations may or may not extend to a global scale, accurately describing the model's overall behavior. Instead, they usually approximate the model around the instance the user wishes to explain in order to derive explanations detailing how it works when confronted with similar situations;
- **Explanations by example** - selecting exemplary examples from the training dataset to understand how the model works. In many circumstances, this is comparable to how humans approach explanations, using unique examples to convey a more general process;
- **Explanations by simplification** - refers to approaches for approximating an opaque model with a simpler one that is easier to interpret. The fundamental difficulty arises from the fact that the basic model must be adaptable enough to accurately represent the complicated model;
- **Feature relevance** - by measuring the influence of each input variable, these explanations seek to explain a model's decision. As a result, relevance ratings are ranked, with higher scores indicating that the associated variable was more important for the model. These scores may not always provide a thorough explanation, but they do provide a starting point for learning more about the model's logic.

These explanations are extremely important and are usually combined to support each other. When choosing which types of explanations to implement, it is essential to consider the users that these explanations are made for, as some explanations might be more appropriate than others depending on the context.

The application of explainability techniques can help any model become more explainable and possibly increase its performance if explanations give enough information for the user to improve the model's performance (Figure 2.3). Nevertheless, transparency-based approaches are more ambitious in their goal of obtaining self-explanatory models. Genetic Programming is one of the techniques used to generate those models.

### 2.1.5 Genetic Programming

Genetic programming (GP) is a type of evolutionary algorithm used to generate and evolve symbolic expressions. It starts by generating an initial population (usually random), and it evolves that population by applying operations analogous to natural genetic processes such as crossover, mutation, and reproduction. Then, it selects individuals for the new population based on the current one and the offspring population. It evaluates that new population and the process is repeated until

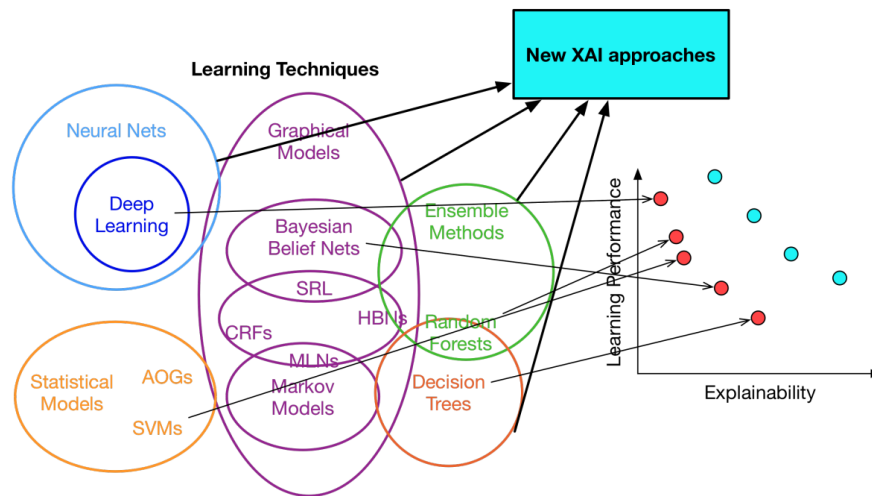


Figure 2.3: Performance and explainability trade-off with XAI techniques (Gunning and Aha, 2019).

a minimum fitness is achieved or reaches the maximum computing or time budget (Banzhaf et al., 1998).

To define a population, it is necessary to define a terminal set: inputs of programs and constants (e.g.,  $x$ ,  $y$ , 2) that will correspond to the leaf nodes of the tree; and a function set: operators that are applied to the inputs (e.g.,  $+$ ,  $-$ ,  $\max$ ) and form the non-leaf nodes. These terminals and operators are defined before the program is executed and can heavily influence the possible outcome expressions (Goldberg, 1989).

GP models can be applied to regression, classification, and prescriptive tasks and can be easily represented as trees or graphs, making them an ideal candidate for a fully transparent model. However, that is not always the case (e.g., a very lengthy and complex mathematical expression might not be interpretable at all). To improve the interpretability of GP methods, one should keep in mind the following dimensions (Du et al., 2019):

- Model size (e.g., number of nodes);
- Number of features used in the model;
- Model complexity (e.g., non-linear operators are more complex);
- Physical meaning of the expression.

And may use the following techniques:

- Constrained GP (penalise less interpretable models);
- Multi-objective GP (accuracy vs interpretability measures);

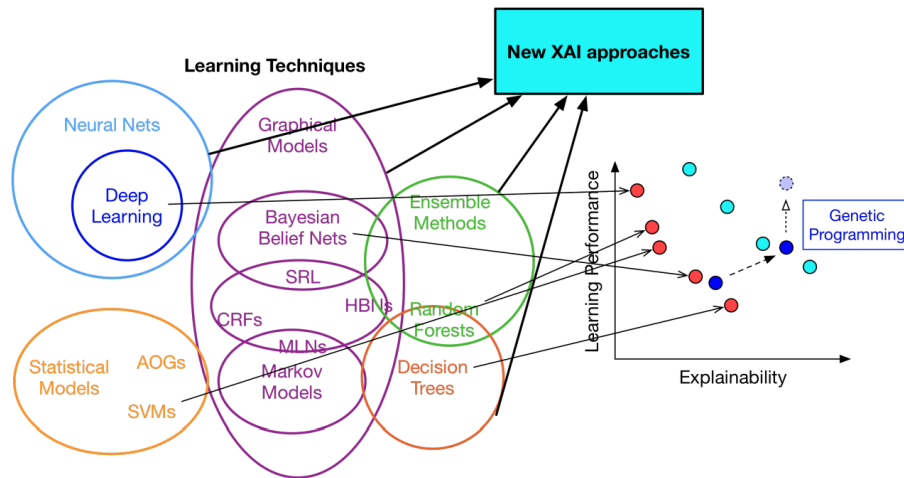


Figure 2.4: Performance and explainability trade-off with genetic programming.

- Simplification (e.g., tree pruning);
- Different GP representations (e.g., strongly-typed, grammar-guided, ensemble/multi-tree);
- Intuitive Visualisation.

Although time-consuming and computationally expensive, genetic algorithms are a proven solution that can solve various problems with satisfactory performances (Chen et al., 2017; Masood et al., 2021; Padillo et al., 2019), and can be the answer to improving performance while maintaining a high level of explainability (Figure 2.4). If the mentioned techniques are applied correctly, GP models can be quickly inspected and interpretable by humans, potentially making them an essential component of the future of XAI.

## 2.2 Interaction Design

In the last couple of decades, the number of computational systems has proliferated, and with them, the number of visual interfaces that are required. Often, the first point of contact of a product or a service is an interface that can be decisive for the user to be satisfied. The variety of users of the systems has also increased over time: there are users with different backgrounds, different types of knowledge, different capacities, different limitations, and so on (Blair-Early and Zender, 2008).

The necessity for effective and adaptive interface design grows as the number of interfaces, and the diversity of users expands. Interaction design is referred to as the practice of designing interactive digital products and services. John Kolko, the author of “Thoughts on Interaction Design” (Kolko, 2011), gives the following definition to interaction design:

“Interaction Design is the creation of a dialogue between a person and a product, system, or service. This dialogue is both physical and emotional and is manifested



in the interplay between form, function, and technology as experienced over time.”  
(Kolko, 2011)

Interaction design focuses on facilitating how users interact with products and aims to create a product that matches the desired user experiences and expectations.

The terms “interaction design” and “user experience (UX) design” are sometimes used interchangeably. Although the concepts are not synonymous, this is understandable given the significant overlap between both, as it can be seen in Figure 2.5. After all, UX design is all about shaping the user’s experience with a product or service, and interaction between the user and the product is a significant part of that.

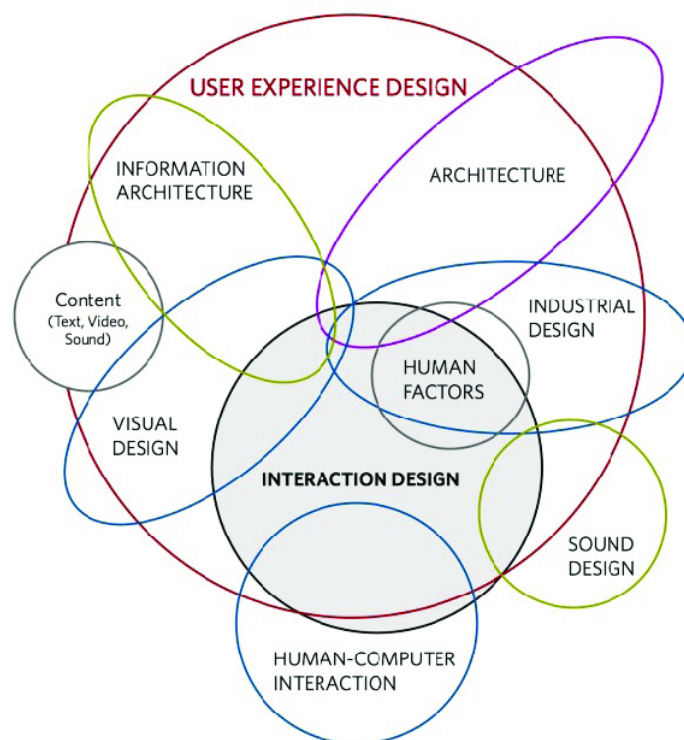


Figure 2.5: Different disciplines of interaction design and user experience.

### 2.2.1 Usability

Usability is a concept that encompasses both the user and the system. The term is related to how user-friendly the interactions with the system are for performing specific tasks. Usability is not a one-dimensional property of the interface of a system. Instead, it has multiple components and is typically associated with five usability attributes (Jordan, 2020):

- **Learnability** - the system should be simple to understand so that the user may become productive quickly;



- **Efficiency** - when a user learns how to utilize the system, that learning process should be efficient enough for the user to reach high levels of productivity;
- **Memorability** - the system should be simple to remember so that a casual user can return to it after a period of time without having to relearn how to use it;
- **Errors** - the system should have a low error rate and be tolerant to failures so that the users can smoothly use the system and recover from errors;
- **Satisfaction** - the system should be enjoyable to use so that users are satisfied with their experience.

This definition of usability derives from three different views of what usability is (Bevana et al., 1991):

- **Product-oriented view** - ergonomic attributes of the product;
- **User-oriented view** - mental effort and attitude of the user;
- **User performance view** - user interactions with the product, with a focus on ease-of-use (how easy the product is to use) and acceptability (is the product ready to be used in a real scenario).

Based on this definition, Jakob Nielsen proposes ten general principles for usability in interface design (Nielsen, 2005):

- **Visibility of system status** - the system should always keep users up to date on what's going on by providing suitable feedback in a timely manner;
- **Match between system and the real world** - the system should use human-understandable language, using concepts that are known to them, rather than system-oriented jargon. Follow real-world norms to present data in a logical and natural sequence;
- **User control and freedom** - users frequently select system functions by accident, necessitating the presence of a clearly defined "emergency escape" that allows them to quit the unwanted state without having to go through a lengthy process;
- **Consistency and standards** - users should not have to guess what different words, situations, or actions signify. Stick to the platform's conventions;
- **Error prevention** - a smart design that prevents a problem from arising is even better than effective error messages. Either eliminate error-prone conditions or check for them and provide users the chance to confirm their actions before proceeding;
- **Recognition rather than recall** - minimize the user's memory effort by making relevant options accessible and easily observable. The user should not have to recall information from one dialogue segment to the next;

- **Flexibility and efficiency of use** - accelerators can often speed up the interaction for the expert user, allowing the system to serve both inexperienced and experienced users. Allow users to customize their routine actions;
- **Aesthetic and minimalist design** - information that is irrelevant or only occasionally required should not be included in dialogues. In a discourse, each additional unit of information competes with the relevant information units, lowering their relative prominence;
- **Help users recognize, diagnose, and recover from errors** - Error messages should be written in plain language (not just using codes), clearly state the problem, and offer a constructive resolution;
- **Help and documentation** - even while it is preferable if users can operate the system without documentation, assistance, and documentation may be required. Such material should be searchable, focused on the user's task, include a list of concrete steps to follow, and not be too extensive.

Usability is typically assessed using a variety of observable and quantifiable criteria that eliminate the need for intuitive judgment. Because there are so many quantitative data sources from which to measure, it is easy to get lost in a black hole of interesting data but no meaningful knowledge. Generally, one should aim to measure the different dimensions of usability (learnability, efficiency, memorability, errors, and satisfaction) and metrics such as workload, completion rate, the number of errors, efficiency, and task-level satisfaction. This can be done by doing usability tests where users perform specific tasks (ISO/IEC 25010:2011).

### 2.2.2 User Experience

User experience is related to the user's emotional response and experiences when interacting with a system. It is a more subjective concept when compared with usability, being influenced not only by the system but also by the time, place, and emotional state of the user (Sharp et al., 2007). The concepts of user experience and usability, although distinct, are heavily influenced by each other: usability greatly influences the user experience, and some aspects of the user experience, such as appearance and responsiveness, contribute to the usability of the system. Both concepts help identify the primary goals of the system being designed.

As seen in Figure 2.6, we can consider three different prominent perspectives that contribute to the various facets of UX (Hassenzahl and Tractinsky, 2006):

- **Beyond the instrumental** - this perspective focuses on the importance of non-instrumental needs such as surprise, diversion, or intimacy, and how they can be approached with technology. It makes a clear connection between product attributes and customer needs and values. The novelty of a product and the challenges it presents, for example, contribute to its hedonic character, which is important because it promises to satisfy an underlying human need – a drive to be stimulated, to improve one's skills and knowledge, and to grow;

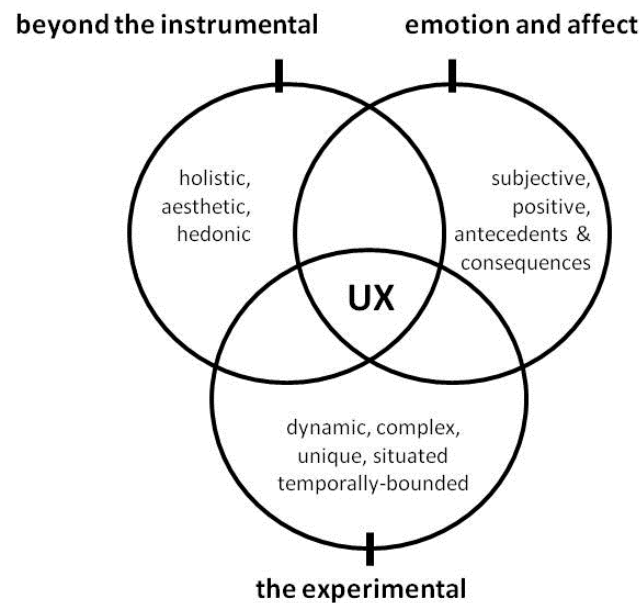


Figure 2.6: Different facets to the understanding of users' interaction with technology (Hassenzahl and Tractinsky, 2006)

- **Emotion and affect** - this view deals predominantly with questions such as how computers can sense user emotion, adapt to it, or even express their own affective response. Here, UX is concerned in understanding the function of affect as an antecedent, a consequence, and a mediator of technology use from a "human perspective." Furthermore, it focuses on good emotions such as joy, fun, and pride rather than negative emotions;
- **The experiential** - two characteristics of technology use are highlighted in this viewpoint: its situatedness and its temporality. From this perspective, an experience is a combination of factors, such as the product and the user's states (e.g., mood, expectations, goals, etc.) that spans time and has a distinct beginning and conclusion. All of these factors are assumed to be interconnected - they interact and modify each other. The actual experience is the result of this interaction. In contrast to material outcomes, experiential outcomes have a more positive impact on one's well-being.

In conclusion, UX is more than just instrumental needs, acknowledging its use as a subjective, contextual, complex, and a dynamic encounter. UX is the result of a user's internal state (e.g., predispositions, expectations, needs, motivation, and mood), the traits of the system (e.g., complexity, purpose, usability, and functionality), and the context (or environment) in which the interaction takes place (e.g., organizational and social setting) (Hassenzahl and Tractinsky, 2006).

### 2.2.3 User-centered Design

User-Centered Design (UCD) is a multidisciplinary design approach that involves active user participation to increase understanding of user and task requirements, as well as design iteration and evaluation. It is commonly regarded as the key to product usability and utility, as well as an effective way to get over the restrictions of traditional system-centered design (Mao et al., 2005).

UCD is a general term for design processes in which end-users have a say in how a design is created. It encompasses both a broad concept and a wide range of techniques. Users can be involved in UCD in a variety of ways, but the crucial principle is that they are involved in some way (Abbras et al., 2004).

Several authors propose a list of design principles that leads to a user-centered design, and they are mostly in accordance with the six principles defined by the ISO 9241-210 standard (ISO 9241-210:2019):

- **The Design is based upon an explicit understanding of users, tasks, and environments** - design analysis should focus on people and not processes. Final results can be improved by understanding what people think and how they feel when completing a task. Emotion drives behavior and attitudes;
- **Users are involved throughout design and development** - early input is necessary, but it is not enough. User involvement should continue throughout the whole design and implementation process;
- **The Design is driven and refined by user-centered evaluation** - the stakeholders of a system should be the ones who review and test the software. The standard should be that the user can easily complete tasks and feel a sense of accomplishment;
- **The process is iterative** - prototyping should be continuous and iterative with constant input from the user. Testing new processes and configurations begin a cycle of innovation as users discover new possibilities. Expect and welcome changes as people work through the process;
- **The Design addresses the whole user experience** - the user experience begins at the moment each becomes aware of an impending change. Change management and communication should start as soon as the system starts being thought out;
- **The design team includes multidisciplinary skills and perspectives** - most human capital technology decisions are made by a tiny group of human resources and information technologies people. Better results can be attained by engaging every discipline in the organization from the very start of the project.

UCD is described as an iterative process with the user at its core, where the user has an active role in the testing of the various iterations of the system and its design phases. The ISO 9241-210

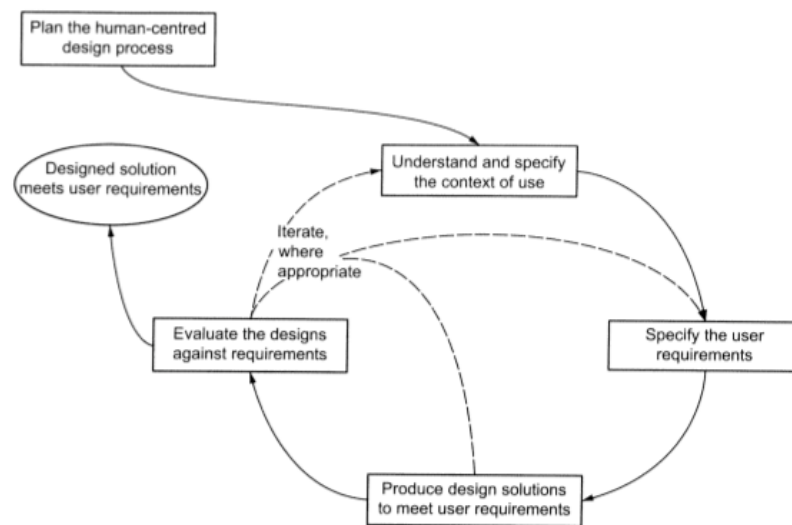


Figure 2.7: User-centered design process for interactive systems (ISO 9241-210:2019)

standard (ISO 9241-210:2019) also proposes the definition of four different phases that support the user-centered design process (Figure 2.7):

- **Understanding and specifying the context of use** - the main goal of this phase is to establish why the end-users would be interested in the product being designed and how they want to use it. It's crucial to remember that consumers use products to achieve specific goals because they regard them as a solution to their problems. By clearly formulating a problem, we increase the chances of creating a better use case for the product;
- **Specifying the user requirements** - identify any business or user goals that must be satisfied in order for the product to succeed. This should be done by actively involving the end-users to understand their needs and gather their feedback;
- **Producing design solutions** - discover and implement methods to present the information and functionalities of the system. This involves activities such as creating mock-up interfaces, validating them with the stakeholders, and, finally, implementing them;
- **Evaluating the design** - this phase serves as a way to gather formal feedback from the end-users of a system (i.e., through user tests, questionnaires, etc.), providing the designers with qualitative and quantitative metrics that help identify weaknesses and strengths of the system. This feedback can then be used to iterate the design of the platform and even the requirements, contributing to a high level of user acceptance.

In general, the proposed user-centered design activities focus on maintaining the user involved in each step of a system's development and have the necessary flexibility to iterate according to user feedback. On a deeper level, the activities in Table 2.1 are suggested to achieve that (Sharp et al., 2007)

Table 2.1: Techniques to involve users in the design process (adapted from Sharp et al. (2007)).

Technique	Purpose	Stage of the Design
Background In- terviews and questionnaires	Collecting data related to the needs and expectations of users; evaluation of design alternatives, prototypes, and the final artifact	At the beginning of the design project
Sequence of work interviews and questionnaires	Collecting data related to the sequence of work to be performed with the artifact	Early in the Design cycle
Focus groups	Include a wide range of stakeholders to discuss issues and requirements	Early in the Design cycle
On-site observation	Collecting information concerning the environment in which the artifact will be used	Early in the Design cycle
Role-Playing, walk- throughs, and simu- lations	Evaluation of alternative designs and gaining additional information about user needs and expectations	Early and mid-point in the Design cycle
Usability testing	Collecting quantitative data related to measurable usability criteria	Final stage of the design cycle
Interviews and questionnaires	Collecting qualitative data related to user satisfaction with the artifact	Final stage of the design cycle

## 2.2.4 Design Research

Design research is a broad term for the process that designers use to better understand the underlying and sometimes hidden desires, needs, and challenges of end users, also known as the target audience. Anthropology, scientific and sociological research, theater, and design itself, among other disciplines, heavily influence design research. The methods can range from pure observation to engaging with subjects in active play, such as role-playing (Saffer, 2009).

Designers employ these research approaches to learn more about their subjects and their surroundings that they might not have known otherwise, allowing them to better design for those subjects and environments. It aids designers in comprehending the product's or service's emotional, cultural, and aesthetic context (Saffer, 2009).

Many designers don't usually practice design research and choose to trust their instincts, knowledge, and experience to create products. Although this might work on small projects, the approach can be extremely risky for larger projects in unfamiliar domains, cultures, or subject areas. Without any upfront research, the stakeholders of a product risk only finding out defects later in the development process, such as in the testing phase, or worse, after launch (O'Grady and O'Grady, 2017).

Design research helps designers avoid inappropriate choices that could frustrate, embarrass, confuse, or make a situation unpleasant for users. Design research can also lead to moments of inspiration, when a research subject provides insights or the environment reveals how a product may fit into it. Meeting just one user will almost certainly alter one's perspective on a project (Saffer, 2009).

### 2.2.4.1 Design Research Principles

Based on anthropology rules proposed by Rick E. Robinson, Dan Saffer outlines three main rules for design research (Saffer, 2009):

- **You go to them** - designers should consider as an essential component of research, observing the environment and context where their subjects perform their activities. This involves meeting the actual end-users and testing the product on their environment, instead of making the subjects come to the designers and make them test in an unfamiliar and artificial location that might not be well representative of the real one;
- **You talk to them** - designers should do more than just read about their fields. They should also not try to understand their subjects by inquiring other people about them. Instead, designers should concentrate on hearing the project's subjects speak about their own experiences from their own perspectives. The intricacies of how a story is delivered can frequently reveal as much about the designer as the story itself;
- **You write stuff down** - it is fundamental that designers properly document what they see and hear as they research since human memory is faulty and can trick the designer's mind later.

It should be noted that it is also imperative that design research is done while treating the subjects ethically. Not only is it the morally correct thing to do, but it will also lead to better results, as the subjects are more likely to open up and provide relevant feedback if they feel they are being treated well and respectfully. This treatment includes guidelines such as:

- **Getting consent from subjects** - the subjects should be made aware that a research study is being conducted and of its purpose;
- **Explaining the risks and benefits of the study** - the subjects should be made aware of any risk associated with the study. They should also know the benefits and goals of the study;
- **Respect the subjects' privacy** - users' private data and identity should be respected and protected;
- **Providing data and research results to subjects** - when asked, the subjects should have the possibility to see what was recorded and the outcomes of the research.

### 2.2.4.2 Design Research Methods

Design research studies can be performed at any of the development phases of a product, from the product's definition to its evaluation. Design research has many methods, drawn from other disciplines or created by designers and researchers over the years. Commonly these methods can be split into five different categories:

- **Primary** - it aims to understand who the system is being designed for and involves going directly to the end-users to ask questions and gather data. It allows the designer to validate ideas with the users and design more meaningful solutions for them. This research is typically done through interviews with individuals or small groups, surveys, and questionnaires;
- **Secondary** - it involves using existing data such as books or articles to support design choices and better understand the context behind the design. It can also help validate user insights extracted from primary research and provide additional support;
- **Generative** - also known as exploratory research, focuses on gaining a deep understanding and familiarity of the topic at hand and using the insights gained to define the problem to be solved and create solutions for it;
- **Evaluative** - focuses on assessing the quality of the solution, allowing the users to provide essential feedback to evaluate the product. This method aims to collect that feedback and use it to refine and improve the design experience. Usability studies are a perfect example of this research method.

When considering the metrics extracted from the studies, design research methods can be split into two different dimensions (Laurel, 2003):

- **Quantitative research** - aims to capture and measure certain aspects of users' behavior through numerical data, making it suitable for statistical analysis. It answers questions such as "how many people clicked here?", "what percentage of users can find the call to action?" or "how fast are users to find something or perform an action?". It's useful for deducing statistical probabilities and figuring out what's happening with a product;
- **Qualitative research** - aims to understand users based on experiences and impressions. It answers questions like "why didn't people see the call to action?" and "what else did people notice on the page?" and often takes the form of interviews or conversations. Qualitative research helps in the understanding of why people act the way they do.

Although design research is a powerful tool when it comes to interaction design, it is almost useless without a proper analysis and the making of structured findings from the research data that can be used to build a system that better fits the end-user necessities.

### 2.2.5 Interaction Design Patterns

Despite the fact that each person is unique, people behave in predictable ways. For years, designers and researchers have conducted site visits and user studies and have spent hundreds of hours observing how people do things and how they think to do them. An interaction design pattern is a reusable solution to a common usability issue in interface or interaction design (Borchers, 2000).

Interaction design patterns help impose a faster pace in the design of systems, minimizing time and effort employed and ensuring the application of proven solutions, resulting in better



systems. These patterns also promote clearer communication between designers, developers, and application domain experts in an interdisciplinary team, by providing a common terminology to exchange ideas, opinions, and values (Borchers, 2000). Jenifer Tidwell comprises interaction design patterns into eleven different categories (Tidwell, 2010):

- **“What Users Do”** - related to facilitating users to achieve their goals by leveraging common behaviors and emotions. Examples of these patterns include: Safe Exploration Instant Gratification, Incremental Construction, and Personal Recommendations;
- **“Organizing the Content: Information Architecture and Application Structure”** - patterns that help organize, categorize, and order information of the system, making it structured in a way that is the most useful to users. Examples of these patterns include: Feature, Search, and Browse, News Stream, Picture Manager, and Dashboard;
- **“Getting Around: Navigation, Signposts, and Wayfinding”** - patterns that deal with the problem of navigation and aim to minimize the time and energy users spend to get where they want. Examples of these patterns include: Clear Entry Points, Menu Page, Sitemap Footer, and Breadcrumbs;
- **“Organizing the Page: Layout of Page Elements”** - related to structuring the content of the application in a way that the most relevant information for the user rapidly captures its attention. Examples of these patterns include: Visual Frameworks, Titled Sections, Accordion, and Collapsible Panels;
- **“List of Things”** - patterns that cover how to display a list of items in an interactive setting. Examples of these patterns are: List Inlay, Thumbnail Grid, Carousel, Pagination, and Cascading Lists;
- **“Doing Things: Actions and Commands”** - related to components that help the user perform actions or commands inside the system, as well as understanding the status and outcomes of these actions. Examples of these patterns include: Button Groups, Hover Tools, Progress Indicator, and Command History;
- **“Showing Complex Data: Trees, Charts, and Other Information Graphics”** - related to the presentations of information graphics - such as maps, tables, and graphs - that communicate knowledge efficiently and let users use their eyes and minds to draw conclusions rapidly. Examples of these patterns include: Datatips, Dynamic Queries, Local Zooming, and Treemap;
- **“Getting Input from Users: Forms and Controls”** - patterns that deal with asking and retaining information that is input by the user. Examples of these patterns include: Forgiving Format, Input Hints, Input Prompt, Autocompletion, and Same-Page Error Messages;

- **“Using Social Media”** - related to patterns that aim to connect social media links on a website or application, maximizing user interaction with those links. Examples of these patterns include: Personal Voices, Response and Comment, Conversations Starters and Social Links;
- **“Going Mobile”** - related to patterns that aim at making the system responsive and working correctly for any standard device size, such as personal computers, tablets, or phones. Examples of these patterns include: Vertical Stack, Bottom Navigation, Infinite List, and Generous Borders;
- **“Making It Look Good: Visual Style and Aesthetics”** - related to patterns that aim to build a visually appealing and professional-looking system, increasing the trust in the system and visual easiness on the user. Examples of these patterns include: Deep Background, Corner Treatments, Borders That Echo Fonts, and Contrasting Font Weights.

### 2.2.6 Interaction Design for Explainable Artificial Intelligence

A spike in interest in XAI has resulted in a large vast collection of algorithmic research on the subject. While many people realize the importance of including explainability capabilities in AI systems, the topic of how to fulfill real-world user needs for AI understanding remains an open question (Liao et al., 2020).

Most XAI systems follow an algorithm-centric approach and depend on researchers’ understanding about what makes for effective explanations. This can be troublesome because users, who may not have a deep technical understanding of AI, are contextualized differently and hold a different preconception of what constitutes valuable explanations (Ribera and Lapedriza, 2019).

An explanation user interface (XUI) is the sum of the outputs of an XAI process with which the user can interact directly. Two modes of XUIs can be considered (Chromik and Butz, 2021):

- **Explanatory XUIs** - also known as static XUIs, are designed to deliver a single explanation (e.g., a visual or a textual);
- **Exploratory XUIs** - also known as interactive XUIs, let users freely explore the AI model’s behavior, and they’re most effective when users can adjust or impact the inputs.

#### 2.2.6.1 Human-centered XAI

What constitutes a good explanation, i.e., providing information that is comprehended and utilized, is greatly influenced by the receiver’s current knowledge and objective for receiving the explanation, among other human elements. Explainability is an inherently human-centric feature that demands human-centered approaches that focus technological decisions on people’s explainability demands and measure success based on human experience. As a consequence, XAI is as much a design challenge as it is an algorithmic challenge (Ehsan and Riedl, 2020).

HCI researchers and design practitioners are essential to ensure good explainability. There is a lot of space for them to contribute with insights, solutions, and methods to make AI more

explainable. Only by considering cognitive, sociotechnical, and design perspectives can we truly achieve human-centered XAI (Ehsan et al., 2022).

It is possible to move away from a techno-centric focus on developing algorithmic explanations by focusing the study on people, on how they interact with and digest information about AI, and whether they can achieve their goals. Only then is it possible to start looking for ways to improve user experiences between algorithmic explanations and actionable knowledge. XAI design solutions can address how to communicate algorithmic explanations, such as selecting the appropriate modalities, level of abstraction, privacy or security limitations, and so on. They could also take the shape of interventions aimed at influencing how individuals process XAI, such as cognitive forcing functions or coaching to assist people better analyze explanatory data. Furthermore, beyond algorithmic explanations, it is required to address users' knowledge or information gaps in order to reach practical understanding, such as supplying relevant domain expertise and broad concepts of how AI works (Ehsan et al., 2022).

It is important to note that design choices for explainability should be made accordingly to the type and context of users. For instance, while machine learning experts might prefer highly detailed information that requires a certain level of knowledge, users with not much AI expertise expect simple explanations that are easily understandable.

## 2.3 Similar Platforms

Researching existing platforms with similar goals to TRUST is crucial for any dissertation that includes building a platform or part of it. Analyzing those platforms can provide beneficial insights: we can take inspiration from good features, aim to improve the ones that are poorly implemented, find missing opportunities and take them as a way to innovate and separate our platform from the competition.

Throughout the last decade, many AI platforms that let users input data and train models were developed and made available to the public, such as *Amazon SageMaker Autopilot*<sup>1</sup>, *Google Vertex AI*<sup>2</sup>, and *Microsoft Azure AI*<sup>3</sup>, to name a few. However, these platforms were never focused on explainability and are only starting to add some post-hoc explainability features in recent years.

In this section, we analyze platforms that are mainly focused on allowing users to solve typical AI problems using explainability techniques. These platforms are *Heuristiclab*<sup>4</sup>, *Eureqa*<sup>5</sup>, and *TuringBot*<sup>6</sup>.

---

<sup>1</sup>Found at <https://aws.amazon.com/pt/sagemaker/autopilot>

<sup>2</sup>Found at <https://cloud.google.com/vertex-ai>

<sup>3</sup>Found at <https://azure.microsoft.com/en-us/overview/ai-platform>

<sup>4</sup>Found at <https://dev.heuristiclab.com/trac.fcgi>

<sup>5</sup>Found at <https://www.creativemachineslab.com/eureqa.html>

<sup>6</sup>Found at <https://turingbotsoftware.com>

### 2.3.1 HeuristicLab

HeuristicLab is a software environment for heuristic and evolutionary algorithms. HeuristicLab is distinguished from existing heuristic optimization frameworks by a very comprehensive, albeit outdated, graphical user interface, which generally require comprehensive programming abilities to adjust and extend the algorithms for a particular task. The usual workflow of the platform is to start by entering the data used to train and evaluate the model (Figure 2.8), specify model parameters (Figure 2.9), train the model (Figure 2.10), and see its results (Figure 2.11).

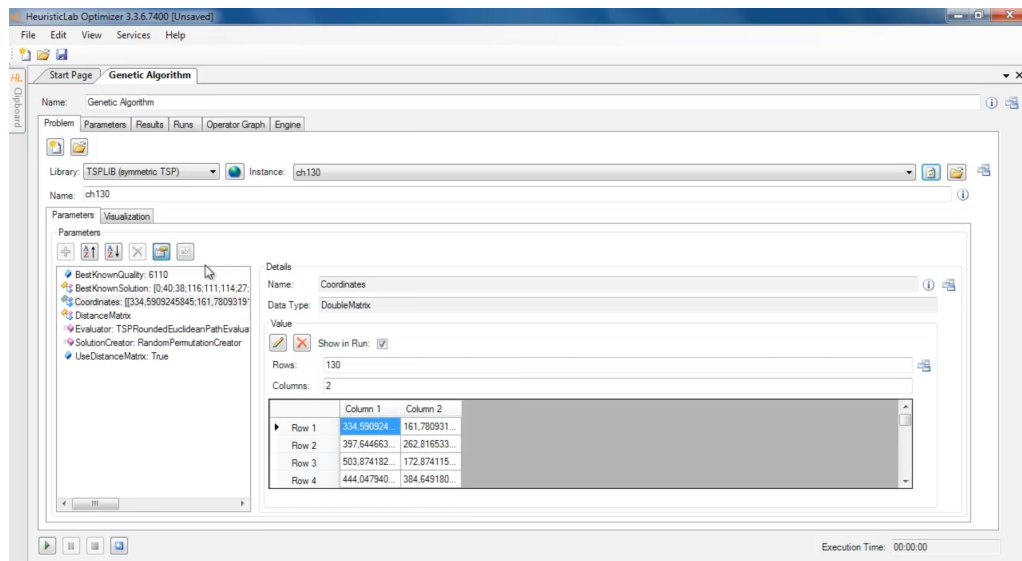


Figure 2.8: Heuristiclab UI - Enter Data Section

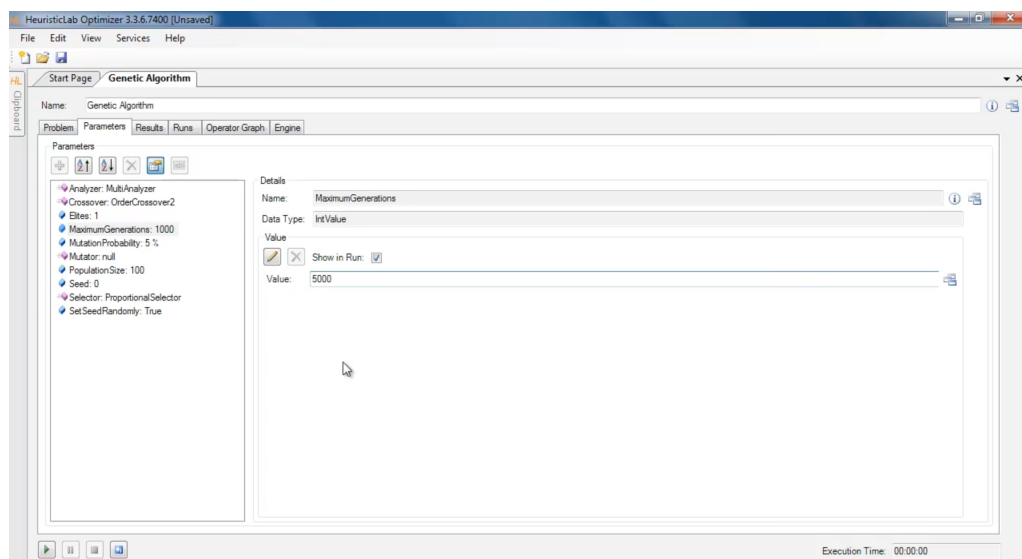


Figure 2.9: Heuristiclab UI - Set Model Parameters Section

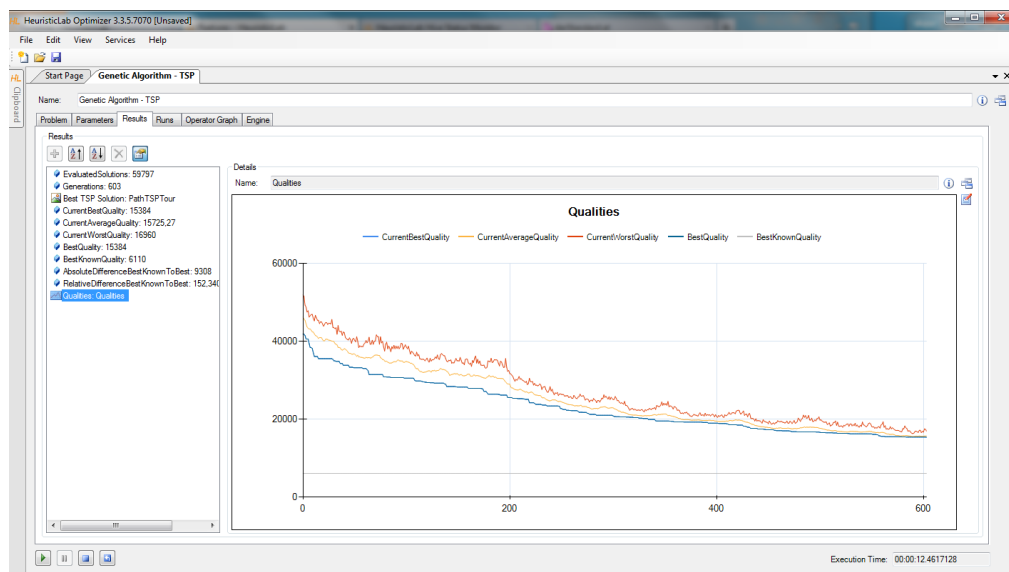


Figure 2.10: Heuristiclab UI - Training Section

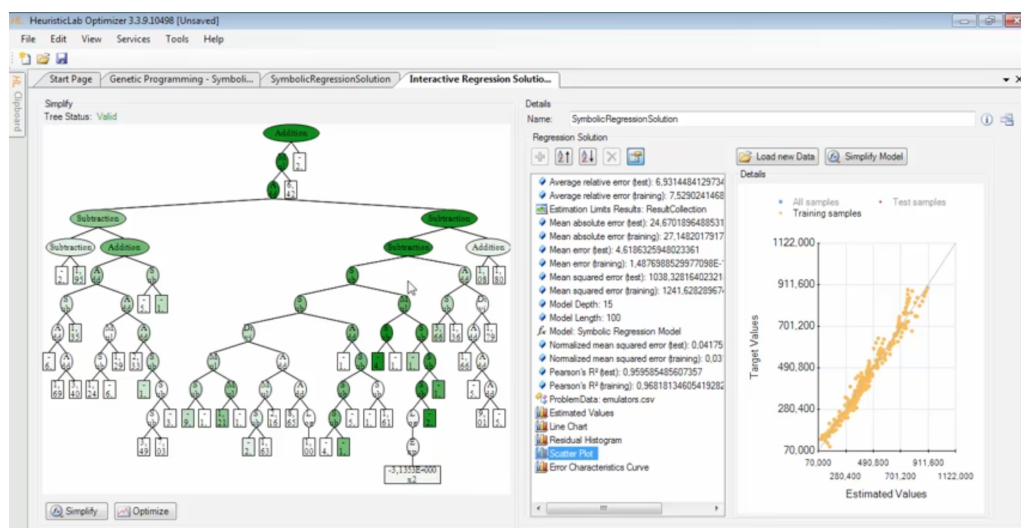


Figure 2.11: Heuristiclab UI - Results Section

### 2.3.1.1 Strengths

Heuristiclab abounds with excellent properties such as:

- Very complete and detailed UI that allows users to intuitively define problems and train models;
- Creation of experiments that examine the comparison of different operators;
- Run, pause, and resume experiments at will;

- Generation of statistics and charts used to analyze results;
- Creation of new algorithms and extension of predefined ones;
- Capacity to interact (manipulate and simplify) with models through tree representation;
- Feature importance;
- It is available for free.

#### **2.3.1.2 Weaknesses**

However, the platform also lacks in some regards:

- The UI is extremely outdated, therefore not being very visually appealing;
- Lacks proper documentation, making it hard to learn;
- Not many explainability features besides feature importance and using interpretable models;
- Software is no longer updated.

### **2.3.2 Eureqa**

Eureqa is a tool that finds equations and hidden mathematical relationships in your data. Its main purpose is to find the simplest mathematical formulas that may be used to describe the data's underlying structure.

The tool introduces a method that can automatically generate sets of symbolic equations using genetic programming. This method is applicable to any system that can be described through non-linear differential equations. It has a similar basic workflow to Heuristiclab starting with data input (Figure 2.12), selection model parameters (Figure 2.13), and visualize its training and results (Figure 2.14).

#### **2.3.2.1 Strengths**

Eureqa implements some excellent features such as:

- Very complete and detailed UI that allows users to intuitively define problems and train models;
- Multimodel Visualizations;
- Run, pause, and resume experiments at will.

	A	B	C	D	E	F	G	H	I	J	K	L	M
desc.	Time of measurement (seconds)	Angle of the Pendulum (radians)											
var	t	x											
1	0.08	1.19											
2	0.10	1.03											
3	0.12	0.86											
4	0.14	0.81											
5	0.16	0.36											
6	0.18	0.09											
7	0.20	-0.13											
8	0.22	0.47											
9	0.24	-0.52											
10	0.28	0.79											
11	0.28	-0.90											
12	0.30	1.10											
13	0.32	-1.21											
14	0.34	-1.25											
15	0.36	-1.32											
16	0.38	-1.21											
17	0.40	-1.21											
18	0.42	-1.13											
19	0.44	-1.05											
20	0.48	0.89											
21	0.48	-0.72											
22	0.50	0.67											
23	0.52	-0.29											
24	0.54	0.92											
25	0.56	0.22											
26	0.58	0.60											
27	0.60	0.88											

Figure 2.12: Eureka UI - Enter Data Section

### 2.3.2.2 Weaknesses

But again, it also lacks in some regards:

- The UI is extremely outdated, therefore not being very visually appealing;
- Lacks proper documentation, making it hard to learn;
- Not many explainability features besides using interpretable models;
- Doesn't allow to run experiments and compare different models;
- Only supports regression problems;
- Does not support tree visualizations;
- Does not support model manipulation;
- Is no longer commercially available.

### 2.3.3 TuringBot

TuringBot is a desktop program that finds mathematical formulas from data values via symbolic regression. TuringBot is a very similar platform to Eureka, but instead of using genetic programming to fit the data, it is based on simulated annealing.

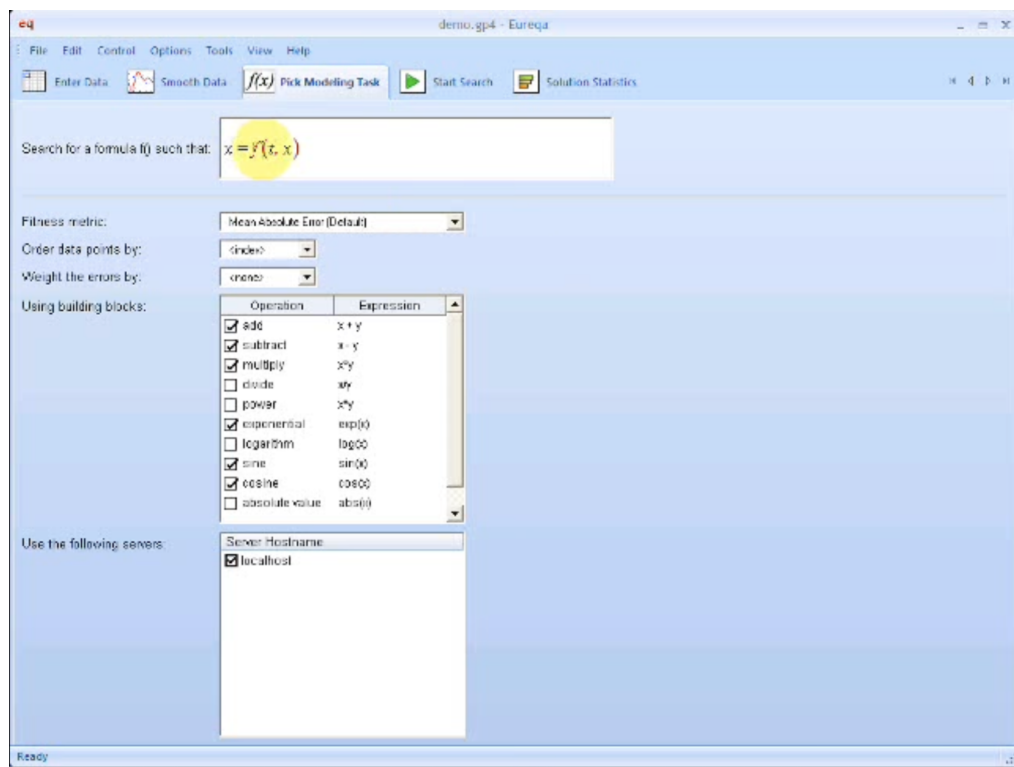


Figure 2.13: Eureka UI - Set Model Parameters Section

Although simplistic, the tool offers a modern and intuitive UI that quickly gets you started with real work. It has almost the same workflow as Eureka but in a simpler way, as the user is capable of selecting the input, model parameters, and visualizing training and results in the same Section (Figure 2.15). There is also a Section where it is possible to visualize and filter logs (Figure 2.16).

### 2.3.3.1 Strengths

TuringBot excels in features such as:

- Very complete and detailed UI that allows users to intuitively define problems and train models;
- Multimodel Visualizations;
- Clean and modern UI;
- Has a programmable API;
- Run, pause, and resume experiments at will;
- Has excellent documentation;
- It has a powerful free version available.



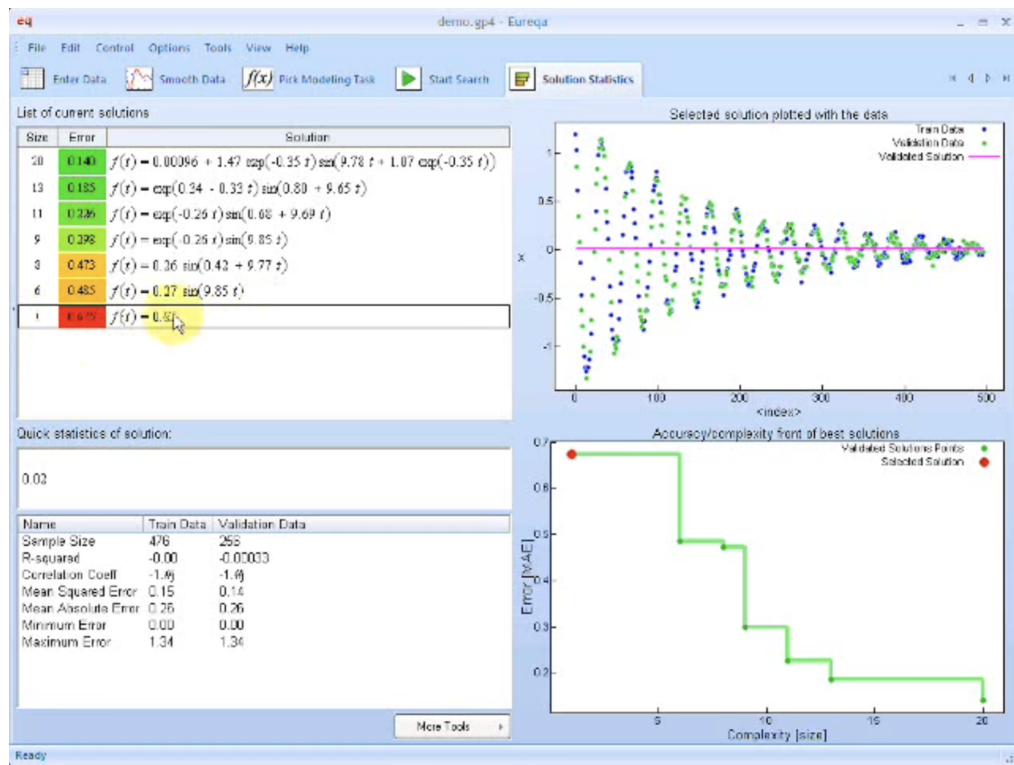


Figure 2.14: Eureka UI - Training and Results Section

### 2.3.3.2 Weaknesses

But leaves a lot to desire in some aspects:

- Not many explainability features besides using interpretable models;
- Doesn't generate a lot of statistics about data and results;
- Doesn't allow to run experiments and compare different models;
- Does not support tree visualizations;
- Does not support model manipulation.

### 2.3.4 Summary

There are some powerful tools using explainability techniques that are focused on utilizing interpretable models. Overall, HeuristicLab is the most complete and versatile tool out of the three, as it holds an enormous variety of features and even allows users to define their own problems by implementing custom genetic programs. However, the platform stopped being updated, leaving it with an outdated UI and almost no available documentation.

Between Eureka and TuringBot, the latter was made to be a better version of the first one and thrives on the UI and UX regard. TuringBot, although lacking when it comes to visualizing results

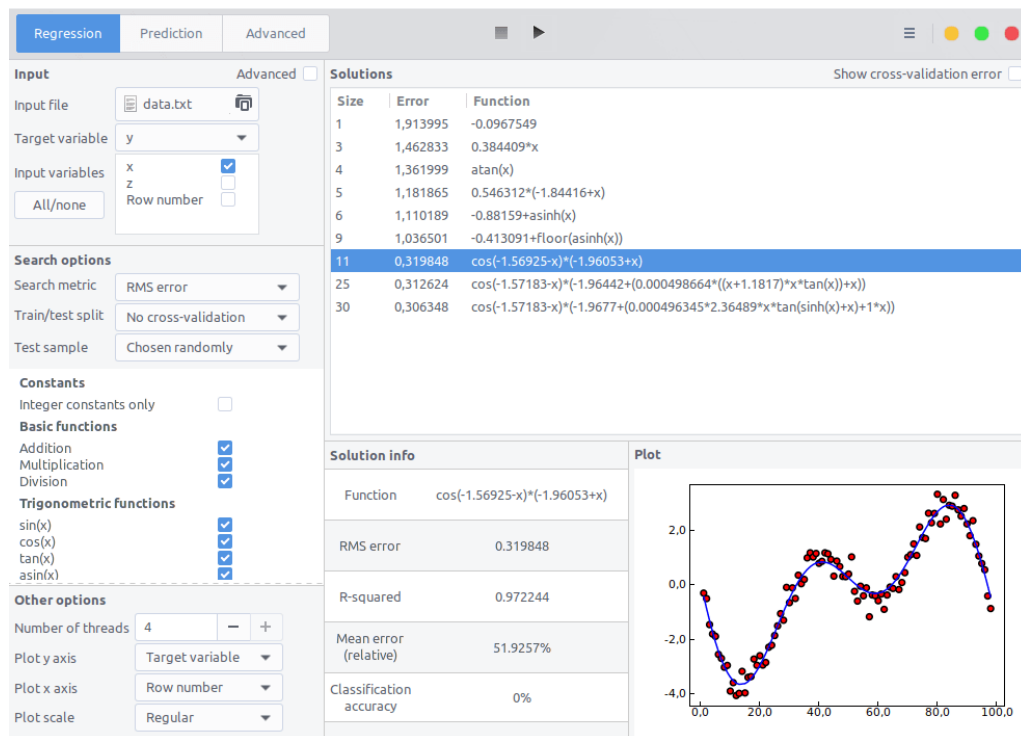


Figure 2.15: TuringBot UI - Set Model Parameters, Train and Visualize Results Section

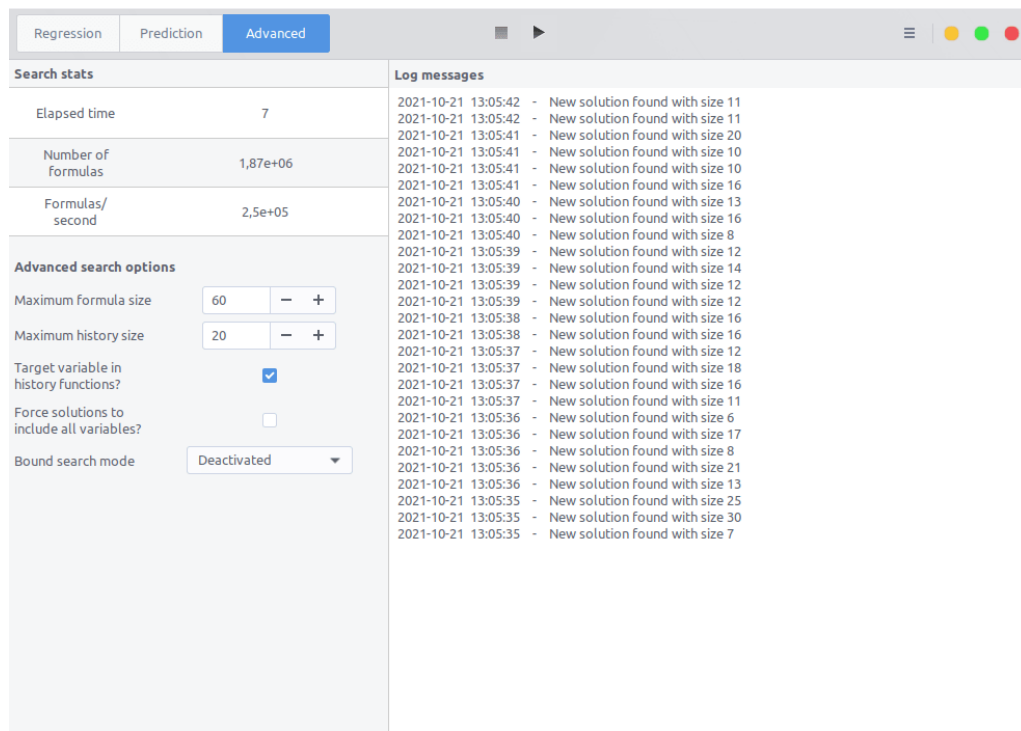


Figure 2.16: TuringBot UI - Visualize and Filter Logs Section

and being much simpler than HeuristicLab, has detailed documentation and programmable API that can help researchers and developers come together with ease and shows the potential to grow even further as it implements more functionalities.

In general, these tools lack actionable explanations that lead the user to improve the model and a human-guided AI. Still, they have some relevant features and patterns that can provide us with inspiration and work towards a more complete and richer platform.

## Chapter 3

# Problem and Proposed Approach

In this chapter, Section 3.1 analyses the details and current knowledge about the problem at hand, challenges, and obstacles to have in mind. It follows with the definition of an approach, in Section 3.2.

### 3.1 Problem Description

This dissertation aims at designing, implement, and evaluate the user interface for TRUST. This platform should allow users to train genetic programs using their own data and use them to solve classification, regression, and prescriptive problems.

The main problem resides in discovering the best user interface design methods to achieve better explainability during users' interactions with the system, as interaction design for XAI is a topic that still lacks thorough research and shared practices.

The UI of TRUST should not only enhance explainability but also empower the users with actionable information that allows them to interact with the AI by improving it and learning from it, ultimately leading to a human-guided XAI system, as shown in Figure 3.1, and bridging the gap between humans and machines.

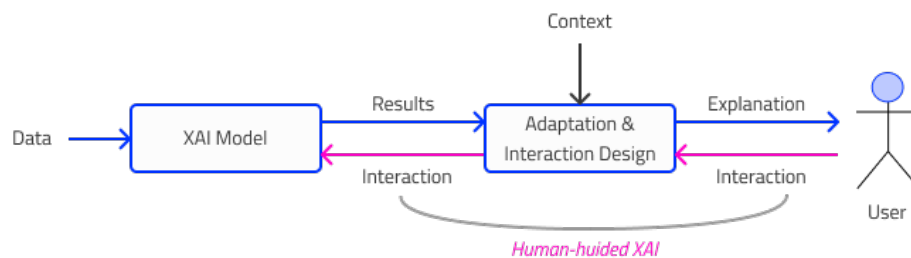


Figure 3.1: Human-guided XAI Framework

In terms of usability, the user experience should be as easy and intuitive as possible. The end-user should be able to navigate the website and perform tasks with ease. For this purpose, the

application should provide a minimalist user interface that only shows what is necessary and allow the user to focus on what is essential.

A smooth human-machine interaction is essential to building a system that involves the human in the learning process, as shown in Figure 3.1. Thus, the information on the interface must be readily available, and it should be transparent for the user when and what he can interact with, especially when it comes to interacting with the models.

The developed prototype will be focused on the usage of symbolic models to solve problems, which have the potential to achieve high levels of explainability and perform well in various problems. We aim to allow users to view and interact with the generated symbolic models through text and graphical visualizations.

As shown in Figure 3.2, TRUST envisions three main types of users to interact with the application:

- **Algorithm Expert** - is in charge of adding algorithms to the platform, whether generic or problem-specific. The Algorithm Expert possesses the greatest depth of understanding regarding AI methods and techniques;
- **Model Developer** - has some level of understanding of the problem's domain and the algorithms present in the platform. The Model Developer is responsible for configuring and training the algorithms (runs) to produce models that can be customized for a specific problem;
- **Domain Expert** - holds the expertise in the domain of the problem and is usually a decision-maker. As such, the Domain Expert will use the models generated by the platform to achieve solutions and make decisions about a particular problem.

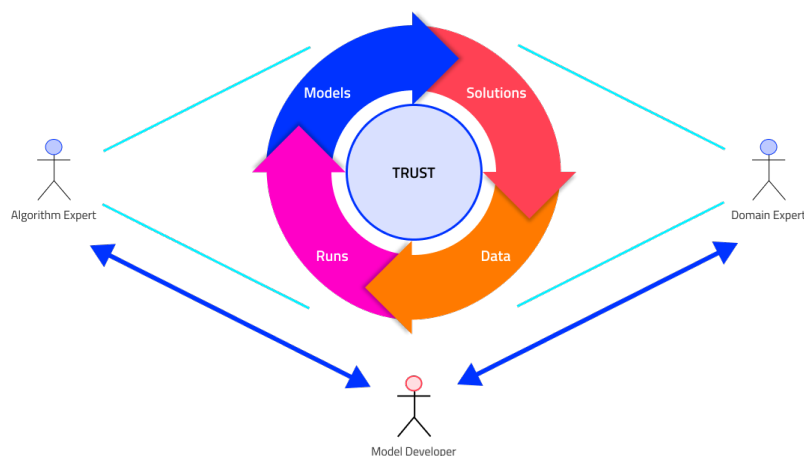


Figure 3.2: TRUST Platform Users

It is important to note that these types of users are not necessarily mutually exclusive. One user might be a part of two or more different user categories. For instance, a Medical Doctor with

knowledge about Genetic Programming can develop its own models in the platform and use them as a support to make decisions.

In the scope of this dissertation, we will focus mainly on designing features that support the needs of the Model Developer.

## 3.2 Solution Approach and Methodology

We aim to build the user interface following the guidelines of a user-centered design methodology (described in [User-centered Design](#)) and respect the principles of [Usability](#) and [User Experience](#). For that reason, we have devised a typical UCD methodology while aiming toward a system that supports [Human-centered XAI](#).

A cooperative design methodology like this will help identify and understand what users value more in the context of human-AI interaction, resulting in a better user experience and a more complete product.

Our methodology will follow a four-phase process with the goal of understanding, defining, designing, and evaluating, as shown in [Figure 3.3](#). A modular process explicitly links requirements to design solutions through a design rationale that involves the users and contributes to existing theory by facilitating the production of reusable knowledge for XAI ([Schoonderwoerd et al., 2021](#)). This methodology will also encompass state-of-the-art [Design Research Methods](#) that will allow understanding the emotional, cultural, and aesthetic context that the platform will exist in.

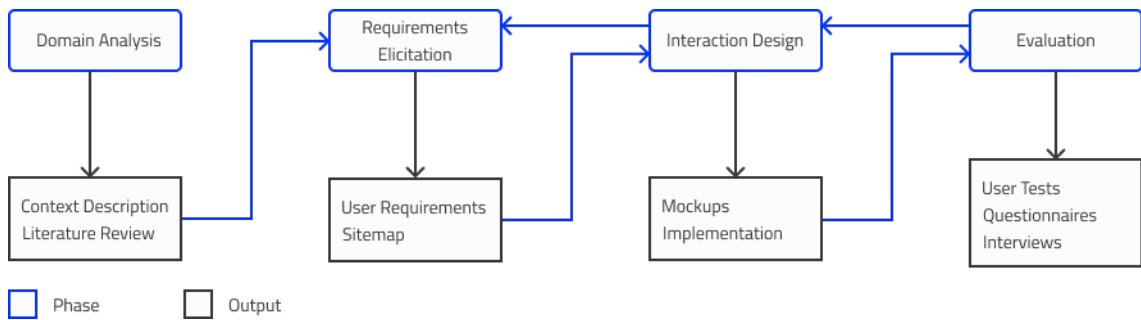


Figure 3.3: Flow diagram of the proposed methodology for UCD

### 3.2.1 Domain Analysis

A typical starting point for user-centered design is understanding the context of use in which the system will be introduced. The goal here is to recognize which questions we want to give an answer to and which problems we want to solve. In our case, we want to know who is going to use the TRUST tool, what are the primary functions of the system, what are the expected benefits of using the system, and how to improve human-system interaction in this context.

A domain analysis as such allows us to take into account human values during the design process, and it will be performed mainly by consulting available literature about topics such as eXplainable Artificial Intelligence, Interaction Design, and similar existing tools.

The output of this phase is a description of the context and goals of the user interface, with answers to the aforementioned questions and an extensive literature review. These outputs are present in Sections **Context**, **Goals**, **Problem Description**, and Chapter **Literature Review**.

### 3.2.2 Requirements Elicitation

After gathering the information from the previous phase, we will be able to start identifying the requirements and expectations that users pose for our system. To elicit user requirements while actively involving our end-users, we intend to discuss directly with them and perform questionnaires. These requirements will also allow us to build a sitemap where all the pages and their main components are identified and connected, serving as a high-level representation of the TRUST tool.

It is expected that during the Interaction Design and Evaluation phases, some requirements might change due to user feedback.

### 3.2.3 Interaction Design

The requirements analysis provides insights about what information the users want to receive from the system, but not how it should be presented. This phase aims to discover how that information can be effectively communicated to the user.

This phase will involve researching and choosing appropriate methods to present the information, create mock-ups, validate them with the project's stakeholders, and, finally, implement them. We are also aiming to develop solutions that can be easily applicable to other XAI applications.

### 3.2.4 Evaluation

The evaluation phase is one of the most important phases to guarantee a user-centered design since it allows humans to provide formal feedback (i.e., through user tests, questionnaires, etc.), providing us with qualitative and quantitative metrics that will help us identify weaknesses and strengths of the platform. These metrics can then be used to iterate the design of the platform and even the requirements, contributing to a high level of user acceptance.

It is important to note that all this process is iterative, and some of the phases can overlap with each other. This level of flexibility is necessary to make the most out of the users' input.

## 3.3 Summary

In this dissertation, we tackle the problem of designing and implementing the user interface of a platform that is used to solve classification, regression, and prescriptive problems, using genetic programming. The interface must be easy to use and intuitive and allow smooth human interaction.

For that purpose, a user-centered design methodology was devised, which includes performing domain analysis, requirements, elicitation, interaction design, and evaluation.



## Chapter 4

# Requirements Elicitation

Gathering requirements helps define a blueprint of objectives, rules, and criteria to which a final product should adhere. By answering as many questions as possible, they allow the reduction of the project's scope and help stakeholders make sure the project's development is on the right track.

This chapter starts with Section 4.1 that identifies the strategy followed to elicit the requirements that supported the definition of implementation of the interface of the platform. Section 4.2 describes the non-functional requirements, while Section 4.3 describes the key functional requirements. Section 4.4 follows by describing the use cases produced to satisfy those requirements. Then, in Section 4.5 the proposed sitemap of the platform is presented and the content for each of its pages and sections.

### 4.1 Methodology

In order to gather the platform's requirements, the stakeholders and possible end-users of the project were actively involved during the process. The requirements of this project were mainly elicited through techniques such as interviews, focus groups, workshops, brainstorming sessions, and document analysis, which were performed more intensively in the early stages of the design process, but kept happening throughout the process during various meetings.

These techniques helped collect information whose analysis resulted in the formal definition of functional and non-functional requirements.

### 4.2 Non-functional Requirements

Non-functional requirements are specifications that describe a system in terms of operational capabilities and constraints that support its functionality (Hartson and Pyla, 2012). This includes characteristics such as usability, performance, and security. The user interface of the platform should abide by the following key requirements:

- **Usability** - the user experience should be as easy and intuitive as possible. The end-user (usually a decision-maker) should be able to navigate the website and perform tasks with

ease. For this, the application should provide minimalist interfaces that only show what is necessary and allow the user to focus on what is essential;

- **Performance** - the system should have short response times to ensure the user's attention and a smooth user experience. Loading indicators should be shown, allowing the user to understand that the application is working and not get impatient;
- **Security** - the system shall protect information from unauthorized access through an authentication and verification system;
- **Server interaction** - once a service is needed, HTTP requests will be sent to the Application Server, which will respond with JSON files;
- **Human-guided learning** - a smooth human-machine interaction is vital to involve human intelligence in the learning process. Thus, the user interface must include easy access to information and interactive elements that clearly show the user can interact with the models and their training process;

### 4.3 Functional Requirements

Functional requirements are specifications that outline what functionalities a system should support in order to enable users to accomplish their tasks. These requirements describe the system's behavior under specific conditions and should respect non-functional requirements (Hartson and Pyla, 2012). The elicited functional requirements that the user interface of the TRUST project should support are listed in Table 4.1.

Table 4.1: TRUST - Key Functional Requirements

RID	Name	Requirement
R1	Create XAI Projects	Create projects to better organize datasets, algorithms, and sessions
R2	Upload datasets	Inside a project, upload multiple tabular datasets (e.g., CSV files) that should persist in a database for later access and use
R3	Train algorithms	After picking a dataset, and selecting a GP algorithm that is adequate for the problem at hand, fine-tune its parameters and specify operators
R4	Visualize training	Visualize information about the training process in real-time (textually and graphically), such as performance statistics and, most importantly, results
R5	Filter solutions	Filter the generated symbolic models during training phase, using different metrics

R6	Visualize solutions	Visualize the model's mathematical expression as well as its tree format
R7	Evaluate solutions	Visualize indicators/metrics about the model's overall performance as well as information and statistics that contribute to the model's explainability
R8	Compare solutions	Compare any of the saved trained models inside an XAI project side by side
R9	Interact with solutions	Fine-tune the model, including changes/simplifications to the mathematical expression or even rearrange its nodes
R10	Save solution	After the training session is finished, save any of the generated models for later access or usage
R11	Use the solutions	Use generated and saved models, on new data points
R12	Explanations	Human-understandable explanations (e.g., feature importance) should be generated, capable of giving the user enough confidence for the user to trust the model or adjust it

## 4.4 Use Cases

After the requirements were specified, we designed use cases in the form of user stories, which define how the functional requirements will be met. User stories are short descriptions of a system's features from the users' perspective. They focus on the interaction between user and interface and help define more thoroughly what a system should be capable of doing (Hartson and Pyla, 2012). Table 4.2 shows the defined user stories (UID) associated with the respective functional requirements (RIDs).

Table 4.2: TRUST - User Stories

UID	RIDs	As a user, I want to	So that
US1	R1	Create a project	I can better organize my training sessions
US2	R1	List my projects	I know what projects are available to me
US3	R1	Delete a project	When it is no longer relevant to me, it disappears
US4	R1	Search projects	I can easily find a project that I want to open
US5	R2	Upload a dataset in a project	I use it to train a session
US6	R2	View the datasets of a project and their data	I know what data is available to use on my sessions

US7	R2	Delete datasets from a project	When datasets are no longer relevant to a project, they disappear
US8	R2	Download datasets	I can use its data for other tasks or upload it to another project
US9	R1	Search datasets	I can easily find a dataset whose data I want to see
US10	R3	Create a session	I can train an algorithm inside that session
US11	R3	List the sessions of a project	I know what sessions are available in that project
US12	R3	Delete a session	When it is no longer relevant to me, it disappears
US13	R3	Search sessions	I can easily find a session that I want to open
US14	R3	Set a session's algorithm	I can run that session with a specific algorithm
US15	R3	Set a session's dataset	I can train an algorithm using a specific dataset
US16	R3	Set a session's configuration	I can train an algorithm using a specific configuration
US17	R4	See the session status	I can know if the results are ready or if something went wrong
US18	R4	See the best solutions of each generation, and its metrics, during the training process	I know how the training process is progressing and its results
US19	R4	See the metrics evolution in a chart	I know how the training process is progressing and its results
US20	R4	Stop a session from running	I do not have to wait for the last generation to be ready, in case I want to see the results earlier
US21	R6, R7, R8	See solutions, and its metrics, generated in one or more sessions, on a table	I can find and compare the ones that are relevant to me
US22	R6, R7, R8	Order solutions, using their metrics, generated in one or more sessions	I can easily find the ones that are relevant to me
US23	R10	Bookmark solutions	I can easily find and use them later

US24	R5, R6, R7	Filter solutions using their metrics	I can easily find the ones that are relevant to me
US25	R6, R7, R8	See solutions, and its metrics, generated in a session, on a scatter-plot	I can find and compare the ones that are relevant to me
US26	R6, R7, R8	Pick the metrics shown in the scatter-plot	I can find and compare the solutions that are relevant to me
US27	R6, R7, R8	See solutions, and its metrics, generated in a session, on a bar-chart	I can find and compare the ones that are relevant to me
US28	R6, R7, R8	Pick the metrics shown in the bar-chart	I can find and compare the solutions that are relevant to me
US29	R6, R7, R8	See a solution in text and tree format side-by-side	I can understand how a model functions
US30	R6	See a solution in <i>LaTeX</i> format	I can easily read the solution's expression
US31	R6	Copy a solution in <i>LaTeX</i> format	I can easily include it in my <i>LaTeX</i> documents
US32	R8	See general statistics about a session	I can see how a solution performs when compared to the other solutions of the same sessions
US33	R9	Edit a solution in its text format	I can tweak it and see how it performs
US34	R9	See which operators and terminals are available in the problem	I can use them to tweak an expression

Note that no user stories were created for requirements R11 and R12. This is because, as mentioned in the **Problem Description** Section, we prioritized the focus on the Model Developer user. R11 and R12 are heavily related to the Domain Expert user, but due to time constraints, they have not been tackled.

## 4.5 Sitemap

With all of the use cases defined, a sitemap, which is a blueprint of the website's pages and main section, was defined (Figure 4.1). This sitemap helps understand the overall structure of the

website and the navigation flow between its different components and main pages (Hartson and Pyla, 2012).

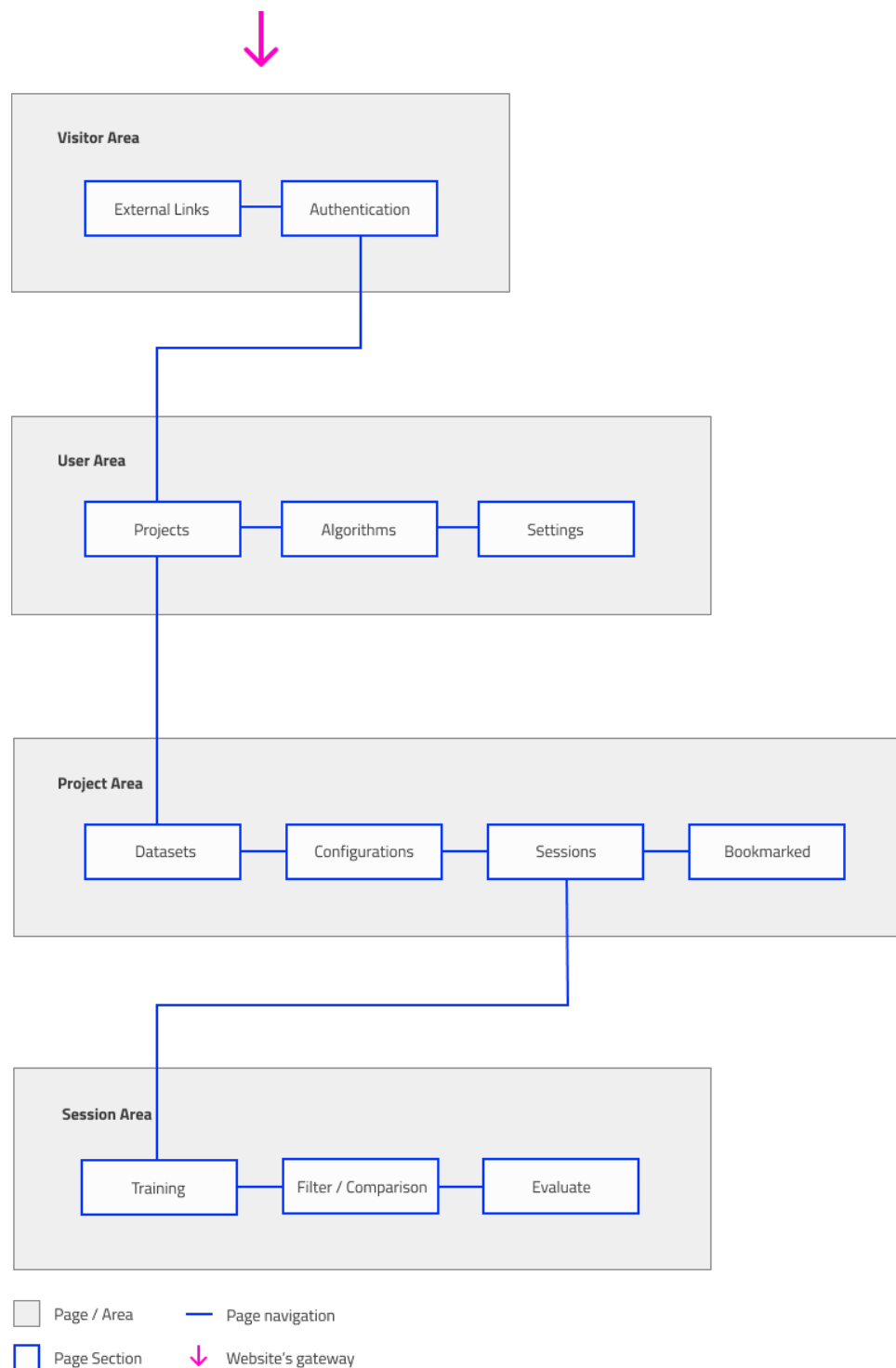


Figure 4.1: TRUST - High-level Sitemap

We can separate the platform's user interface into four different areas:

- **Visitor Area** - the first point of contact with the end-user, containing sections for **Authentication** and **External Links**;
- **User Area** - after a visitor authenticates itself, the sections **Projects**, **Algorithms**, and **Settings** become available to it;
- **Project Area** - a user can have various projects. When he selects one of them, the user navigates to the respective project area, which contains the necessary components to manage a project's datasets, configurations, datasets, and saved results, namely the **Datasets**, **Configurations**, **Sessions**, and **Bookmarked** sections;
- **Session Area** - a project can hold various sessions, and when one is selected, the user navigates to the respective session area, which contains the components necessary to start a session, visualize and interact with the generated solutions, namely the **Training**, **Filter / Comparison**, and **Evaluate** sections.

It is worth noting that inside each area, the user can freely navigate directly to any of its Sections.

Before initiating the **Interaction Design** phase, we have listed all of the elements and contents that the website's pages and sections should contain. This list served as an initial guide for development but suffered modifications through the development. Appendix **A** presents the latest version of the content inventory. A catalog like this enables the stakeholders to understand how information is organized and make sure no element is missing during the development phase.

## 4.6 Summary

Requirements elicitation is an essential step of the design process as it enables us to define clear goals and constraints that the platform should adhere to. The system's requirements were elicited through techniques such as interviews, workshops, focus groups, and brainstorming sessions. The definition of those requirements allowed us to specify the platform's use cases, design a sitemap and create a content inventory, which lists most of the website's page elements and content.

## Chapter 5

# Interaction Design

Requirements analysis reveals what the users want to receive from the system, but it does not address how that information should be received. It is during the Interaction Design phase that we determine how that information should be delivered. This phase involves prototyping, creating mock-ups, and implementing solutions that meet the requirements.

This chapter describes the Interaction Design strategy used in the TRUST project to implement its user interface, which is detailed in Section 5.1. It follows by presenting and explaining the resulting solutions in Section 5.2 and their primary capabilities in Section 5.3. Finally, Section 5.4 details how those solutions were implemented and the rationale behind them.

### 5.1 Methodology

Although the requirements elicitation was an iterative process throughout the design of the interface, so was its actual implementation. The interaction design process followed a user-centered design approach, as described in Section **User-centered Design**, and started as soon as the first requirements were defined. This phase required the employment of research techniques, such as the ones listed in the **Design Research** Section, to discover how to effectively communicate with the user and implement the defined user stories.

With the user requirements defined, we produced design solutions, namely high-fidelity mock-ups (using the *Figma*<sup>1</sup> design tool), that aimed at meeting those requirements while following the principles of usability, that were previously defined in Section **Usability**. We have also incorporated some of the patterns described in Section **Interaction Design Patterns** to guarantee the design of reliable and consistent solutions.

The developed mockups were evaluated and validated against the user requirements. This was done iteratively through different meetings with the stakeholders and possible end-users of the project. This evaluation allowed people with multidisciplinary skills and perspectives to review the design, resulting in refinement, sometimes change of requirements, and, finally, acceptance. After being accepted, the designed solutions were implemented as part prototype and re-validated.

---

<sup>1</sup>Found at <https://www.figma.com/>



## 5.2 Screens and Features

At the end of the Interaction Design phase, we have ended up with the following screens that can be mapped to the components detailed in the [Sitemap](#) Section.

### 5.2.1 Authentication Page

The authentication page, shown in Figures 5.1 and 5.2, is the first point of contact for the end-user, allowing it to create an account and log onto it. This can be performed by toggling between the “Sign In” and “Sign Up” forms. On this page, the user can also see the TRUST logo and slogan that redirects it to the project’s page when clicked. At the bottom, it also contains the logos of each partner of the project.

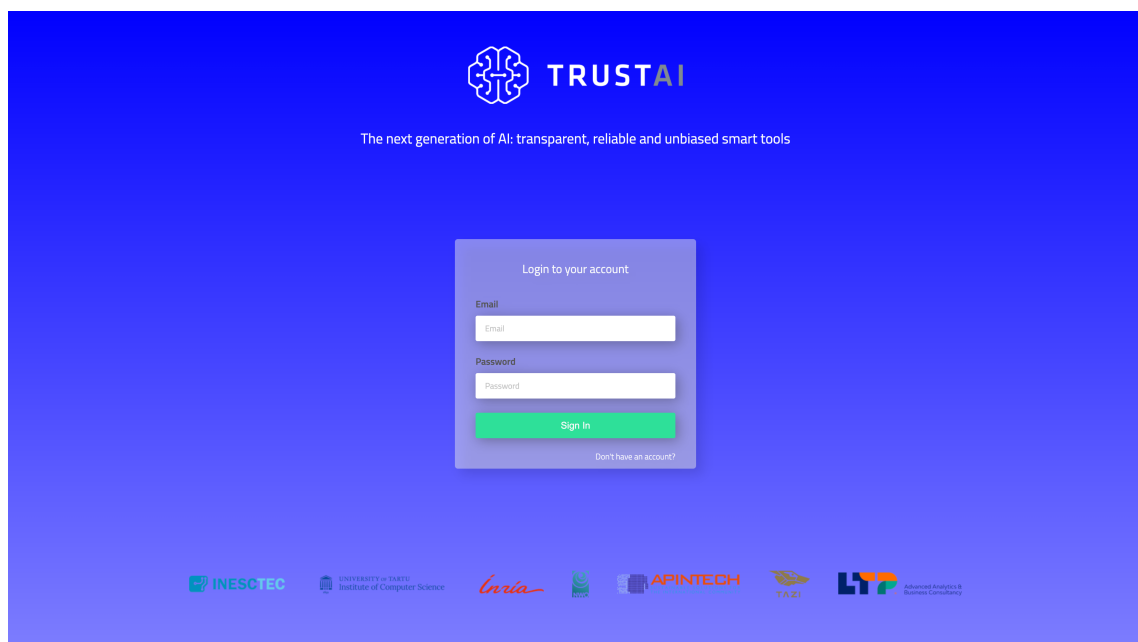
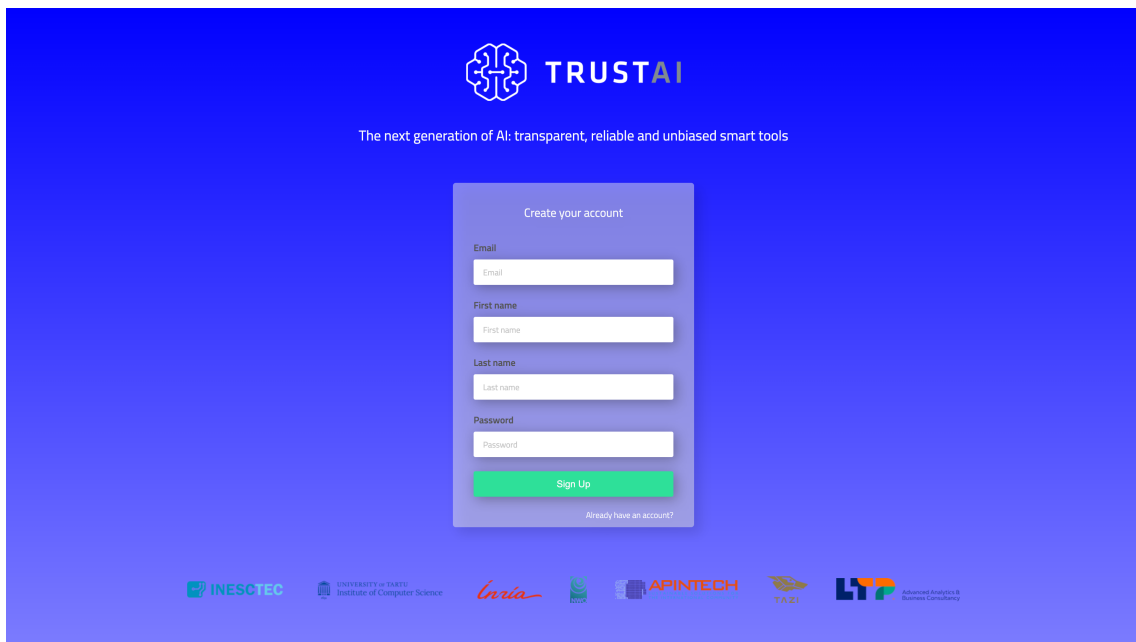


Figure 5.1: TRUST - Authentication Page (Sign In)

### 5.2.2 Projects Page

The project page, shown in Figure 5.3, is the first page the user sees after logging in. This page contains a list of the XAI projects created by the user that can be filtered or searched using their names. Here, the user can also create or delete a project. To create a project, a user must click on the “New Project” button, which will open a modal form containing the necessary fields for project creation. After creating a project or selecting an existing one, the user will be redirected to the [Datasets Section](#).



The image shows the TRUST AI sign-up page. At the top, the TRUST AI logo is displayed with the tagline "The next generation of AI: transparent, reliable and unbiased smart tools". Below this is a "Create your account" form with fields for Email, First name, Last name, and Password. A green "Sign Up" button is at the bottom of the form, with a link "Already have an account?" below it. At the bottom of the page, there is a row of partner logos: INESC TEC, UNIVERSITY of TARTU Institute of Computer Science, Inria, INRAE, APINTECH, T IN ZI, and LYP Advanced Analytics & Business Consulting.

TRUST AI

The next generation of AI: transparent, reliable and unbiased smart tools

Create your account

Email

First name

Last name

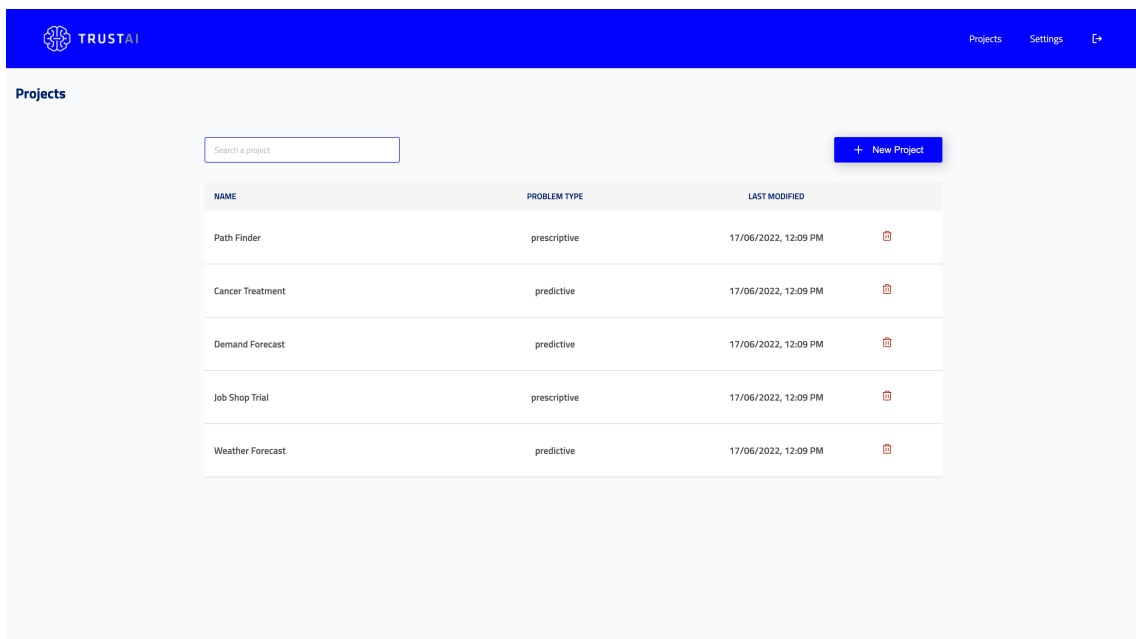
Password

Sign Up

Already have an account?

INESC TEC UNIVERSITY of TARTU Institute of Computer Science Inria INRAE APINTECH T IN ZI LYP Advanced Analytics & Business Consulting

Figure 5.2: TRUST - Authentication Page (Sign Up)



The image shows the TRUST AI Projects page. At the top, the TRUST AI logo is displayed. Below it is a "Projects" section with a search bar and a "+ New Project" button. A table lists the projects with columns for NAME, PROBLEM TYPE, and LAST MODIFIED. Each row has a red delete icon on the right.

TRUST AI

Projects Settings [+]

Projects

Search a project

+ New Project

NAME	PROBLEM TYPE	LAST MODIFIED
Path Finder	prescriptive	17/06/2022, 12:09 PM
Cancer Treatment	predictive	17/06/2022, 12:09 PM
Demand Forecast	predictive	17/06/2022, 12:09 PM
Job Shop Trial	prescriptive	17/06/2022, 12:09 PM
Weather Forecast	predictive	17/06/2022, 12:09 PM

Figure 5.3: TRUST - Projects Page

### 5.2.3 Datasets Section

The datasets section, shown in Figure 5.4, contains a list of the datasets belonging to a project. The user can download, delete and select them to view their data in a table format. Here, the user can search for datasets and upload new ones by clicking the "Upload" button and choosing a file.

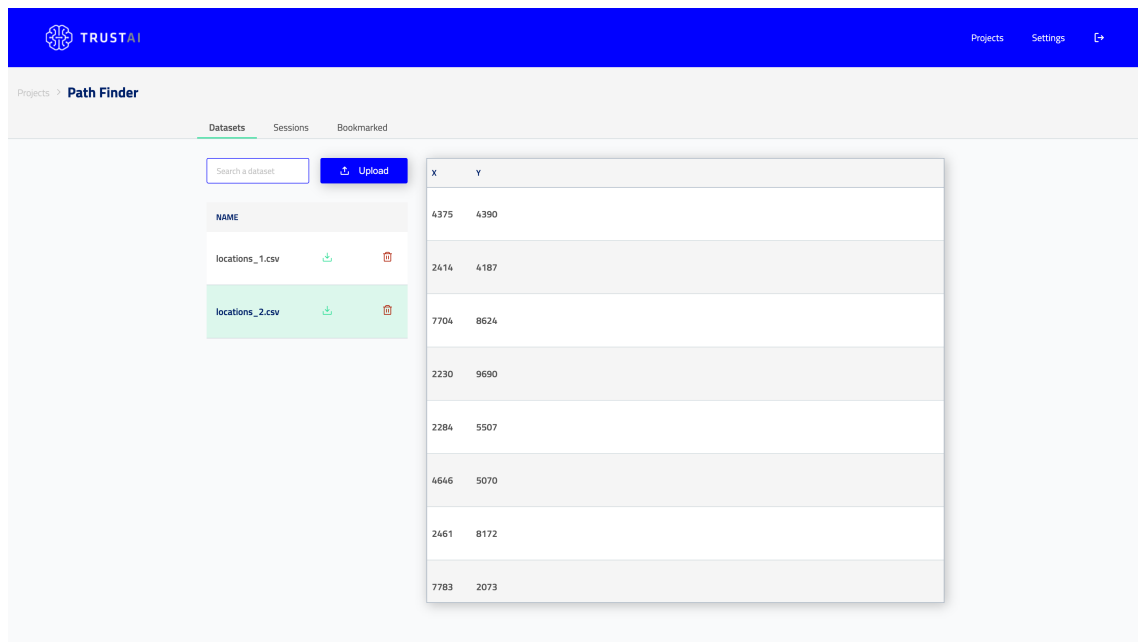


Figure 5.4: TRUST - Datasets Section

### 5.2.4 Sessions Section

The sessions section, shown in Figure 5.5, contains a table with all the sessions created inside the project. Each entry of the table has the name and algorithm of the sessions, its status (e.g., “Created”, “Running”, and “Finished”), the date when it was last modified, and a button to delete it. The user can create a new session by clicking the “New Session” button, which will open a modal with a form for the session’s name, the dataset to be used, algorithm, and configuration. After creating a session or selecting an existing one, the user will be taken to the [Training Section](#).

### 5.2.5 Bookmarked Section

The bookmarked section, shown in Figure 5.6, contains a table with all the models that were bookmarked inside the project. Each table entry has the model’s expression, its metrics, and a button to remove it from the bookmarked models list. When the model’s expression is clicked, it is automatically copied to the user’s clipboard.

### 5.2.6 Training Section

The training section, shown in Figure 5.7, is where the user can start training the actual algorithm. For that purpose, the “start” button needs to be clicked. On the left side of the page, the user can also edit the session settings, such as the dataset, algorithm, and configuration. After the training starts, it is possible to stop the training by clicking on the “stop” button, and to visualize the content of three different windows:

Projects > Path Finder

Datasets Sessions Bookmarked

Search a session [+ New Session](#)

NAME	STATUS	ALGORITHM	LAST MODIFIED
Demo Session	Finished	TSP	17/06/2022, 12:19 PM
First Session	Finished	TSP	17/06/2022, 12:19 PM
New Config Test	Created	TSP	17/06/2022, 12:19 PM
Comparison Session	Running	EAS	17/06/2022, 12:19 PM

Figure 5.5: TRUST - Sessions Section

Projects > Path Finder

Datasets Sessions Bookmarked

EXPRESSION	FITNESS	COST	TIME	DISTANCE	DELAYS
$(((((time - eTW) + (slack/dist)) - ((slack - sTW) + (slack/time)))) / (((s$	209590	54590	155.85	10090	4.45
$(((((sTW * sTW) - (eTW + slack)) / ((sTW - dist) / (eTW - slack))) + (((sla$	195100	40100	143.1	9100	3.1
$(((((time * time) * (time + time)) + ((sTW + dist) - (eTW/dist))) + (((dist$	174500	19500	76.1	4500	1.5
$((dist * time) / (eTW * eTW))$	50350	15350	90.05	5350	1
$((dist + sTW) - (dist/dist))$	49400	14400	73.7	4400	1
$((sTW/dist) + (dist + sTW))$	49400	14400	73.7	4400	1
$(sTW / (slack/dist) + slack))$	88105	53105	130.25	8605	4.45
$(time * time)$	29950	14950	83.7	4950	1
$(sTW * dist)$	36200	21200	95.4	6200	1.5

Figure 5.6: TRUST - Bookmarked Section

- a table with the best solution of each generation where each entry has the generation number, the solution's expression, and the generated metrics. When the model's expression is clicked, it is automatically copied to the user's clipboard;

- a chart with the evolution of the metrics throughout the different generations, where the user can select which metrics to visualize;
- the logs that are generated by the algorithm that is running.

These three windows are resizable and can be dragged to be organized as the user prefers. These structural changes are saved for users so that their favorite settings are applied when re-visiting the section. After the training finishes, the sections **Filter Section** and **Evaluation Section** become available.

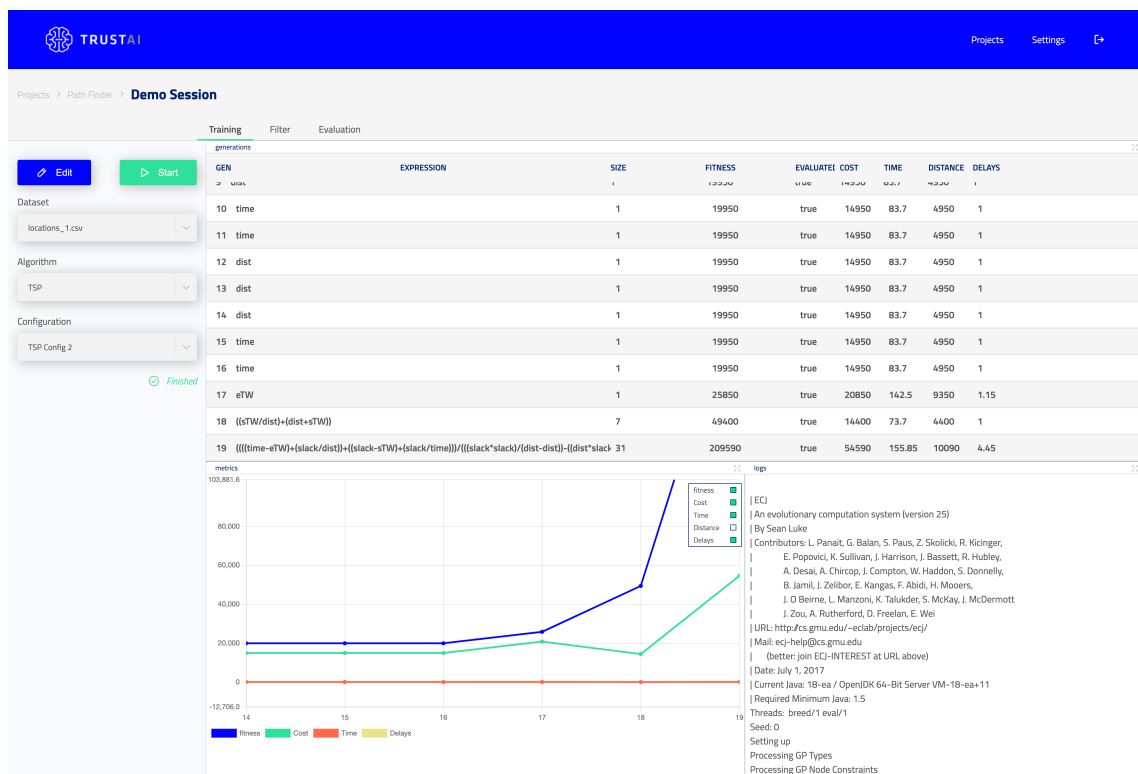


Figure 5.7: TRUST - Training Section

### 5.2.7 Filter Section

The filter section, shown in Figure 5.8, is one of the main components of the user interface. Here the user can compare all the generated solutions in a session. This is possible through three different windows that work similarly as in the training section.

- a table with all the solutions, where each entry should have the solution id (that also identifies the generation), the solution's expression, the generated metrics, and a button to bookmark it. These entries can be ordered using the different available metrics;
- a scatter-plot with all the solutions represented and two drop-down buttons to select which metrics appear on the X and Y axes;

- a bar chart with all the solutions represented, where the user can select which metrics to visualize.

On the left side of the page, the user has a dropdown button that allows the addition of other sessions for comparison. Each session will be attributed a different color in the scatter plot. After that, a slider for each of the metrics allows the user to filter the represented solutions (after clicking on the “Apply” button).



Figure 5.8: TRUST - Filter Section

## 5.2.8 Evaluation Section

The evaluation section, shown in Figure 5.9, is also an extremely important one. Here, the user can evaluate the expressions that passed the filters applied in the **Filter Section**. There are four main windows represented in this section:

- a tree visualization of the solution where the user can zoom in and out and scroll through the tree to better analyze it. This window also contains arrow buttons allowing the user to change the analyzed solution. This change is also applied to all the other windows;
- a text visualization of the solution that is also editable. Here, the user can write a “?” to get access to a list of the available operators and a “\$” to see the variables of the problem. After the user clicks the “Apply” button, the changes are applied unless there is a parsing or syntax

error which will trigger the appearance of an informative error message and an indicator of where the error happened. In this window, the user can also visualize the expression in *LaTeX* format and copy the expression in that same format with just a click so that expressions can be easily included in any *LaTeX* document;

- a table containing the metrics for the solution represented and general statistics about the session metrics;
- a bar chart containing the metrics for the solution represented and general statistics about the session metrics where the user can select which metrics to visualize.

These four windows function similarly to the ones in the [Training Section](#) and [Filter Section](#) and with even more functionalities as the user can also decide to close or open these same windows.

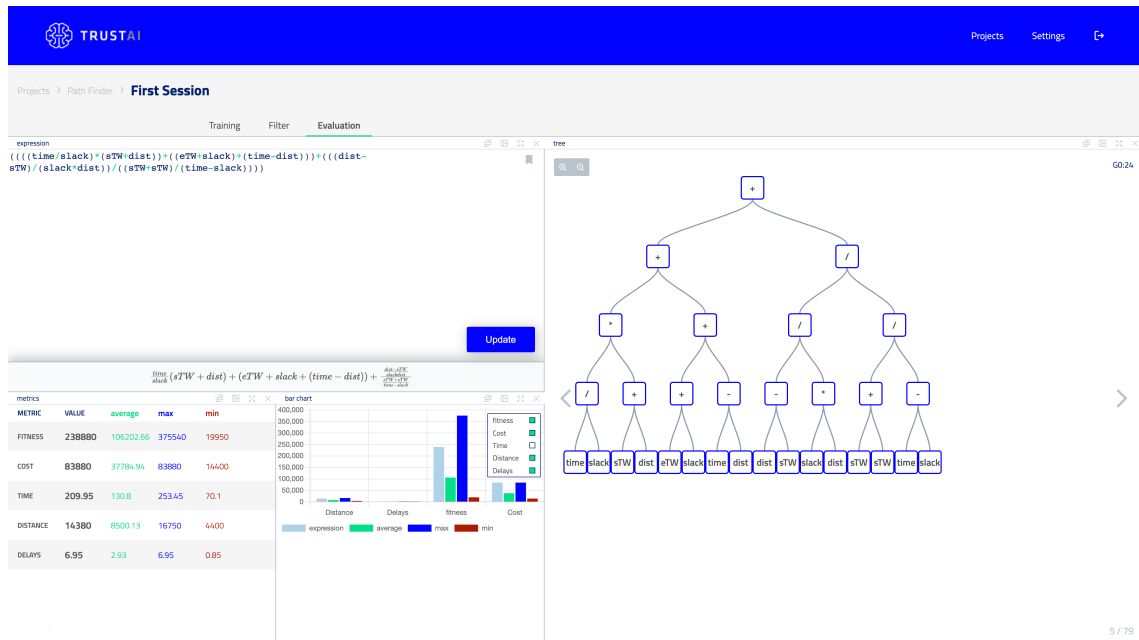


Figure 5.9: TRUST - Evaluation Section

## 5.3 Storyboards

Storyboards are a method to represent interactions between the user and the system using a sequence of interfaces and explaining how navigation is done between them. This section describes the main storyboards of the TRUST platform.

### 5.3.1 Sign In and Sign Up

The storyboard shown in Figure 5.10 shows how the user can interact with the Authentication page. A page visitor can create an account by filling out the “Sign Up” form. If the user has an

account already, then the “Already have an account?” button can be clicked, and the “Sign In” form will appear instead, where the user can insert his username and password and click the “Sign In” button to log in. This flow is essential for a user to access the rest of the website’s content.

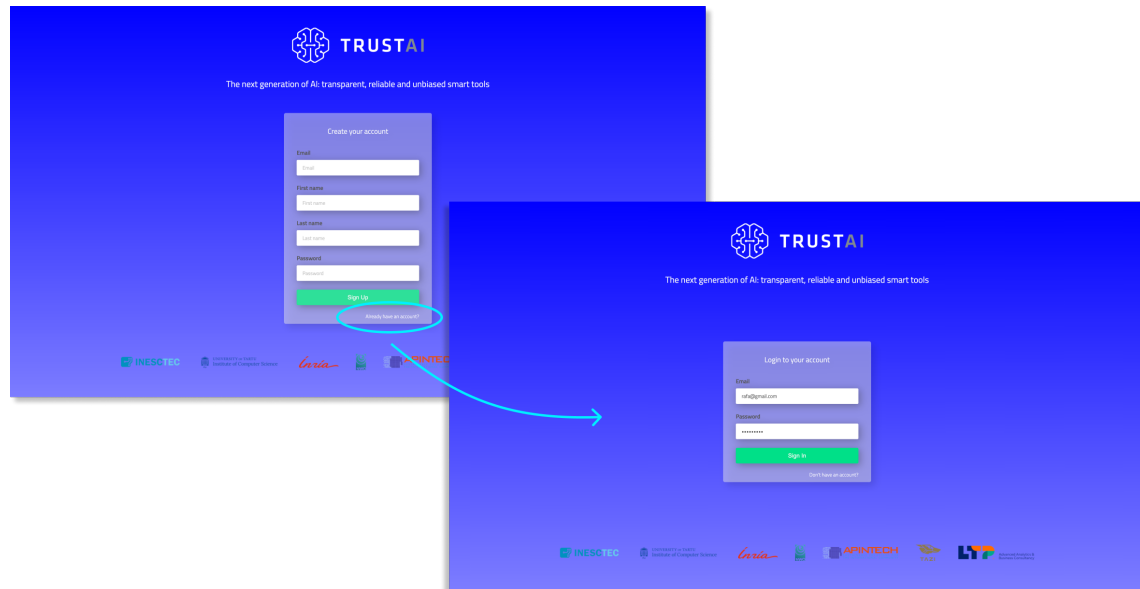


Figure 5.10: Sign In & Sign Up storyboard

### 5.3.2 Project Creation

The storyboard shown in Figure 5.11 represents how a user can create a project. This is done on the “Projects” page by clicking on the “New Project” button. This action will trigger the appearance of a modal section where the user can pick a name and type for the project. When the user clicks the “Create” button, the project is created, and the user is redirected to the project’s page. This storyboard exercises the user stories US1, and US2.

### 5.3.3 Session Setup

For a session to be created, first, a project must contain at least one dataset to be used. As shown in Figure 5.12 a dataset can be uploaded in the “Datasets” section by clicking the “Upload”, selecting a file, and clicking the “Upload” button. The user will then be able to visualize the dataset’s data and use it when creating a session. This flow exercises user stories US5, and US6.

As seen in Figure 5.13, to create a session, the user needs to click the “New Session” button in the “Sessions” sections. Then a model will be shown where the user can pick a name, dataset, algorithm, and session configuration. The session is created when the user clicks the “Create” button, which redirects it to the “Training” section. This exercises user stories US10, US14, US15, and US11. This storyboard is essential for the user to be able to train an algorithm, as that is only possible inside a session.



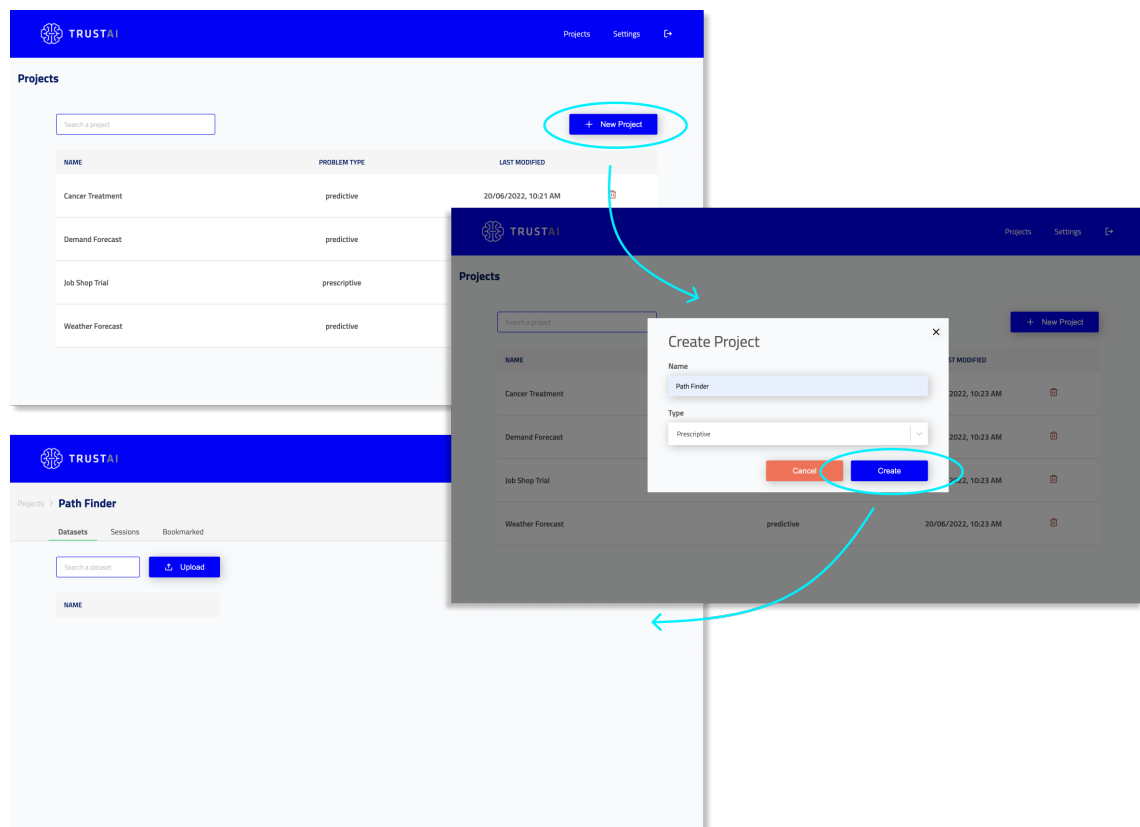


Figure 5.11: Project Creation storyboard

### 5.3.4 Training

The storyboard shown in Figure 5.14 represents how a user can train an algorithm. This is done by opening a pre-existing session in the “Sessions” section and clicking on the “Start” button. This will trigger the train to start, and the user will be able to visualize the results throughout the different generations in real-time. This storyboard exercises user stories US17, US18, and US19. Training an algorithm is an essential step for the user to obtain solutions to their problems.

### 5.3.5 Filter Solutions

As seen in Figure 5.15 the user can filter results in the “Filter” section by changing the sliders for the available metrics and then clicking the “Apply” button. These filters are beneficial for the user to quickly discard unwanted solutions and reach a small set of easily comparable solutions. This storyboard exercises user stories US20, US23, US24, and US26, and it is an essential step for users to find only those solutions that meet their requirements.

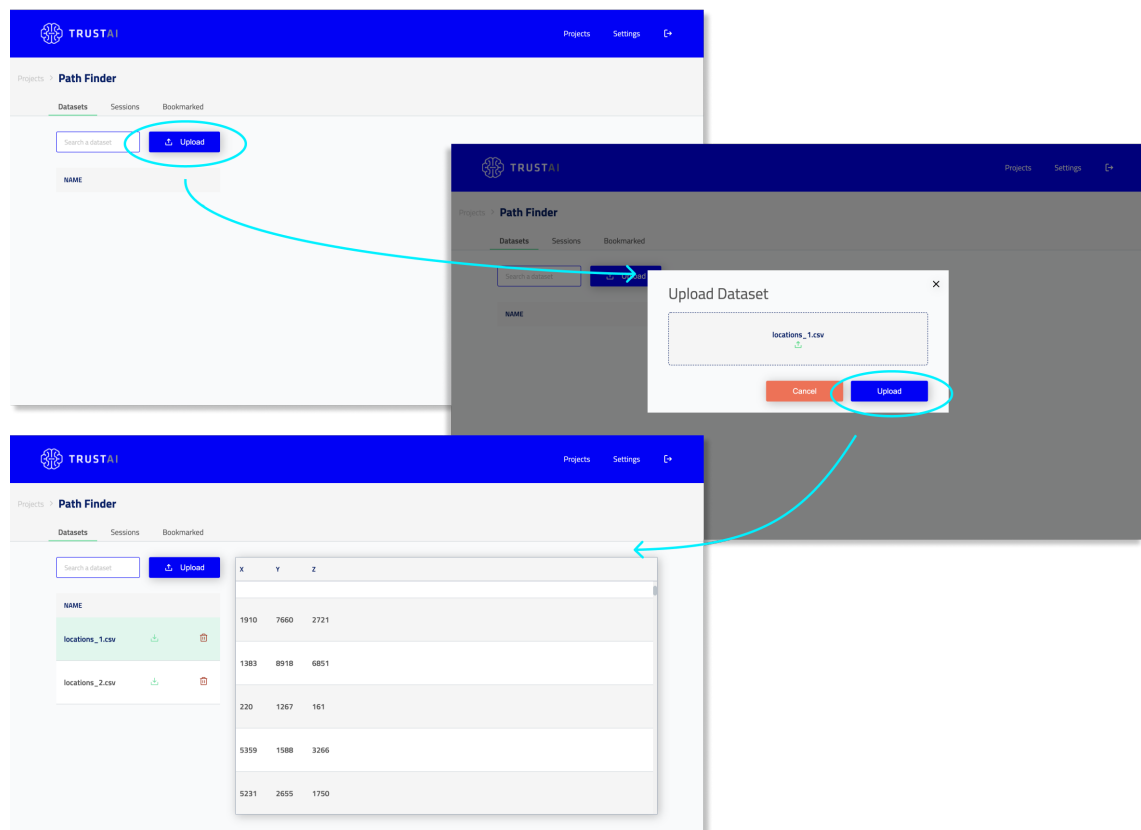


Figure 5.12: Session Setup storyboard - Dataset Uploading

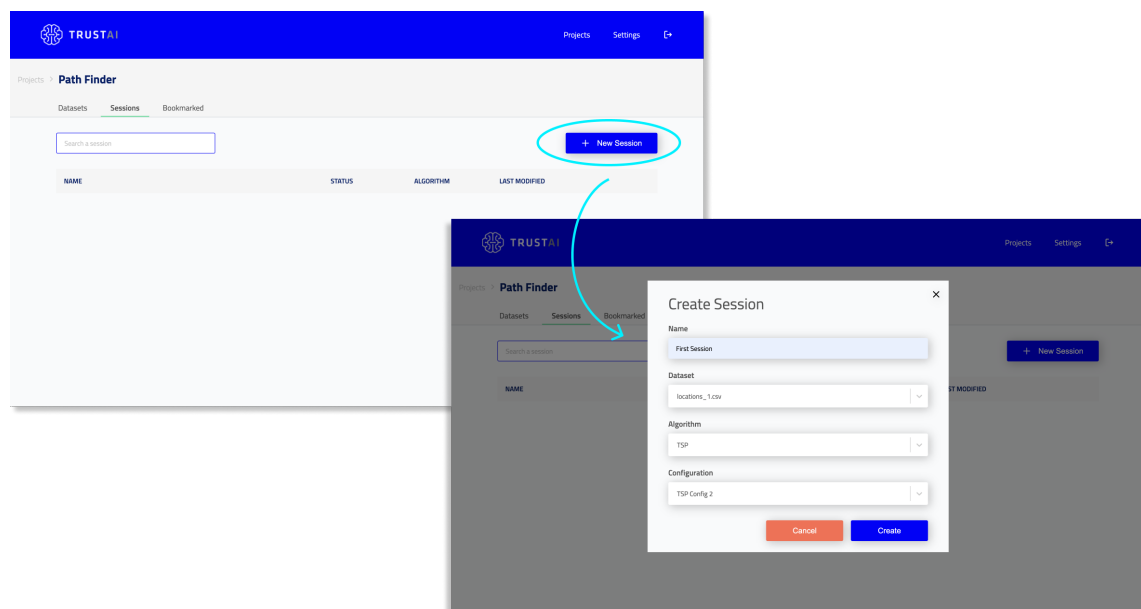


Figure 5.13: Session Setup storyboard - Session Creation

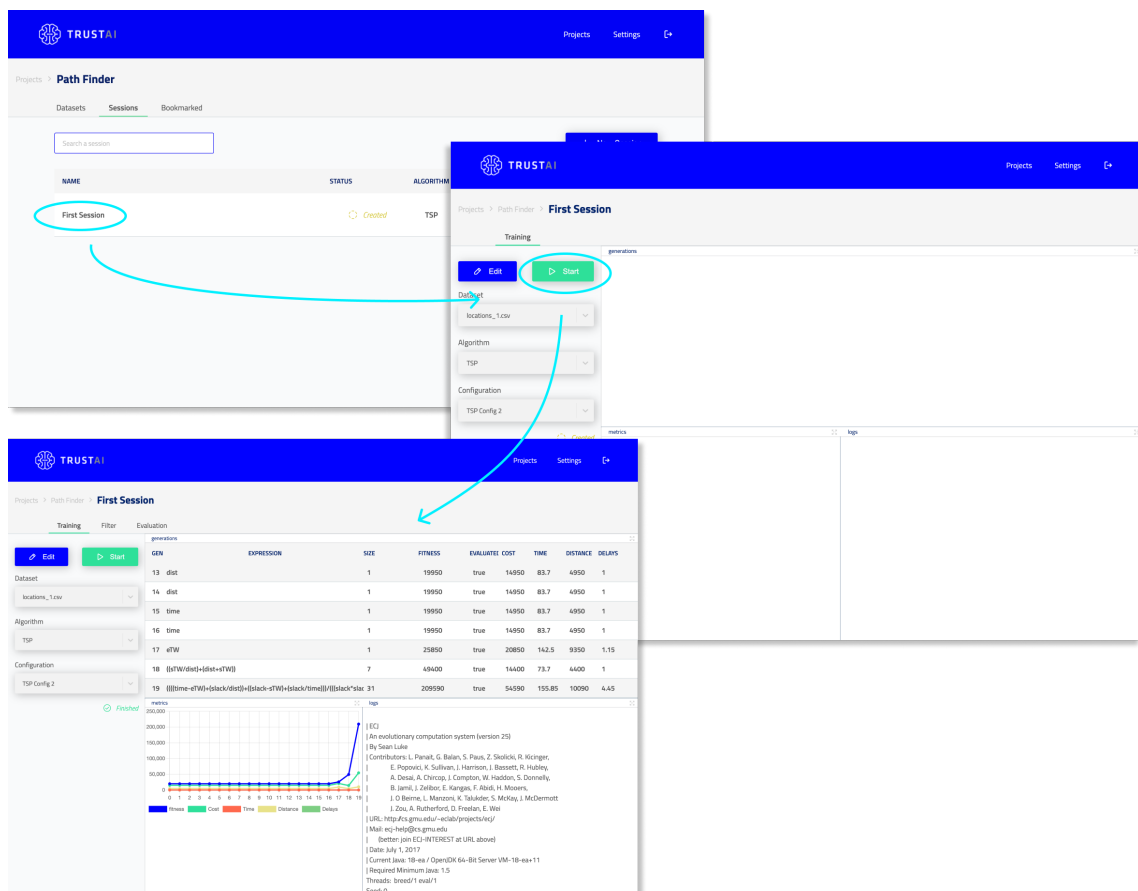


Figure 5.14: Training storyboard

### 5.3.6 Compare Sessions

Besides comparing results from the same session, it is also possible to compare results between different sessions as long as the solutions of both sessions are comparable (i.e., they are being used to solve the same problem). As shown in Figure 5.16 this can be done in the “Filter” section by clicking on the “Sessions” dropdown and adding one of the available sessions. For better clarity, each session will have a different attributed color in the metrics scatter plot. This flow exercises the user story US20, and allows users to compare results of different sessions and ultimately find the solutions that meet their requirements.

### 5.3.7 Highlight and Save Solution

Figure 5.17 shows how users can bookmark an expression that caught their interest. This can be done in the table that contains the filtered solutions by clicking on the “bookmark” icon, which will appear blue when an expression is bookmarked. An expression can be easily found on the table by clicking on a solution (i.e., a point) that appears in the scatter plot, as this will trigger the table to automatically scroll to the selected expression and highlight it with a different color.

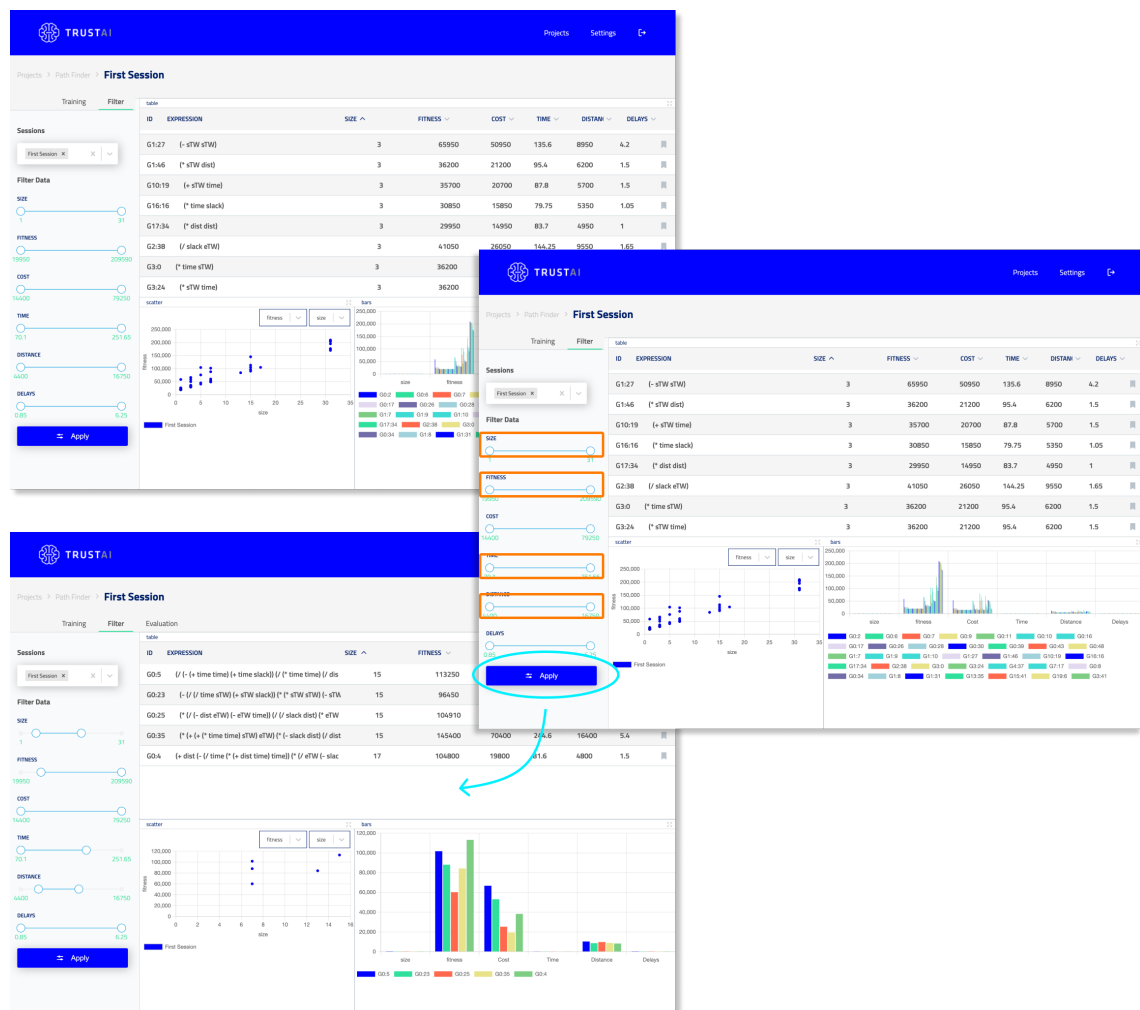


Figure 5.15: Filter Solutions storyboard

This exercises user stories US20, US22, US24, and US25. Overall, this storyboard is important for users to have an alternative manner to find solutions and save them for later usage.

### 5.3.8 Evaluate Solution

Figure 5.18 shows how a user can more deeply inspect and evaluate a solution. After filtering the solutions, the user can go to the "Evaluation" section and see the resulting expressions individually, using multi-modal visualizations. This storyboard exercises user stories US28, US29, and US32, and allows users to more thoroughly understand a specific solution.

### 5.3.9 Rearrange Screen

Although all sections on the "Project" page can be rearranged, the "Evaluation" screen has some additional functionalities. Here, as shown in Figure 5.19 and Figure 5.20 it is possible to add and remove windows from the screen. Currently, there are four possible windows, and they can

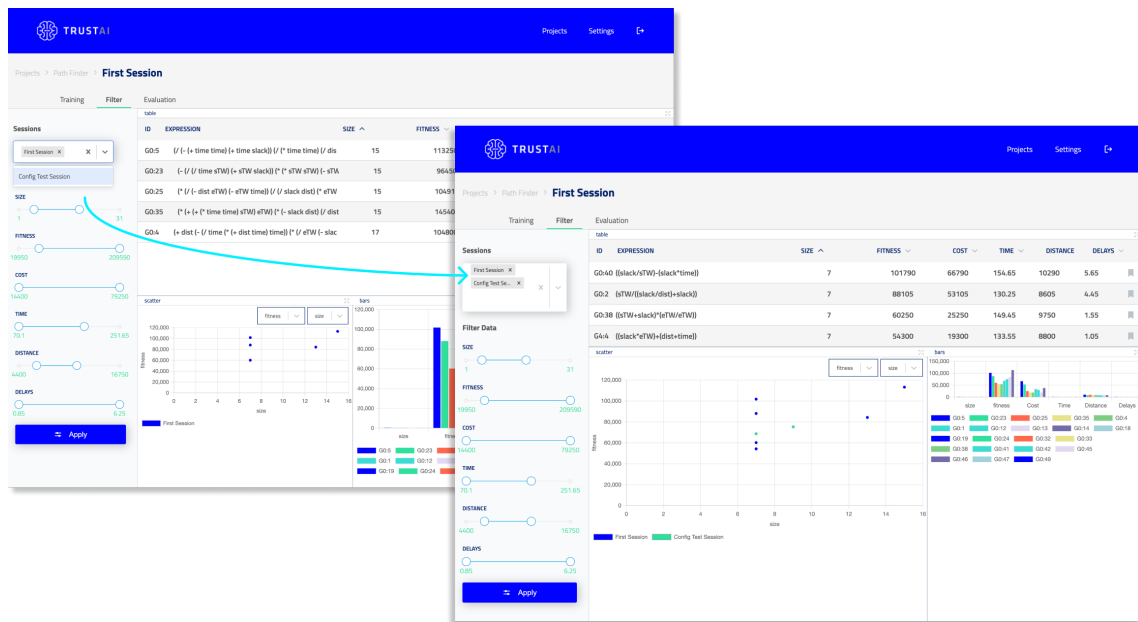


Figure 5.16: Compare Sessions storyboard

also be resized and rearranged by clicking on a window's header and dragging it. This storyboard is extremely helpful as depending on the case and the user's goal, users might need to visualize different information. This storyboard allows them to pick what and how they want to see that information. The resulting layout will be saved for each user so that the screen stays the same for when they come back.

### 5.3.10 Edit Expression

As shown in Figure 5.21 the user can edit an expression in the "Evaluation" screen. The "expression" window works as a text editor with some highlighting features for easier visualization. The user can also write a "?" to get suggestions for the available operators or a "\$" for the available variables. After making a change, the user can click the "Update" button to apply those changes. If they are valid, the other windows will change accordingly. If the changes are invalid, an error message will appear and a visual indicator of where the error happened. This flow exercises user stories US32 and US33, and it is a key part of the project as it allows users to directly interact and modify the generated solutions.

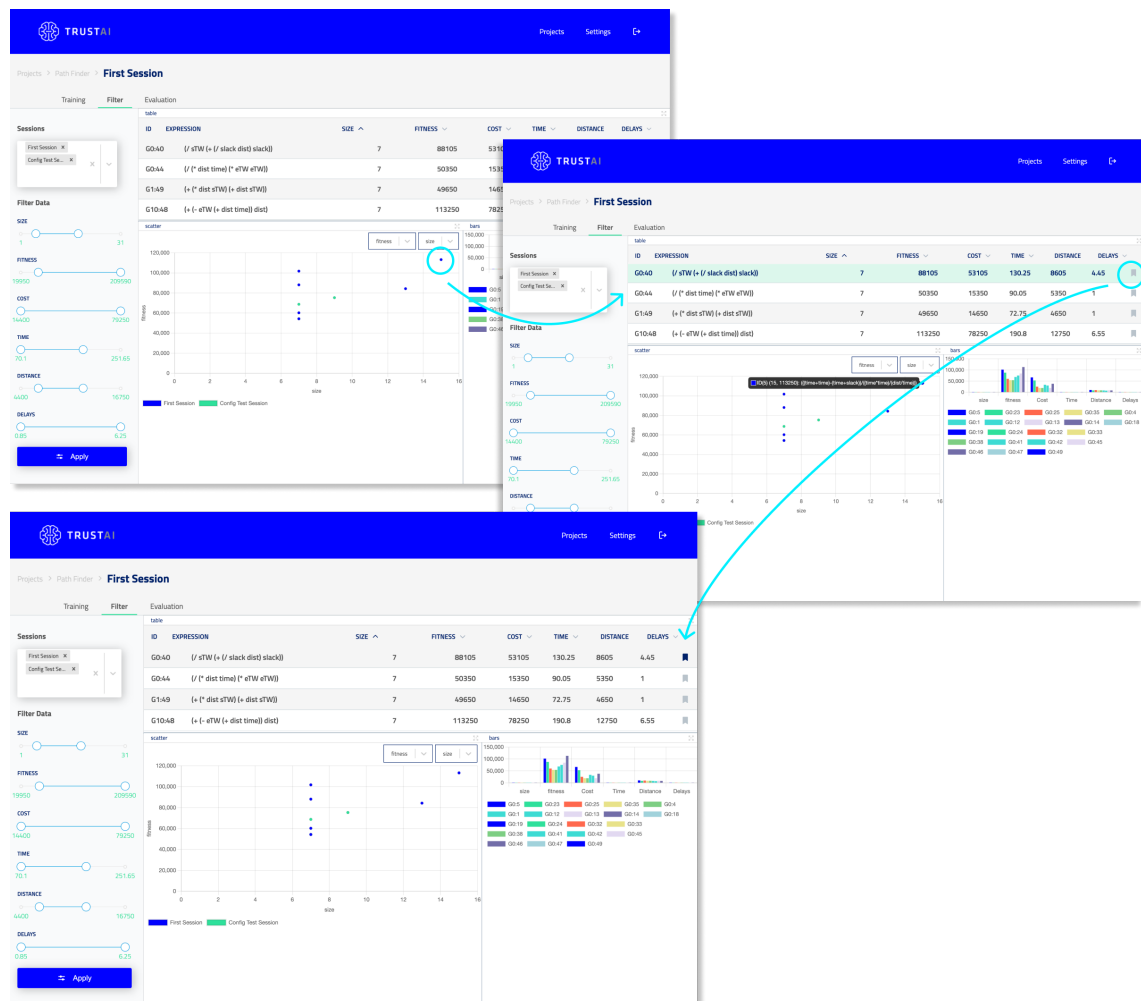


Figure 5.17: Highlight &amp; Save Solution storyboard

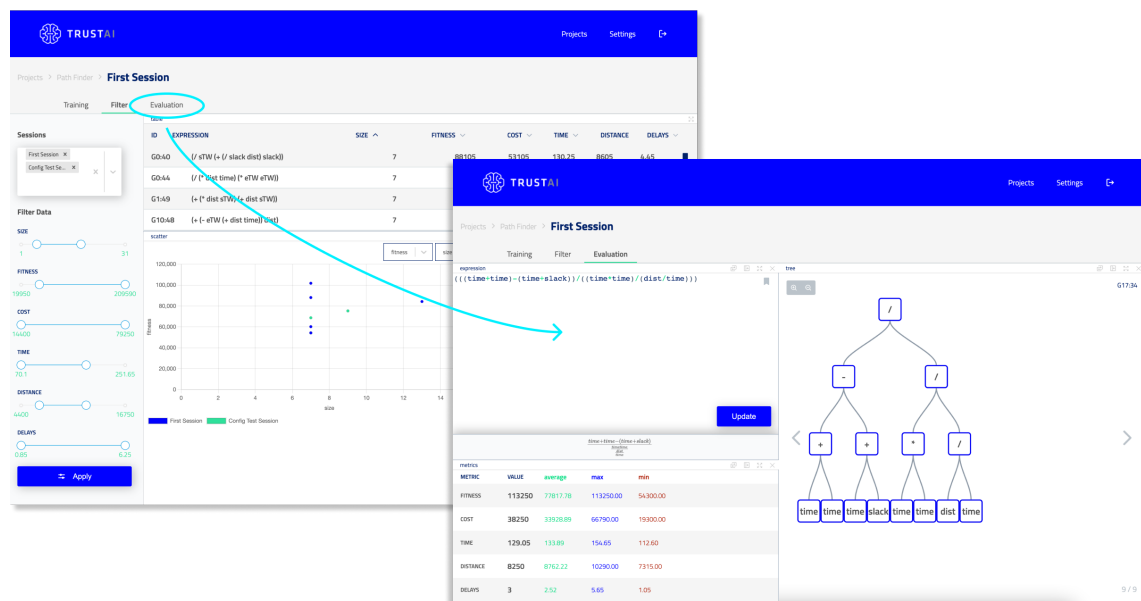


Figure 5.18: Evaluate Solution storyboard

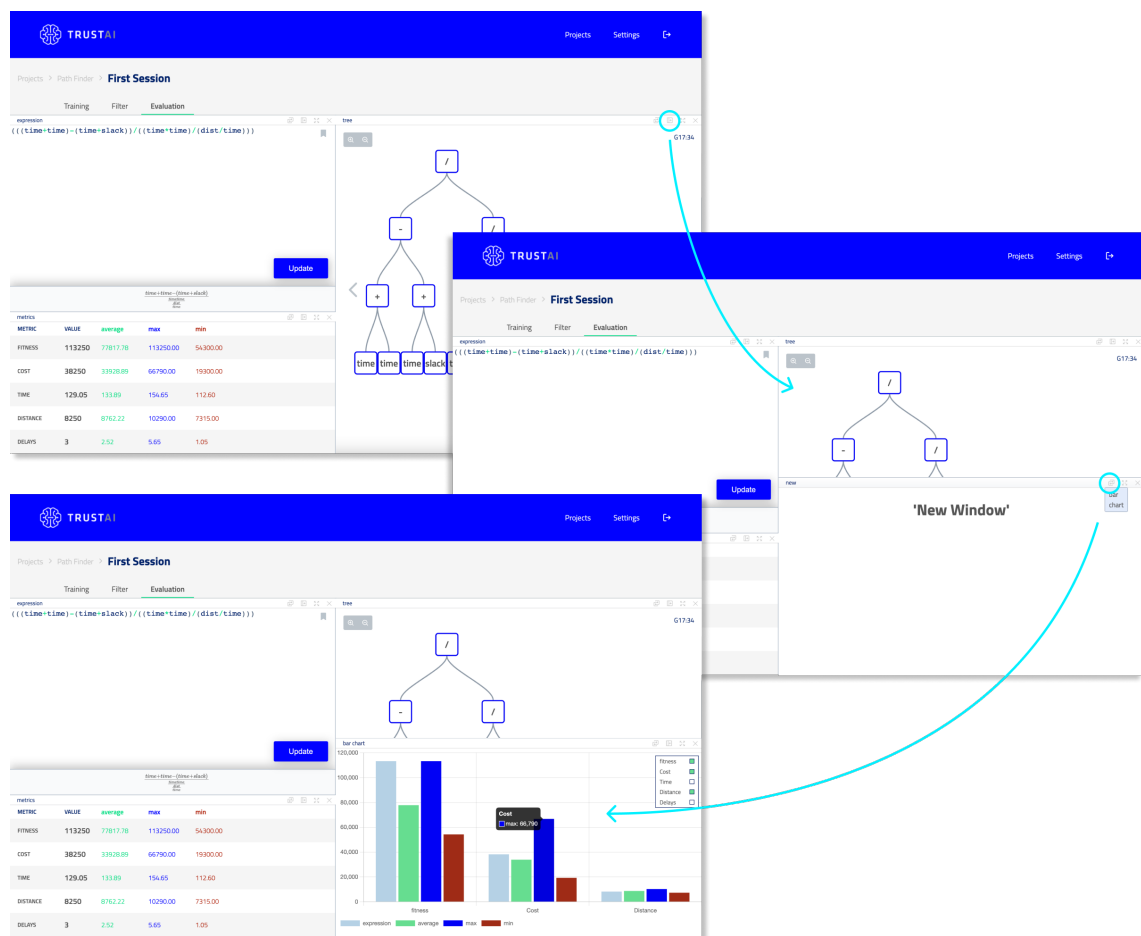


Figure 5.19: Rearrange Screen storyboard - Add New Window

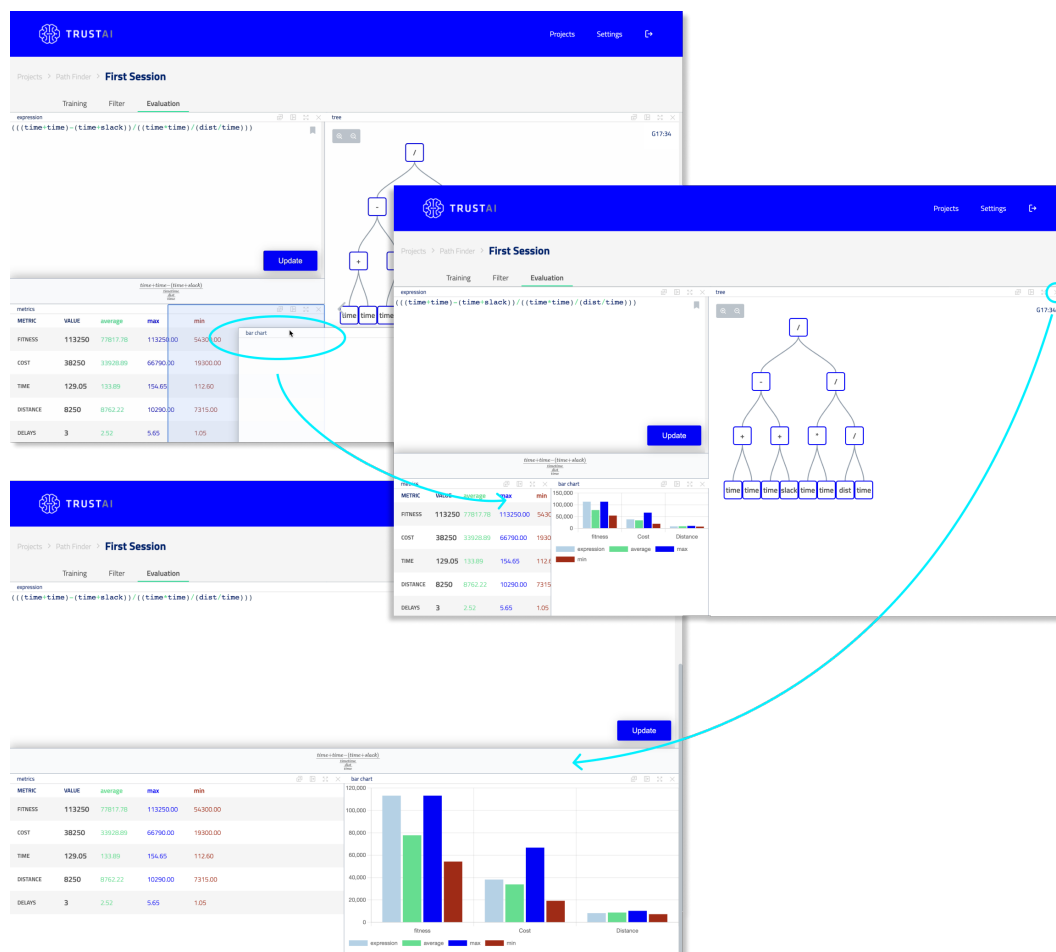


Figure 5.20: Rearrange Screen storyboard - Drag &amp; Close Window



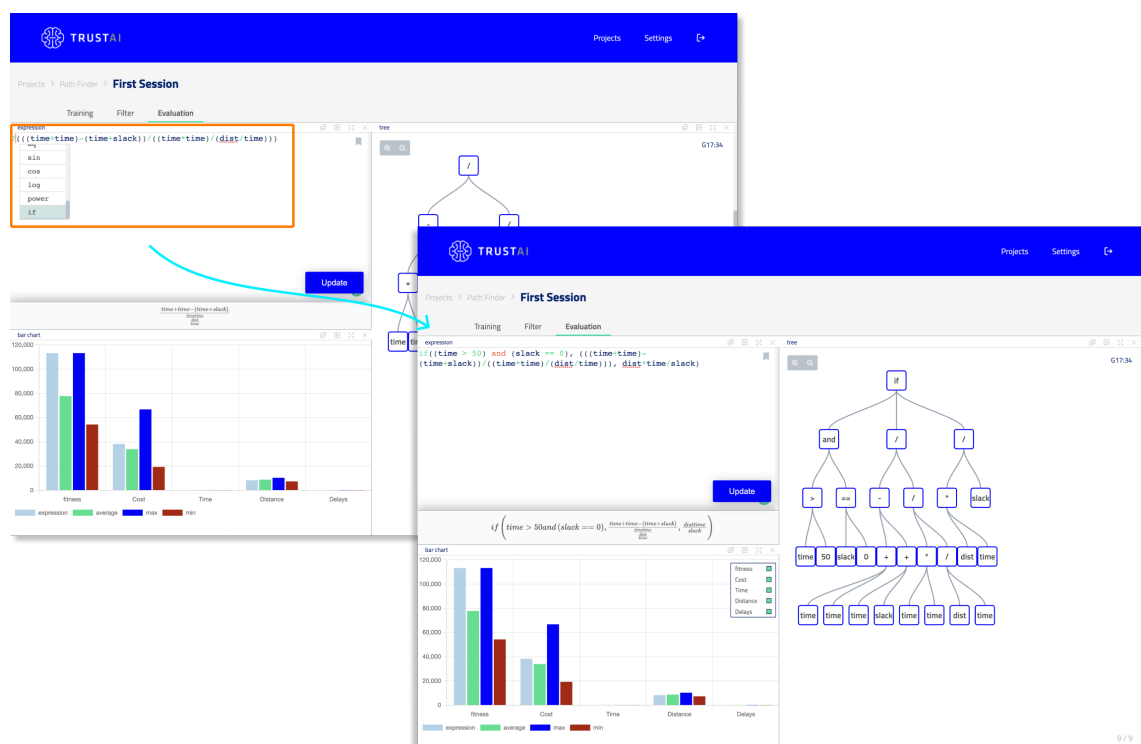


Figure 5.21: Edit Expression storyboard

## 5.4 Architecture and Implementation

As shown in the previous sections, we were able to implement a client that aims at enabling the defined **Use Cases** while abiding by the **Non-functional Requirements**. While designing the architecture of this client, we also made sure to support three different needs that contribute to a smoother development and result in a more robust application:

- **Testability** - the extent to which the quality of any module, requirement, subsystem, or other architecture components can be tested;
- **Flexibility** - the ability of a software system to change without undergoing structural modifications in response to changes in the environment and user requirements;
- **Maintainability** - the ease with which a software system can be repaired, enhanced, and comprehensible during its development cycle.

To achieve this, we mainly focused on the following two client-side design principles:

- **Command Query Separation** - each operation is considered either a *command* - changes the state of a model; or a *query* - returns data but does not change state. The main advantage of this pattern is that it makes reasoning about code a much simpler process. Separating all features as *commands* and *queries* and organizing them together also makes it easier to test our code and re-utilize the testing code;
- **Separation of Concerns** - consciously establish logical separations between each of the system's architectural concerns. This improves visibility of the tasks that need to be completed, the layer they fall under, and the tools that may be applied to address those issues. This principle makes it easier to understand the code base, apply changes to it with minimal effort, and reuse modules and functionalities.

The resulting client's architecture followed a *Model View Presenter* (MVP) (Syromiatnikov and Weyns, 2014) structure, as seen in Figure 5.22. This is a layered architecture that can be decomposed in three different main components, which were implemented using *React*<sup>2</sup>:

- **View** - this layer holds the presentation components that are responsible for rendering the UI, using data received from the *Presenter* layer, and generating user events (e.g., key presses, button clicks, and hover states). Each presentation component has its own view behavior (*UI Logic*) (i.e., conditions that determine what information to show, when to show it, and which user events are being listened to). Each component has its own local state that is updated using the data from the *Presenter* and is used by the *UI Logic* to perform decisions;

---

<sup>2</sup>Found at <https://reactjs.org/>

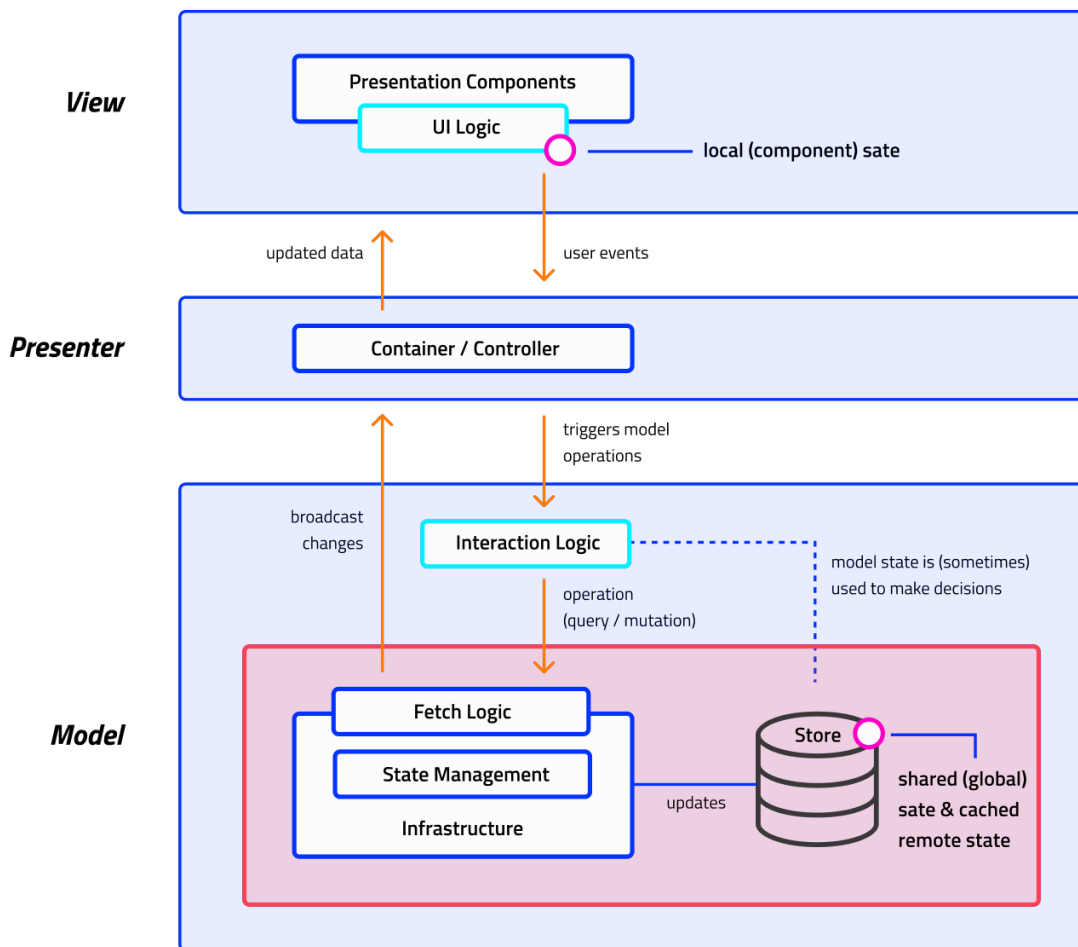


Figure 5.22: TRUST - Frontend High Level Architecture

- **Presenter** - this layer has two primary responsibilities: i) consume user events that delegate operations to the *Model* layer; ii) subscribe to data changes of the *Model* layer and keep the view's data updated. Container components are the top-level modules (usually pages) with no functionality that are aware of a shared state and connect it to the presentational components that need it;
- **Model** - this component is responsible for storing data and acts as an interface for communication between the client and the backend server. The first layer of this module is the *Interaction Logic* layer which defines the behavior of the model: it decides when and which operations (query or mutation) are performed based on the information provided by the *Presenter* layer and information available on the *Model*'s store. The *Model* component also contains networking and data fetching functionalities (*Fetch Logic*) that know where the backend services are, perform API calls, and formulate responses that are broadcasted to the *Presenter* layer. Finally, the *State Management* component has three main responsibilities: i) it holds onto a global (shared) state that supports caching; ii) it uses the information from the backend to update the global state; iii) it provides the presentation components

a way to subscribe to data, which will be used to re-render those components when data is updated. Both the *Fetch Logic* and *State Management* are *Infrastructure* components and were implemented using the *React-Query*<sup>3</sup> library.

### 5.4.1 Technologies

When designing a complex platform like TRUST, it is essential to thoroughly consider which technologies are better suited to solve the problem at hand and when to implement things from scratch or use external libraries. Throughout the design phase, we considered various alternatives for what technologies to use and even tested different libraries to implement some features. Here are some of the leading technologies we have ended up using for the final prototype:

- **React** - frontend javascript framework that allows for a high development speed, fueled by its relatively low learning curve, a considerable amount of resources and guides online, and access to countless packages through NPM<sup>4</sup> that simplify implementing certain features, namely libraries that provide powerful charts and diagrams visualization tools;
- **Sass**<sup>5</sup> - which stands for “Syntactically awesome style sheets” is an extension of CSS that enables the usage of variables, nested rules, inline imports, and more. Overall Sass facilitates writing clean, easy, and less CSS in a programming construct also contributing to a higher development speed. It is more stable, powerful, and elegant because it is an extension of CSS. So, it is easy for designers and developers to work more efficiently and quickly;
- **React-Query** - is a library that implements *fetch logic* and *state management* capabilities and eases the process of caching, synchronizing and updating server state;
- **React-Mosaic**<sup>6</sup> - a fully functional React Tiling Window Manager designed to provide the user total control over their workspace. It offers a straightforward and adaptable API to tile any number of complex React components throughout a user’s display. This library is extremely important to give users the ability to rearrange the Training, Filter and Evaluation screens;
- **Chart.js**<sup>7</sup> - open-source *Javascript* data visualization library that supports various different chart types, which are responsive, interactive, configurable, and support user events. This library was essential to easily include line, bar, and scatter charts that are reactive to filters and allow the user to zoom in and out on the data, pan across the X and Y axis, and data to see additional information.

---

<sup>3</sup>Found at <https://react-query.tanstack.com/>

<sup>4</sup>Found at <https://www.npmjs.com/>

<sup>5</sup>Found at <https://sass-lang.com/>

<sup>6</sup>Found at <https://github.com/nomcopter/react-mosaic>

<sup>7</sup>Found at <https://www.chartjs.org/>

### 5.4.2 Implementation Details

Some design and implementation aspects were especially challenging and made the TRUST user interface stand out compared to other similar applications. In this section, we explain those challenges and the devised solutions to tackle them.

#### 5.4.2.1 Tile System

As seen in [Screens and Features](#), the [Training Section](#), [Filter Section](#), and [Evaluation Section](#) allow the user to rearrange the different windows (or tiles) that compose the section. For instance, looking at the example shown in [Rearrange Screen](#), we can see that it is possible to add new windows, close windows, resize windows, substitute a window with another window and drag them to restructure the screen.

These capabilities give the user total control over what it wants to see. Although we are currently supporting only a few different tiles, this design makes it so that new windows with new content can be easily added to the project in the future. This is extremely important for the project's growth, given the different user types and use-cases that it can consider in the future.

This solution was not only hard to be thought out, but it also required a challenging implementation, which was made easier with the support of the *React-Mosaic* library.

#### 5.4.2.2 Dynamic Mathematical Expressions Parser

The genetic programming algorithms used in the TRUST tool generate symbolic models that can contain constants, variables, mathematical operators, logical operators, and any type of function. An illustrative example of a symbolic expression generated through a GP algorithm is presented in [Figure 5.23](#).

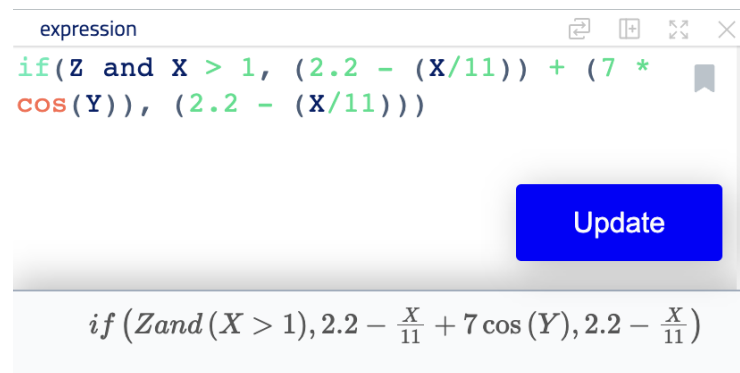


Figure 5.23: TRUST - Representation of a Genetic Programming Solution

To implement the Editing and Tree Visualization features, we were required to parse and validate these expressions. However, two main problems were identified:

1. Each algorithm has its own set of variables, operators, and functions;

2. Some genetic programs use a Lisp notation instead of infix notation (e.g., “5 \* 2 + 3” would be “(+ (\* 5 2) 3)”).

To address these problems, we first built a converter that converts Lisp notation to infix notation. This way, the user only needs to work with infix notation which is much more natural and intuitive when working with mathematical expressions. This solved problem number 2.

One option to solve problem number 1 would be to implement a different parser for each algorithm. However, this solution is not only highly inefficient but also not scalable as the tool aims to support any algorithm that can be uploaded by the user. The route we chose included implementing a parser that dynamically receives a grammar where the variables, operators, and functions are defined. This grammar can be defined in a JSON object that contains three properties:

- **operators** - an array with the available operators. Each operator is defined by an array whose first element is the text of the operator, and the second is an object which contains the children types and the returning type of the operator;
- **functions** - an array with the available functions. Each function is defined by an array whose first element is the name of the function, and the second is an object which contains the children types and the returning type of the function;
- **variables** - an array with the available variables. Each variable is defined by an array whose first element is the name of the variable, and the second is its type.

This format is flexible and allows the definition of custom grammars relatively easily, enabling the TRUST interface to support any type of problem easily. The following object represents an example of a simple grammar:

```

1 {
2   "operators": [
3     ["and", { "children": ["boolean", "boolean"], "type": "boolean" }],
4     ["or", { "children": ["boolean", "boolean"], "type": "boolean" }],
5     ["<", { "children": ["number", "number"], "type": "boolean" }],
6     [">", { "children": ["number", "number"], "type": "boolean" }],
7     ["!=", { "children": ["number", "number"], "type": "boolean" }],
8     ["==", { "children": ["number", "number"], "type": "boolean" }],
9     ["+", { "children": ["number", "number"], "type": "number" }],
10    ["-", { "children": ["number", "number"], "type": "number" }],
11  ],
12  "functions": [
13    ["sin", { "children": ["number"], "type": "number" }],
14    ["cos", { "children": ["number"], "type": "number" }],
15    ["log", { "children": ["number"], "type": "number" }],
16    ["if", { "children": ["boolean", "number", "number"], "type": "number" }],
17  ],
18  "variables": [
19    ["X", "number"],

```

```

20     ["Y", "number"],
21     ["Z", "boolean"]
22 ]
23 }

```

### 5.4.2.3 Expression Editor

As shown in the **Edit Expression** storyboard, the user can individually edit expressions on the evaluation screen. What allows users to make these changes is a small window that works like a text editor, with some highlighting features (see Figure 5.23). Here, the user can write any changes it wants and click on the “Update” button to validate and update the new expression.

In case the user is unsure of what operators, functions, and variables are, this editor supports auto-complete capabilities that aid the user with that problem. When the user writes the “?” character, a list of the operators and functions is shown, and the user can keep writing to filter through that list, as shown in Figure 5.24. The selected suggestion will automatically be written if the user presses the “tab” key. The same can be said for the available variables, which are shown when the user writes the “\$” char.

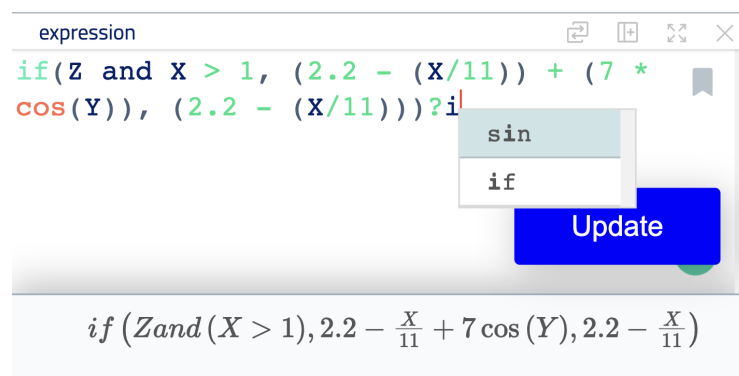


Figure 5.24: TRUST - Editor Operators & Functions List

If the updated expressions contain a parsing or syntax error, the token causing that error will be highlighted, and an informative error message will appear, as shown in Figure 5.25.

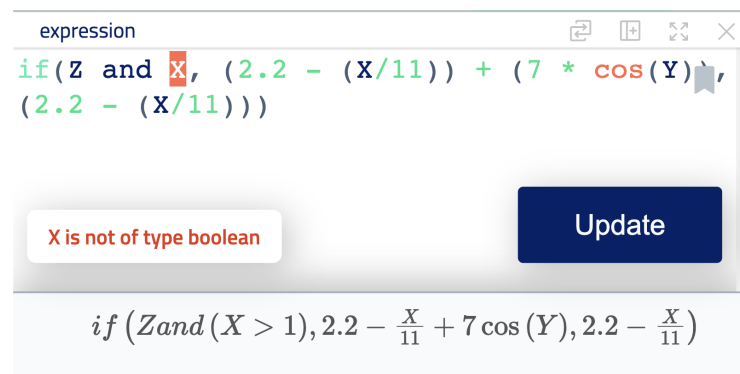


Figure 5.25: TRUST - Editor Error Message

## 5.5 Summary

The Interaction Design phase allowed us to define how information can be effectively communicated to the user. Our typical workflow for this phase was an iterative one but started with the design of different high-fidelity mock-ups for the application's screens. Those mock-ups would then be evaluated and validated, with the stakeholders, against the requirements. If they were not accepted, they would be refined using the gathered feedback. Otherwise, they would be implemented.

The implementation of the user interface followed an architecture that focused on supporting a smoother development experience by aiming toward testable, flexible, and maintainable code. The resulting architecture followed a Model View Presenter structure and was implemented using *React*.

Some design and implementation aspects were especially challenging and made the TRUST interface stand out compared to other similar applications, such as the ones mentioned in Section [Similar Platforms](#). This includes features such as the rearrangeable *Tile System*, the *Dynamic Mathematical Expressions Parser*, and the *Expression Editor*. These features are an important contribution to the process of finding adequate solutions with ease and play a vital role in a platform to support human-guided AI.



## Chapter 6

# Evaluation

The process of Evaluation is vital to assess the quality of the design. It allows us to verify if the user interface satisfies the proposed requirements and how well they match the user's context and needs. Evaluation is an iterative process, which involves extensive user testing and gathering feedback, that can be used to refine the design until the evaluation results are satisfactory.

This chapter describes the evaluation process of the developed user interface, which allowed us to assess the system's usability and verify if the functional requirements were met. This process was done partially in parallel to the development of the platform, across multiple meetings and evaluation sessions, detailed in Section 6.1. Section 6.2 describes the usability study performed during the Evaluation phase, where quantitative and qualitative data was captured and analyzed through a combination of testing, interviews, and questionnaires. Finally, Section 6.3 describes the workshop where multiple participants had the opportunity to learn more about the TRUST platform, test the developed prototype, and provide feedback regarding its usability.

### 6.1 Continuous Evaluation

As stated in Section **User-centered Design**, in order to guarantee that the user is at the core of the system, we must keep it involved through all phases of the design process and get their feedback and reviews continuously and iteratively, leading to a solution that meets the user requirements.

After eliciting the fundamental user requirements, various meetings were held, which mainly consisted of three different categories:

- weekly meetings with the supervisor and co-supervisor of the dissertations;
- biweekly meetings with all the collaborators from the INESC TEC team;
- biweekly meetings with one of the project partners, TAZI;
- monthly meetings with all the project partners.

All of these meetings let the different entities share and discuss the current state of their work, ideas, and plans for the solution. Regarding the development of the interface, the meetings served

as a means to present ideas, solutions, mockups, and prototypes iteratively. After that, what was presented was discussed and evaluated qualitatively by the participants, which included potential users of the platform and AI and GP experts.

This was an excellent opportunity to gather feedback from users with multidisciplinary skills and perspectives and apply principles discussed in the **Design Research Principles** Section. The collected information allowed us to understand users better based on experiences and impressions. This continuous assessment of the quality of the solution would then be used to refine and improve the design experience iteratively.

## 6.2 Usability Study

After the implementation of the user interface was completed, a usability study was performed as a way to do a final usability assessment of the prototype, helping identify weaknesses and strengths, validate the implementation of the functional requirements, evaluate their overall usability and contribute to the definition of a working plan following this dissertation.

During this study, the participants had the opportunity to test the interface and provide formal feedback, both qualitative and quantitative, through questionnaires and interviews.

### 6.2.1 Procedure

After building the possible participants list, we designed the overall usability study structure and planned its performance. Multiple forms of gathering feedback for the study were discussed, and we concluded that doing one-on-one online sessions would yield the best results. Although more time-consuming than some of the alternatives, it would allow us to get a more accurate description of the participant's thoughts and actions, resulting in a broader amount of feedback (both verbal and visual). This would also enable us to easily measure times, guide, and help the participant when needed, ensuring a smooth experience.

Taking this into account, we first started by preparing a document, presented in Appendix B, which contains some details of the study, such as context, goals, and requirements to participate, but also a small video demonstration of the platform (so that the participants can get familiarized with the application before testing it), a link to a document with the instructions and features of the application (seen in Appendix C), and, finally, a link for the participant to schedule its session, through *Doodle*<sup>1</sup>. This document would later be sent by email to the considered participants, inviting them to participate in the study.

After defining this document, we prepared the structure of the sessions and how they should be performed. For that, we prepared a *usability test guide* for the participants to read during the session and a *usability test script* to support the guidance performed by the interviewer. Both of these documents were tested and refined during a series of "Pilot Sessions" and then used for the "Main Sessions" after being finalized.

---

<sup>1</sup>Found at <https://doodle.com/>

### 6.2.1.1 Gathering Feedback

During the sessions, participants were asked to perform a set of tasks. When it comes to gathering qualitative feedback, we have asked users to be honest when giving their thoughts while performing those tasks and, if possible, try to verbalize them using the think-aloud protocol (Charters, 2010). The think-aloud protocol is a method used to gather data in usability testing in product design and development. It involves participants thinking aloud as they are performing a set of tasks. Users are asked to say whatever they are looking at, thinking, doing, and feeling as they do their tasks. This enables observers to see the process of task completion and collect as much feedback as possible. At the end of each session, we also encouraged participants to express feedback about the study or the platform itself, either verbally or textually and anonymously.

When it comes to quantitative feedback, we used a much more direct strategy, which included using questionnaires that required quantitative opinions from the participants. For each task, participants were asked to fill in a questionnaire based on the NASA Task Load Index, which is a tool for measuring and conducting a subjective mental workload (MWL) assessment. It allows the determination of the MWL of a participant while performing a task (Hoonakker et al., 2011). It rates performance across six dimensions to determine an overall workload rating. The six dimensions are as follows:

- **Mental demand** - how much thinking, deciding, or calculating was required to perform the task.
- **Physical demand** - the amount and intensity of physical activity required to complete the task.
- **Temporal demand** - the amount of time pressure involved in completing the task.
- **Effort** - how hard does the participant have to work to maintain their level of performance.
- **Performance** - the level of success in completing the task.
- **Frustration level** - how insecure, discouraged, or secure the participant felt during the task.

For each of the dimensions, the participant is asked a question to rate their score on an interval scale ranging from low (1) to high (10) (NASA-TLX Rating):

- How mentally demanding was the task?
- How physically demanding was the task?
- How hurried or rushed was the pace of the task?
- How successful were you in accomplishing what you were asked to do?
- How hard did you have to work to accomplish your level of performance?
- How insecure, discouraged, irritated, stressed, and annoyed were you?

The scores of these questions can then be used to provide a quick and straightforward estimation of each task's perceived workload.

After finishing the tasks, participants were asked to answer ten more questions corresponding to the System Usability Scale (SUS) questionnaire. SUS is a quick, cheap, and tested tool for reliably measuring the usability of a system (Reitz et al., 2021). This questionnaire allows scoring usability through a set of 10 items with one of five responses that range from "Strongly Agree" to "Strongly Disagree":

- I think that I would like to use this system frequently.
- I found the system unnecessarily complex.
- I thought the system was easy to use.
- I think that I would need the support of a technical person to be able to use this system.
- I found the various functions in this system were well integrated.
- I thought there was too much inconsistency in this system.
- I would imagine that most people would learn to use this system very quickly.
- I found the system very cumbersome to use.
- I felt very confident using the system.
- I needed to learn a lot of things before I could get going with this system.

The answers are then converted to a new number using a predefined formula and result in a 0 to 100 score (SUS score). According to this method, a SUS score above 68 would be considered above average (above the 50th percentile), and anything below 68 is below average. As seen in Table 6.1, the SUS scores can also be interpreted, not only as percentiles but also as grades or adjectives, as Jeff Sauro recommends (Sauro, 2018):

### 6.2.2 Pilot Sessions

The pilot sessions were performed in order to validate the overall evaluation study setup and ensure that all the necessary steps were being performed for a successful study.

The first version of the *usability test guide*, which is present in Appendix E, contained the study details, such as context, goals, and the steps to be performed by the participants. It also links to a video demonstration of the application to help participants gain a general understanding of the application, and a document with the instructions and features of the platform, presented in Appendix C, which contains the website's sitemap and its main features and capabilities.

Table 6.1: SUS Scores Interpretation

Score	Percentile	Grade	Adjective
A+	84.1-100	96-100	Best Imaginable
A	80.8-84.0	90-95	Excellent
A-	78.9-80.7	85-89	
B+	77.2-78.8	80-84	
B	74.1 – 77.1	70 – 79	
B-	72.6 – 74.0	65 – 69	
C+	71.1 – 72.5	60 – 64	Good
C	65.0 – 71.0	41 – 59	
C-	62.7 – 64.9	35 – 40	
D	51.7 – 62.6	15 – 34	OK
F	25.1 – 51.6	2 – 14	Poor
F	0 - 25	0 - 1.9	Worst Imaginable

### 6.2.2.1 Participants

We conducted six pilot sessions, each taking about one hour. We focused on recruiting participants of different backgrounds to get a better general idea of the quality of the study. To better characterize the participants demographically, we have asked them to fill a questionnaire with the following questions:

- What is your age?
- What is your gender?
- What is your academic background (e.g., computer science, industrial engineering)?
- What is your academic level (e.g., bachelor's, master's, Ph.D)?
- How comfortable do you feel using Artificial Intelligence Algorithms and Techniques?
- How comfortable do you feel using eXplainable Artificial Intelligence Algorithms and Techniques?
- How comfortable do you feel using Genetic Programming Algorithms?
- For how many months (full-time) have you used GP Algorithms?
- In a few paragraphs, describe your experience on the topics above (e.g., projects, work experience, etc.)

The results of this questionnaire, which can be seen in Appendix F, show that the participants' age group is relatively young (all of them had between 22 and 44 years old) and with high levels of education (two bachelors, one master's degree, and three PhDs). Three participants studied computer science-related fields, two studied industrial engineering, and one studied marketing.

Most participants had some knowledge about AI but not so much about XAI and GP. In fact, only two participants had experience using GP algorithms (10 and 3 months).

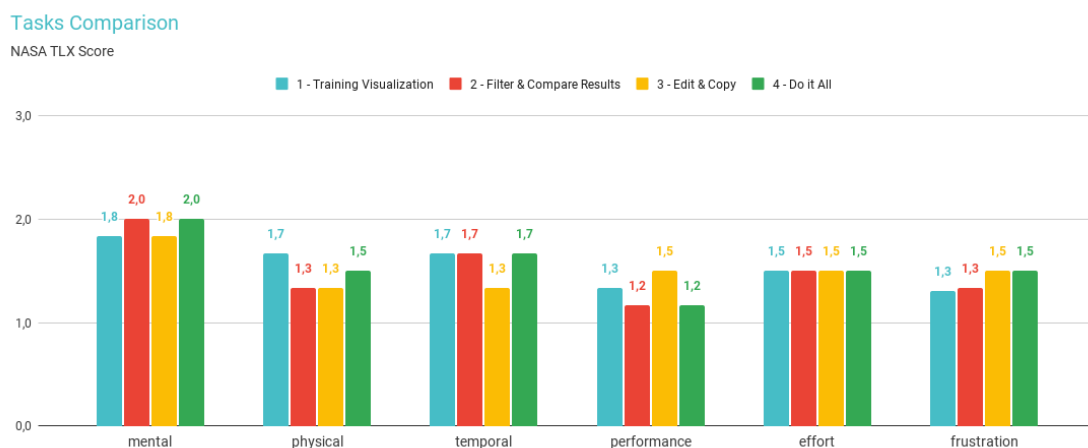
### 6.2.2.2 Tasks

Four different tasks were designed for the participants to perform. The goal of the tasks' design was for them to involve exercising the most important features and **Storyboards** of the platform:

- **Task 1 - Training Visualization** - "In this task, we want you to run a session and visualize the training evolution, using the dataset "locations\_1.csv" and the algorithm "ECJ TSP" (prescriptive problem). For that, you'll have to do some preliminary steps such as logging onto your account and creating a project."
- **Task 2 - Filter & Compare Results** - "in this task, we want you to use the results from the previous task and compare them to the session named "Comparison Session". Can you tell which session resulted in an expression with the best fitness? What about cost? Try to point out those expressions and make use of the filters to get there faster."
- **Task 3 - Edit & Copy** - "find the tree for the expression with the best fitness and make some changes to it (e.g., add a constant or a variable at the end). Bookmark the expression and copy its Latex format to your clipboard."
- **Task 4 - Do it All** - "using the "patients.csv" and the algorithm "EASimple" (predictive problem), find the best expression according to fitness and the best according to cost."

### 6.2.2.3 Results

The complete results of the questionnaire can be seen in Appendix F. Figure 6.1 shows the calculated NASA TLX Scores for each of the performed tasks and their respective dimensions.



NASA TLX Scores range from 1 to 10 (lower is better)

Figure 6.1: Pilot Sessions - Tasks NASA TLX Scores Comparisons

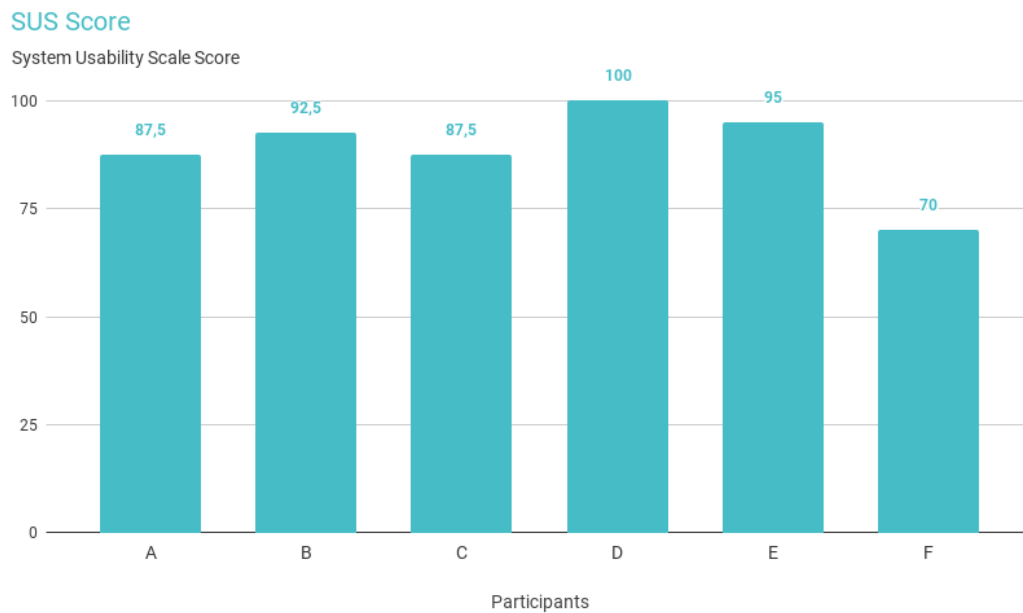


Figure 6.2: Pilot Sessions - SUS Score

In Figure 6.2 we can observe the SUS scores, resulting from the SUS questionnaire, that were calculated for each participant.

Besides the quantitative feedback, some participants also left some additional comments and critiques. Some of these were related to minor details and were addressed before the main sessions. Others were focused on the overall structure of the session (which helped us restructure it) and the difficulty of the tasks which, in general, was said not to be challenging enough.

#### 6.2.2.4 Discussion

When looking at the NASA TLX scores shown in Figure 6.1 it is possible to observe that, in general, the participants thought that all of the tasks required a considerably low amount of workload for each of the six considered dimensions. This can signify the platform's capacity to make the user perform its intended tasks with ease, but it can also mean that the proposed tasks were too simplistic.

We can also verify that all of the tasks were more mentally demanding than physically and temporally, which was to be expected since all of the tasks require a certain amount of thinking, but not so much physical activity (besides moving a mouse and writing on a keyboard) and no time pressure was introduced, although the sessions were somewhat lengthy (around 60 minutes each).

Regarding performance, and considering that in that question, 0 means the task was accomplished perfectly while 10 means failure, the results were pretty satisfactory since all of the participants were able to perform the tasks with success. The required effort and frustration levels also

remained considerably low, which can be a good indicator of a sound system's usability. We can also verify that the ratings of the six dimensions stood similar across the different tasks, with no particular highlight between them.

The SUS Scores represented in Figure 6.2 resulted in an average of 88.75, which, as seen in Table 6.1, is high above the 50th percentile score (68). In fact, a score above 84.1 is considered an A+ or "Best Imaginable" since it is above the 96th percentile. Although this is an indication of an excellent system's usability, this score was obtained from the pilot sessions, which only included six participants, so the results are not necessarily representative of the reality.

### 6.2.2.5 Conclusions

The pilot sessions' results were mainly positive when it came to assessing the workload of the main use-cases of the platform and its overall usability. However, as expected, interviewing only six participants is a severe limitation when it comes to generalizing the results.

The sessions provided the opportunity to validate the reliability of the study, the methods to gather metrics, the questionnaires, the usability session guide, and the moderator's interview script. In addition, they helped practice the interaction between the moderator and participants.

We were also able to collect feedback and critiques regarding both the study and the platform itself. This allowed us to iterate and refine the study and led to a few minor changes and additions to the interface.

The analysis of the pilot sessions pointed out that the tasks to be performed by the study participants were too simplistic. Thus, before the main sessions, they were rethought and revamped to become more challenging and utilize and exercise more of the platform's features.

## 6.2.3 Main Sessions

After analyzing the results from the pilot sessions and assessing the findings and gathered feedback, we were ready to start performing the main study. One with a much more robust structure, more challenging tasks, a more diverse set of participants, and a smoother interaction between them and the moderator.

### 6.2.3.1 Participants

This time, we conducted sessions with 12 participants recruited from people related to the TRUST-AI that represented potential future users of the platform but had no previous knowledge about it. We have applied the same questionnaire as in the pilot studies.

The full results of this questionnaire can be seen in Appendix H. Of 12 participants, 3 (25%) were female and 9 (75%) male and followed the age distribution seen in Figure 6.3.

As for their academic backgrounds, we have inquired 8 (67%) participants with a master's degree and 4 (33%) with a Ph.D. (Figure 6.5). As shown in Figure 6.4 most (41.7%) have studied Industrial Engineering, but other fields were present too, such as Electronics Engineering, Mathematics, Mechanical Engineering and Computer Science.



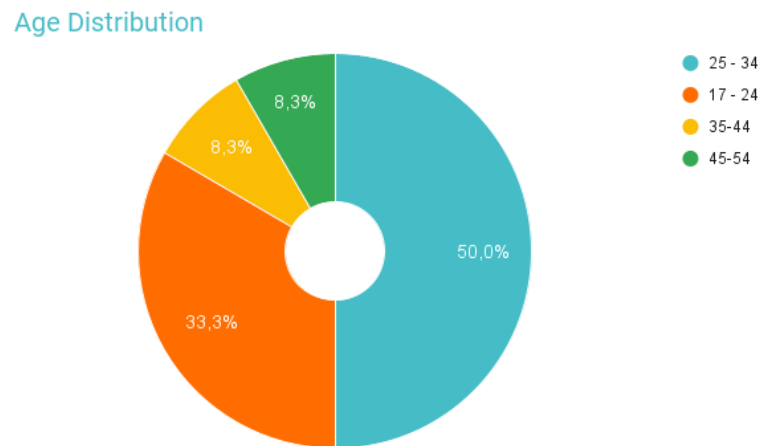


Figure 6.3: Participants Characterization - Age Distribution

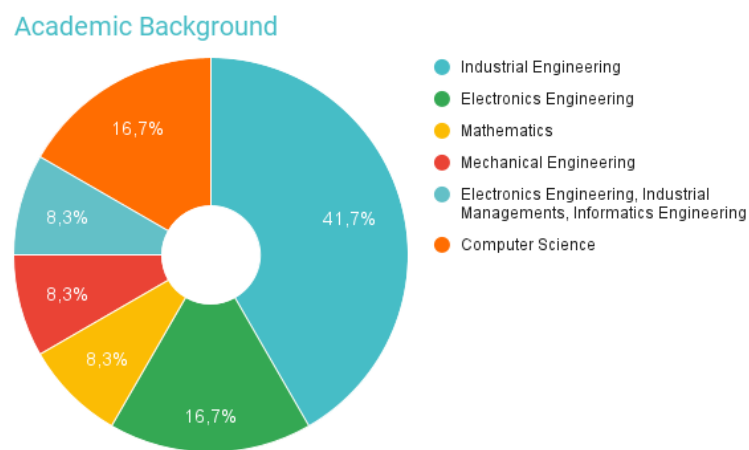


Figure 6.4: Participants Characterization - Academic Background

We made an effort to assess how comfortable the participants were with the topics of Artificial Intelligence, eXplainable Artificial Intelligence, and Genetic Programming. We can see from Figure 6.6 that all participants were at least moderately comfortable using AI algorithms and techniques.

As shown in Figure 6.7, the situation was similar for XAI algorithms and techniques, except for two participants (one felt uncomfortable, and the other very uncomfortable).

Regarding Genetic Programming, which is likely the most well-related field to the TRUST platform, two participants felt uncomfortable with it, five felt comfortable, and three were very comfortable (Figure 6.8). In fact, as shown in Figure 6.9, most participants had at least a few months of experience, with a median of 4.5 months.

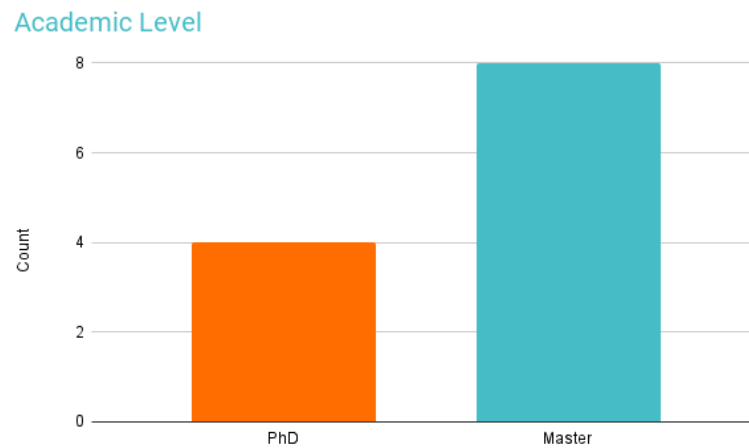


Figure 6.5: Participants Characterization - Academic Level

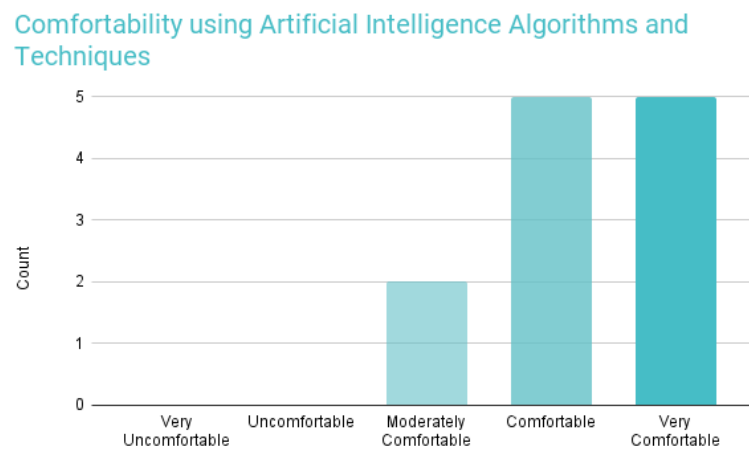


Figure 6.6: Participants Characterization - Comfortability using Artificial Intelligence Algorithms and Techniques

### 6.2.3.2 Procedure

Prior to the actual sessions, we invited 15 potential participants through email. This email served not only as an invitation but also contained the following information:

- Context of the study;
- Brief description of the platform;
- Document with a more detailed explanation of the platform's capabilities (see Annex B);
- Short video demonstration of the platform;
- Brief explanation of the session;

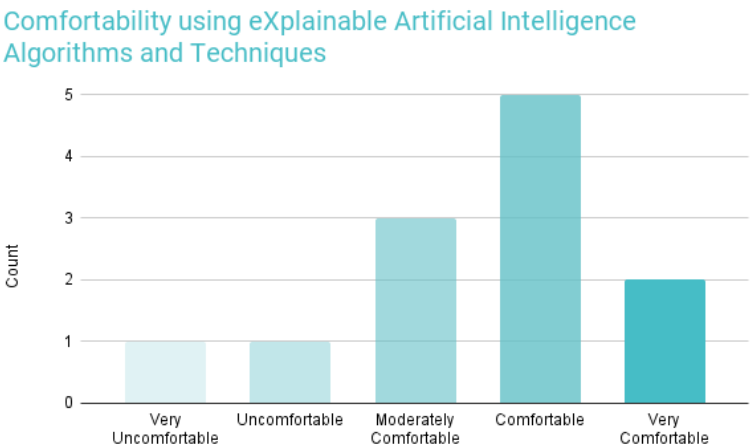


Figure 6.7: Participants Characterization - Comfortability using eXplainable Artificial Intelligence Algorithms and Techniques

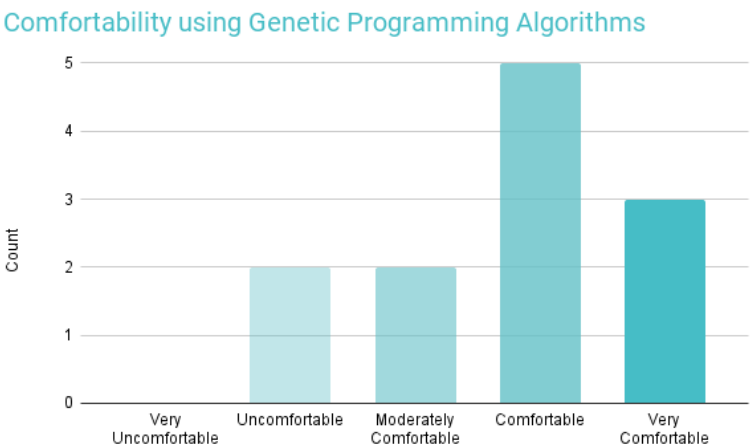


Figure 6.8: Participants Characterization - Comfortability using Genetic Programming Algorithms

- [Link to schedule the session.](#)

The sessions took place during the months of May and June and followed a moderator’s interview script, which is present in Annex D, to ensure reliability and consistency across the various sessions. Each session consisted of a structured interview that can be decomposed into different phases and interactions between the moderator and participant, as shown in Figure 6.10.

The sessions started with the moderator giving a brief introduction and elucidation on the context of the study being performed, followed by an explanation of the session that was accompanied by the sharing of the session guide (see Annex G). The participant would then start reading the document, ask any questions if needed, and consent to continue the session.

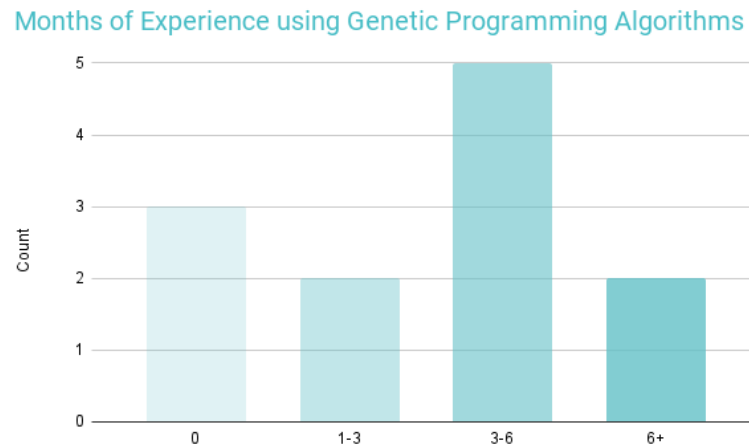


Figure 6.9: Participants Characterization - Months of Experience using GP Algorithms

Then, the moderator starts sharing his screen and gives a small demonstration of the platform. This demonstration includes displaying the main sections and features of the application, especially the ones necessary for performing the tasks.

After the demo, the participant is asked to share his screen, start reading the tasks and perform them. As in the pilot sessions, the moderator also invites the users to verbalize their thoughts as much as possible.

While the participants perform the tasks, the moderator is responsible for taking notes related to the strategy participants use to perform the tasks and any comments they make. At this stage, the moderator also measures the times needed to complete the tasks and guides the participant.

After finishing the four tasks, the participant fills the form as in the pilot sessions: a NASA TLX questionnaire for each of the tasks to assess their workload and a SUS questionnaire to assess the overall usability of the user interface. Finally, the moderator closes the session and writes down the participant's email in case he wants the study's results to be shared with him.

### 6.2.3.3 Tasks

This time, only three tasks were considered for evaluation since the first was considered a setup task as it only required trivial actions by the user:

- **Task 0 (Setup) - Training Visualization** - "In this task, we want you to start by creating a project. Then you should create and run a session and visualize the training evolution, using the Dataset "locations\_1.csv" and the algorithm "TSP" (prescriptive problem) with the config "TSP Config 1".
  - **Exercises Storyboards:** [Project Creation](#), [Session Setup](#), and [Training](#).
- **Task 1 - Filter & Compare Results** - "In this task, we want you to use the results from the previous task and compare them to the session named "First Session". After that:

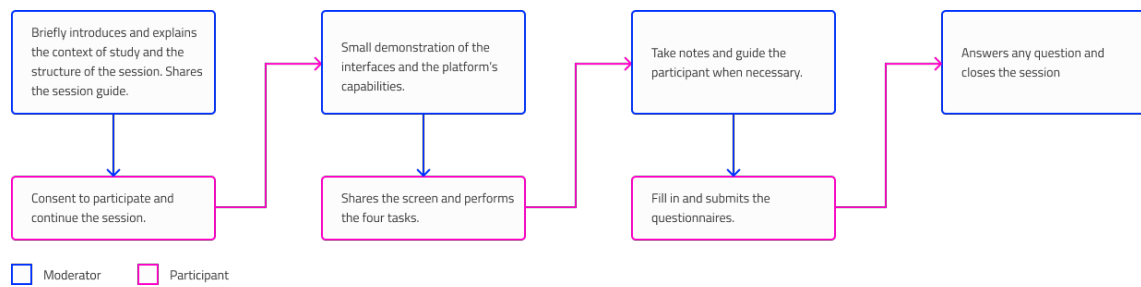


Figure 6.10: Interview Structure

- bookmark the expression with the best (highest) fitness;
  - using the scatterplot, highlight the solution with the least cost;
  - bookmark the expression with the cost equal to 53105;
  - see how many expressions need more than 215s (time);
  - see how many expressions have a size higher than 21, a cost under 58000 and a distance under 8700, and bookmark the one with the lowest delay.
  - using the bar chart, compare only the fitness and cost of the filtered solutions.";
  - **Exercises Storyboards:** **Filter Solutions**, **Compare Sessions**, and **Highlight and Save Solution**.
- **Task 2 - Edit & Evaluate** - "In this task we want you to find some expressions, make a few changes to one of them and see the results.
    - find the tree for the expression with the highest fitness;
    - make some changes to it (e.g., add a constant or a variable at the end);
    - copy its text to your clipboard;
    - find the tree for one of the expressions with the lowest cost;
    - using the if operator change the expression so that if the time is higher than 50, the expression with the highest fitness should be used, otherwise, the one with the lowest cost). Note that the if operator has the following format *if(condition, expression\_if\_condition\_equals\_true, expression\_if\_condition\_false)*;
    - see what is the height of the resulting tree and the number of children of the first node;
    - **Exercises Storyboards:** **Filter Solutions**, **Evaluate Solution**, and **Edit Expression**.
  - **Task 3 - Do it All** - Let us create a project to predict the possibility of patients suffering from strokes (predictive problem). Using the Dataset "patients.csv" and the algorithm "EAS", create a new project.
    - between the expressions with the highest fitness, find the one with the smallest size and bookmark it;

- find the expressions with a size lower than 22.8;
- identify the expressions that are non-dominated (Pareto efficient) and bookmark them. A solution is non-dominated if none of the objective functions (metrics) can be improved in value without degrading some of the other objective values. Beware in this problem we want to maximize the fitness in this problem;
- using any expression with size 13, change it (using the If operator) to stay the same, if the age is higher than 20 and the value isParent is True using the AND Operator, otherwise the expression should be  $\text{bmi} * \text{gender}$ . Note that the and operator has the following format *(condition\_1) AND (condition\_2)*;
- see what is the height of the resulting tree and the number of operators;
- **Exercises Storyboards:** [Filter Solutions](#), [Compare Sessions](#), [Highlight and Save Solution](#), [Evaluate Solution](#), [Edit Expression](#).

#### 6.2.3.4 Results

The complete results of this questionnaire can be seen in Appendix [H](#). For the main sessions, we have also calculated the NASA TLX Scores of each task and their respective dimensions, which can be seen in Figures [6.11](#), [6.12](#), and [6.13](#).

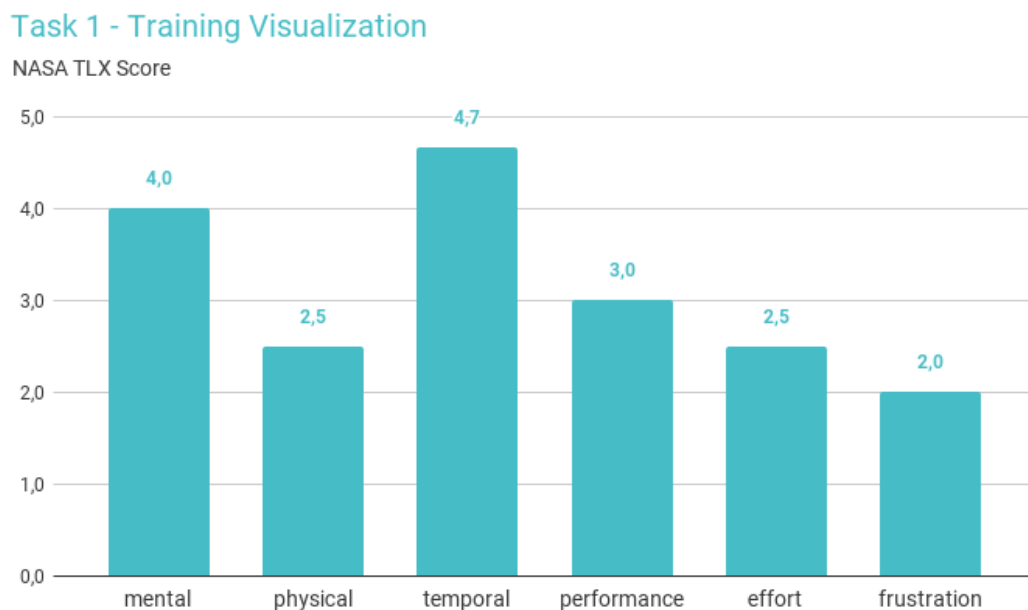


Figure 6.11: Main Session - Task 1 (Training Visualization) TLX Scores

To ease the comparison between tasks, we can also see those scores compared side by side in Figure [6.14](#). Besides, the average time that each task took to be performed is present in Table [6.2](#).

### Task 2 - Filter & Evaluate

NASA TLX Score

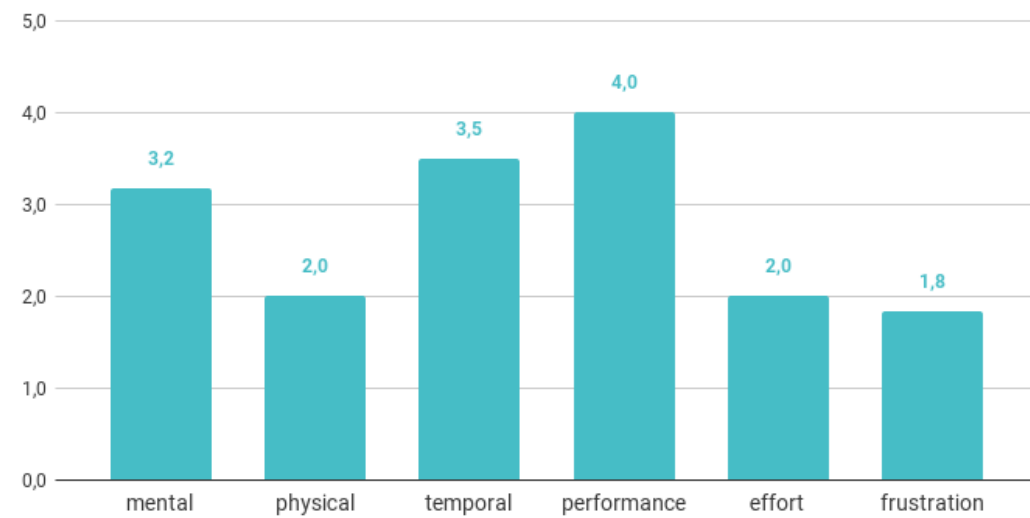


Figure 6.12: Main Session - Task 2 (Edit & Evaluate) TLX Scores

### Task 3 - Do It All

NASA TLX Score

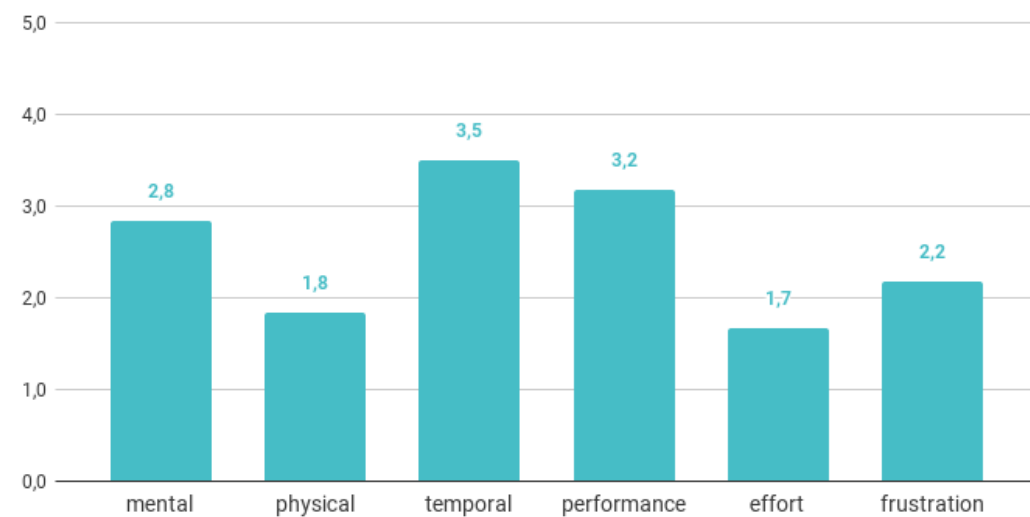
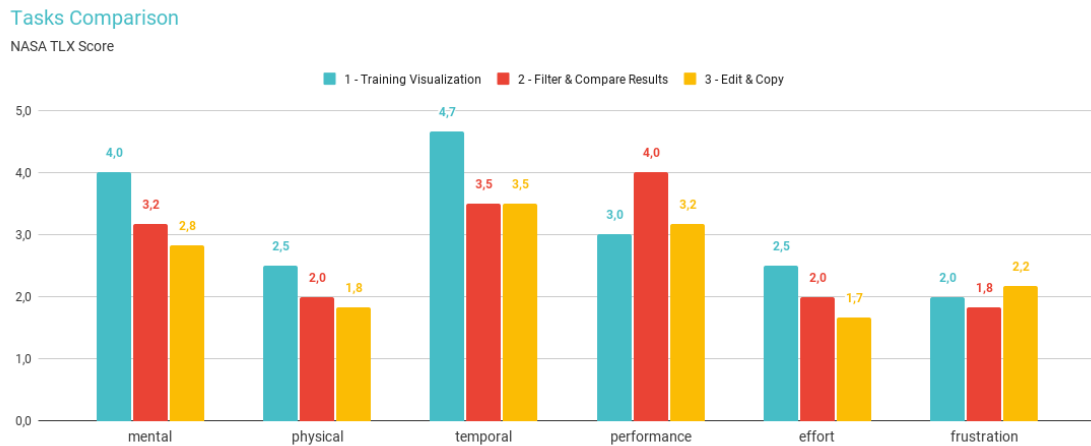


Figure 6.13: Main Session - Task 3 (Do It All) TLX Scores

Table 6.3 contains the summary of the SUS questionnaire results, individually for each question. The overall SUS Score of the interface was 82.1.

We were able to gather qualitative feedback from the questionnaire the users filled out and



NASA TLX Scores range from 1 to 10 (lower is better)

Figure 6.14: Main Session - Tasks NASA TLX Scores Comparisons

Table 6.2: Main Sessions - Tasks Average Time Comparison

	<i>Time (s)</i>	<i>TLX Score</i>
<i>Task 1</i>	<b>7.25</b>	3.11
<i>Task 2</i>	<b>8.14</b>	2.75
<i>Task 3</i>	<b>8.45</b>	2.53

verbal feedback that the users gave during the sessions. Here are some of the key comments about the overall usability of the system:

- “Great user experience, intuitive and simple! Very nice work on reducing the complexity over such a complex topic.”
- “In general, I believe the platform is very well made and contains many interesting functionalities for my workflow.”
- “The interfaces were clean, intuitive, and allowed an efficient and fast flow to find interesting solutions after training an algorithm. Being able to rearrange the sections and add/remove windows on the Evaluation section is a huge plus.”
- “I imagine that most people would learn to use this system very quickly, but these people should have some computer experience and some theoretical knowledge. A user with a high-school diploma and low technical skills will have to take more time. I had to get familiar with the syntax of the GP expressions, and then everything should be easy.”
- “A demo was given prior to performing the tasks. Therefore, it is a bit hard to judge how intuitive things really are for someone who has never seen the UI before. At the same



Table 6.3: Main Sessions - Summary of SUS questionnaire results

SUS Item	Average	Median	SD	Minimum	Maximum
1. I think that I would like to use this system frequently.	4,17	4,00	0,94	2,00	5,00
2. I found the system unnecessarily complex.	1,25	1,00	0,45	1,00	2,00
3. I thought the system was easy to use.	4,50	4,50	0,52	4,00	5,00
4. I think that I would need the support of a technical person to be able to use this system.	1,50	1,00	0,67	1,00	3,00
5. I found the various functions in this system were well integrated.	3,92	4,00	0,79	2,00	5,00
6. I thought there was too much inconsistency in this system.	1,67	1,50	0,89	1,00	4,00
7. I would imagine that most people would learn to use this system very quickly.	3,75	4,00	1,22	1,00	5,00
8. I found the system very cumbersome to use.	1,42	1,00	0,51	1,00	2,00
9. I felt very confident using the system.	4,25	4,00	0,75	3,00	5,00
10. I needed to learn a lot of things before I could get going with this system.	1,92	2,00	0,90	1,00	4,00

Answers range from 1 to 5.

SUS Items 2, 4, 6, 8, 10 are negatively worded: lower represents higher perceived satisfaction.

SD - Standard Deviation

time, it makes sense that instructions are given prior to using this UI, because it is a highly specialized one and is meant for technical people.”

Most users also made sure to mention some features or enhancements that they think are relevant for the future. Here follows a list of the most common ideas mentioned throughout the sessions:

- Identify expressions’ session in the Filter table so that we know which training provided that expression;
- Show expression identifier on the Evaluation page;
- Align metrics on the tables to the right for easier comparison;
- Be able to refer to expressions by ID in the Evaluation screen (when editing expressions);
- Add more highlighting features in the expression editor in the Evaluation screen (such as having different pairs of parenthesis with different colors);

- Be more consistent with all the buttons by always combining an icon and text inside it;
- Search for specific values for metrics in the Filter table;
- Save status of a session (i.e., sessions being compared, filters applied, and ordering);
- Show more information about each expression in the evaluation screen, such as tree height, number of operators, and number of terminals;
- Add a navigation icon in the tables that are used for navigation (it is hard for the user to know that he can enter projects and sessions by clicking on their names);
- A button to reset all filters in the Filter section;
- A button to remove all checkboxes in the bar charts with metrics filters;
- Show the types (i.e., boolean or number) of terminals, operators, and functions (as well as the type of arguments they expect);
- A button that allows bookmarking all the filtered expressions at once;
- Finding non-dominated solutions automatically in the Filter screen.

#### 6.2.3.5 Discussion

When comparing the NASA TLX Scores of the main sessions and the pilot sessions, we can see that we successfully increased the workload of the tasks for the main sessions as their average NASA TLX Score was 2.80, while pilot sessions had an average of 1.50.

Similar to the pilot sessions, these tasks were more mentally demanding than physically, which was to be expected since all of the tasks require a certain amount of thinking but not so much physical activity. However, on average, the temporal demand was the highest of all dimensions. It is possible that what caused this is the fact that these tasks were more complex, and they also required more time from the participants (as seen in Table 6.2). Besides, knowing that these are the main sessions, participants might feel a different sense of pressure compared to the pilot sessions.

Regarding performance, we can see that not everyone was able to successfully perform all of the tasks, especially for Task 2 - Filter & Compare Results, which had a NASA TLX Score of 4.0 in the performance dimension. Finally, the effort and frustration levels remained considerably low, meaning that participants did not feel insecure and did not have to work too hard to accomplish the tasks, which can be a good indicator of a sound system's usability.

Looking at Table 6.3, we can see that overall, participants felt the TRUST platform performed well. The group average for the overall SUS score was 82.1, which is considered an "Excellent" rating based on standard SUS guidelines represented in Table 6.1. The majority of users had a favorable opinion about the interfaces in terms of how confident they felt using the system (Item 9) and ease of use (Items 3 and 7).

Regarding technical support (Item 4) and the need to learn before using the system (Item 10), the results were positive but still indicate that, at the moment, some training is necessary for the users to make the most out of the application. Although the users seem to think the system is simple (Item 2), they believe there are some inconsistencies, which were also denoted in the qualitative feedback.

The qualitative feedback supported the participants' perceived satisfaction with positive comments about the user experience and the relevance of a platform of this kind. Participants were also kind enough to provide some ideas for future features and enhancements that can be implemented.

#### **6.2.4 Threads to Validity**

The main goal of this study was to assess the workload of the TRUST platform's flows and the interface's usability. Multiple threats to validity exist, and they were analyzed and divided into four categories as proposed by Wholin et al. in "Experimentation in Software Engineering" (Wohlin et al., 2012).

##### **6.2.4.1 Internal Validity**

Internal validity is the degree to which one can be confident that the causal links established in a study cannot be explained by other factors (Wohlin et al., 2012).

One threat that could affect our confidence was the possibility of unintended learning throughout the tasks. Although Task 1 - Filter & Compare Results, and Task 2 - Edit & Evaluate consist of completely different sub-tasks, Task 3 - Do it All, is composed by sub-tasks similar to the ones in Task 1 and Task 2, albeit being for a predictive problem instead of a prescriptive one. This could mean that if participants perform the tasks in the order 1, 2, and 3, Task 3 could become more straightforward as participants have learned during tasks one and two. To neglect this effect, half of the participants started with the predictive problem (Task 3), and the other half with the prescriptive problem (Task 1 and Task 2).

##### **6.2.4.2 Conclusion Validity**

Conclusion validity is the degree to which conclusions we reach about relationships in our data are reasonable (Wohlin et al., 2012).

The relatively small sample size (12 participants) is one significant limitation of the study and heavily degrades our statistical power. However, we focused on selecting participants that make up a good representation of the population of possible future users and with relatively high heterogeneity in terms of its demographics. Having directly supervised the sessions also contributes to a higher confidence level in the obtained results.

#### 6.2.4.3 Construct Validity

Construct validity defines how well a test measures the concept it was designed to evaluate. It is crucial to establish the overall validity of a method (Wohlin et al., 2012).

To guarantee that we correctly measured what we wanted to measure, we made use of industry standards questionnaires such as the NASA TLX Index and the SUS questionnaires and asked participants to perform tasks that are typical workflows for our target user. Additionally, we have performed pilot sessions to assess the feasibility, reliability, and validity of the study, which helped us tweak and revise our study to make it even more accurate for testing our construct.

Although SUS allows us to know where our application stands against any other, we did not have the chance to directly compare the prototype with another similar tool during the study. This prevents us from concluding if the development prototype is better or worse than some of its direct competitors.

#### 6.2.4.4 External Validity

External validity is the extent to which one can generalize the findings of a study to other measures, settings, or groups (Wohlin et al., 2012).

Although most participants of the main sessions had a background in Engineering, as seen in Figure 6.4, and had at least a master's degree (Figure 6.5), we believe that most potential users of an application like this will have similar backgrounds and, for that reason, we think the sample is representative of the population.

Another threat is the tendency for participants to change their behaviors simply because they know they are being studied. Although it was impossible to neglect this effect completely, we did our best to ensure participants felt at ease and without pressure while performing their tasks.

Finally, the pre-test treatment interaction can also influence the results of the study. As explained in the Procedure Section, users had access to a small video demonstration and a document with the features of the application prior to the sessions. Furthermore, before beginning their tasks, the moderator would also perform a small demonstration of the platform during the sessions. This is something that may alter the study results as participants get familiar with the interfaces before using them. However, every participant had access to this information, and on a platform like this, it is expected that most users will require some form of training before using it.

#### 6.2.5 Conclusions

In summary, the performed usability study showed promising results for the TRUST platform. The assessment of the interface resulted in a highly positive level of perceived satisfaction, having achieved a SUS Score of 82.1, which is undoubtedly a good indicator of a certain level of quality. However, there are still many improvements that can be made.

Seeing that the participants were able to perform the tasks that represented the storyboards of the application (listed in Section Storyboards) also made it possible to verify that the requirements (described in Section Functional Requirements) were met.

Finally, with the gathered feedback from the participants, it was possible to identify the current weaknesses and strengths of the interface, which contributed heavily to the definition of a development plan following this dissertation. This plan is described in Section [Future Work](#).

## 6.3 Workshop

Although not initially planned, on the 29th and 30th of June, INESC TEC held a physical conference with the different partners of the project. During this conference, we had the chance to perform a workshop where a presentation and demonstration about the TRUST platform were done. The presentation slides can be seen in [Appendix I](#).

The workshop consisted of a small presentation of the platform's main intended usages and the considered types of users for its development. This was followed by a demonstration where most features and workflows of the platform were showcased. Then, we had 24 conference attendees participate in an exercise.

All participants were direct contributors to the TRUST-AI project. In terms of academic background, most had a Ph.D. or a MS.c degree in fields such as Software Engineering, Industrial Engineering, Electronics Engineering, and Applied Mathematics. As part of the TRUST-AI project, most had experience with genetic programming (median of 10 months working with GP algorithms).

The exercise followed a similar structure as the one in the main session but was performed by all participants simultaneously, divided into teams of three. The exercise guide given to the participants is available in [Appendix J](#). All teams successfully performed the tasks and filled out a SUS questionnaire to assess the system's overall usability. The resulting SUS Score was 81, which is considered an "Excellent" rating based on standard SUS guidelines shown in [Table 6.1](#). This result supports the conclusions obtained from the usability study.

All participants also had the opportunity to provide additional comments about the platform. We have compiled the main ideas that were not already mentioned during the usability study into the following list:

- Allow editing of the generated solutions more graphically (e.g., manipulating the expression in its tree format);
- Show the progress bar of the training process and make its status more obvious to better understand the progress of a session;
- On the comparison section, highlight a model on the scatter plot when it is clicked on the table;
- Open the evaluation section for a specific model when selecting it on a table;
- Show tooltips to explain functionalities of the platform;
- Show average and standard deviation in the scatter plot;

- Allow user to scale axes separately in the scatter plot;
- Button to bookmark (and remove) all filtered options;
- Allow the direct comparison of two models.

## 6.4 Summary

The evaluation phase is one of the most critical phases to guarantee a user-centered design since it allows us to gather formal feedback from possible end-users that help validate the quality of the developed interface. Besides, it helps to identify the platform's weaknesses and strengths, which is critical to refining the design.

We performed a continuous evaluation throughout the implementation of the interface, which was done through several meetings with the stakeholders of the project and served as a way to assess the quality of solutions and improve the design experience iteratively.

A usability study was performed as a way to make a final assessment of the interface and validate its usability and functional requirements. This study consisted of a series of interviews where users had to perform a predefined set of tasks using the developed user interface and provide qualitative and quantitative feedback. Overall, the study's results were very satisfactory as they showed a highly positive level of perceived satisfaction and allowed us to identify potential future improvements and features that can be added to the interface.

Finally, a workshop with 24 participants was also held, confirming that the interfaces met the proposed requirements, helping to support the conclusions from the usability study, and allowing us to gather even more ideas for future work.

## Chapter 7

# Conclusions and Future Work

This last chapter presents our final concluding remarks in Section 7.1 and proposal for future work in Section 7.2.

### 7.1 Conclusions

Artificial Intelligence is a powerful concept that is more present than ever in our lives and can impact us daily. For that reason, there is an urgent need for AI systems to be understandable so that we can trust that they work as expected. XAI aims to achieve just that by combining social science and human-computer interaction techniques to produce explanations.

During this dissertation, we achieved our primary goal, which was to design and implement the user interface for an XAI web application that allows users to train genetic programs to solve classification, predictive and prescription problems. As seen in the **Interaction Design** chapter, with TRUST, users can compare solutions, find the most relevant ones, and evaluate and tweak them so that they can later apply those solutions to new data.

We believe that the interface developed can positively impact the XAI and interaction design community as they provide an example of a successful implementation of a system that allows a bidirectional human-machine interaction and enables human-guided explainable AI.

The development of the interface followed a user-centered methodology by keeping users actively involved in all phases of the design and using the techniques mentioned in the **User-centered Design** Section, which is most likely one of the most significant factors as to why we achieved such satisfactory results. We also made sure to fulfill the non-functional requirements and made an effort to architect the developed software to make it flexible, easily testable, and maintainable, as shown in the **Architecture and Implementation** Section.

We performed a thorough **Evaluation**, of the developed prototype, where a study was conducted and allowed the workload assessment of representative tasks and the overall usability of the developed interface. Generally, the results of this study were overwhelmingly positive and demonstrated a highly positive level of perceived satisfaction which is very promising for the

platform. However, not everything was positive. The SUS scores also demonstrated that the interface is lacking in the learnability dimension. Some of the qualitative feedback also pointed out interesting features that were missing or needed to be enhanced.

## 7.2 Future Work

It is important to note that the work developed did not intend to create a definitive interface to visualize and interact with genetic programs but rather to contribute with a possible solution that can serve as a basis for an ambitious project such as TRUST-AI. Thus, the result of this work may serve as a starting point for a broader study in terms of interaction design for human-guided AI.

In the **Problem Description** Section, we have mentioned the distinction between three types of users: Algorithm Expert, Model Developer, and Domain Expert. The interface developed during this dissertation was highly focused on enabling the Model Developer to train algorithms and interact with the solutions. As for the Domain Experts, currently, they would not be able to use the platform directly to help them make decisions, as TRUST does not yet support the usage of models on new data points, and no additional explanations are generated. Generating explanations and adapting explainability techniques and concepts to symbolic expressions (that can be represented by a tree) will be an exciting challenge since it is a topic where not much research has been done. As for the Algorithm Expert, it is also crucial to support the dynamic addition and configuration of new algorithms so that any type of problem can be solved with TRUST.

In future work, we should consider the suggestions given by the participants during the **Usability Study**. Most feedback consisted of simple modifications and additions that would most likely highly benefit the usability of the interface, especially the ones mentioned by more than one participant.

Regarding the implemented dynamic parser for symbolic learning expressions, described in the **Implementation Details** Section, we believe publishing it as *Javascript* library (e.g., using the *npm* package manager), would serve as a beneficial contribution to the XAI and Interaction Design communities, due to its potential for enabling interaction between humans and genetic programming models.

Finally, it is also expected that the application will be customized and validated within real-world use-cases such as the retail sector (dynamic time slot pricing) or the health sector (cancer treatment). This will be a challenging task that will put TRUST to the test and help validate the interface in a professional scenario.



# References

- Explainable explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai. *Information Fusion*, 58:82–115, 6 2020. ISSN 15662535. doi: 10.1016/j.inffus.2019.12.012.
- Chadia Abras, Diane Maloney-Krichmar, Jenny Preece, et al. User-centered design. *Bainbridge, W. Encyclopedia of Human-Computer Interaction*. Thousand Oaks: Sage Publications, 37(4): 445–456, 2004.
- Amina Adadi and Mohammed Berrada. Peeking inside the black-box: A survey on explainable artificial intelligence (xai). *IEEE Access*, 6:52138–52160, 9 2018. ISSN 21693536. doi: 10.1109/ACCESS.2018.2870052.
- Wolfgang Banzhaf, Frank D. Francone, Robert E. Keller, and Peter Nordin. *Genetic Programming: An Introduction: On the Automatic Evolution of Computer Programs and Its Applications*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1998. ISBN 155860510X.
- Vaishak Belle and Ioannis Papantonis. Principles and practice of explainable machine learning. *Frontiers in Big Data*, 4, 07 2021. doi: 10.3389/fdata.2021.688969.
- Nigel Bevana, Jurek Kirakowskib, and Jonathan Maissela. What is usability. In *Proceedings of the 4th International Conference on HCI*, page 24. Citeseer, 1991.
- Adream Blair-Early and Mike Zender. User Interface Design Principles for Interaction Design. *Design Issues*, 24(3):85–107, 07 2008. ISSN 0747-9360. doi: 10.1162/desi.2008.24.3.85. URL <https://doi.org/10.1162/desi.2008.24.3.85>.
- Jan O. Borchers. A pattern approach to interaction design. In *Proceedings of the 3rd Conference on Designing Interactive Systems: Processes, Practices, Methods, and Techniques*, DIS '00, page 369–378, New York, NY, USA, 2000. Association for Computing Machinery. ISBN 1581132190. doi: 10.1145/347642.347795. URL <https://doi.org/10.1145/347642.347795>.
- Elizabeth Charters. The use of think-aloud methods in qualitative research an introduction to think-aloud methods. *Brock Education : a Journal of Educational Research and Practice*, 12: 68–82, 05 2010. ISSN 2371-7750. doi: 10.26522/BROCKED.V12I2.38.
- Qi Chen, Mengjie Zhang, and Bing Xue. Feature selection to improve generalization of genetic programming for high-dimensional symbolic regression. *IEEE Transactions on Evolutionary Computation*, 21:792–806, 10 2017. ISSN 1089778X. doi: 10.1109/TEVC.2017.2683489.
- Michael Chromik and Andreas Butz. Human-xai interaction: A review and design principles for explanation user interfaces. In Carmelo Ardito, Rosa Lanzilotti, Alessio Malizia, Helen Petrie,

- Antonio Piccinno, Giuseppe Desolda, and Kori Inkpen, editors, *Human-Computer Interaction – INTERACT 2021*, pages 619–640. Springer International Publishing, 2021.
- Derek Doran, Sarah Schulz, and Tarek Besold. What does explainable ai really mean? a new conceptualization of perspectives. 10 2017. URL <http://arxiv.org/abs/1710.00794>.
- Filip Karlo Došilović, Mario Brčić, and Nikica Hlupić. Explainable artificial intelligence: A survey. pages 210–215. Institute of Electrical and Electronics Engineers Inc., 6 2018. ISBN 9789532330977. doi: 10.23919/MIPRO.2018.8400040.
- Mengnan Du, Ninghao Liu, and Xia Hu. Techniques for interpretable machine learning. *Commun. ACM*, 63(1):68–77, dec 2019. ISSN 0001-0782. doi: 10.1145/3359786. URL <https://doi.org/10.1145/3359786>.
- Upol Ehsan and Mark O. Riedl. Human-centered explainable ai: Towards a reflective sociotechnical approach. In *HCI International 2020 - Late Breaking Papers: Multimodality and Intelligence: 22nd HCI International Conference, HCII 2020, Copenhagen, Denmark, July 19–24, 2020, Proceedings*, page 449–466, Berlin, Heidelberg, 2020. Springer-Verlag. ISBN 978-3-030-60116-4. doi: 10.1007/978-3-030-60117-1\_33. URL [https://doi.org/10.1007/978-3-030-60117-1\\_33](https://doi.org/10.1007/978-3-030-60117-1_33).
- Upol Ehsan, Philipp Wintersberger, Q. Vera Liao, Elizabeth Anne Watkins, Carina Manger, Hal Daumé III, Andreas Riener, and Mark O Riedl. Human-centered explainable ai (hcxai): Beyond opening the black-box of ai. CHI EA '22, New York, NY, USA, 2022. Association for Computing Machinery. ISBN 9781450391566. doi: 10.1145/3491101.3503727. URL <https://doi.org/10.1145/3491101.3503727>.
- David E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley Longman Publishing Co., Inc., USA, 1st edition, 1989. ISBN 0201157675.
- Bryce Goodman and Seth Flaxman. Eu regulations on algorithmic decision-making and a "right to explanation". *AI Magazine*, 38, 06 2016. doi: 10.1609/aimag.v38i3.2741.
- David Gunning and David Aha. Darpa’s explainable artificial intelligence (xai) program. *AI Magazine*, 40:44–58, 06 2019. doi: 10.1609/aimag.v40i2.2850.
- Hani Hagras. Toward human-understandable, explainable ai. *Computer*, 51(9):28–36, sep 2018. ISSN 0018-9162. doi: 10.1109/MC.2018.3620965. URL <https://doi.org/10.1109/MC.2018.3620965>.
- Rex Hartson and Pardha Pyla. *The UX Book: Process and Guidelines for Ensuring a Quality User Experience*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1st edition, 2012. ISBN 0123852412.
- Marc Hassenzahl and Noam Tractinsky. User experience - a research agenda. *Behaviour and Information Technology*, 25:91–97, 3 2006. ISSN 0144929X. doi: 10.1080/01449290500330331.
- P.L.T. Hoonakker, Pascale Carayon, Ayse Gurses, Roger Brown, Kerry McGuire, Adjhaporn Khunlertkit, and James Walker. Measuring workload of icu nurses with a questionnaire survey: the nasa task load index (tlx). *IIE transactions on healthcare systems engineering*, 1: 131–143, 04 2011. doi: 10.1080/19488300.2011.609524.

- ISO 9241-210:2019. Ergonomics of human-system interaction — Part 210: Human-centred design for interactive systems. Standard, International Organization for Standardization, Geneva, CH, March 2022.
- ISO/IEC 25010:2011. Systems and software engineering — systems and software quality requirements and evaluation (square) — system and software quality models. Standard, International Organization for Standardization, Geneva, CH, March 2022.
- Patrick Jordan. *An Introduction to Usability*. CRC Press, 08 2020. ISBN 9781003062769. doi: 10.1201/9781003062769.
- Frederick Kile. Artificial intelligence and society: a furtive transformation. *AI & SOCIETY*, 28: 107–115, 2 2013. ISSN 0951-5666. doi: 10.1007/s00146-012-0396-0.
- Jon Kolko. *Thoughts on Interaction Design*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2nd edition, 2011. ISBN 0123809304.
- Brenda Laurel. *Design Research: Methods and Perspectives*. MIT Press, Cambridge, MA, USA, 2003. ISBN 0262122634.
- David Leslie. Understanding artificial intelligence ethics and safety. 6 2019. doi: 10.5281/zenodo.3240529. URL <http://arxiv.org/abs/1906.05684>.
- Q. Vera Liao, Daniel Gruen, and Sarah Miller. Questioning the ai: Informing design practices for explainable ai user experiences. Association for Computing Machinery, 4 2020. ISBN 9781450367080. doi: 10.1145/3313831.3376590.
- Ji-Ye Mao, Karel Vredenburg, Paul W. Smith, and Tom Carey. The state of user-centered design practice. *Commun. ACM*, 48(3):105–109, mar 2005. ISSN 0001-0782. doi: 10.1145/1047671.1047677. URL <https://doi.org/10.1145/1047671.1047677>.
- Atiya Masood, Gang Chen, and Mengjie Zhang. Feature selection for evolving many-objective job shop scheduling dispatching rules with genetic programming. In *2021 IEEE Congress on Evolutionary Computation (CEC)*, pages 644–651. Institute of Electrical and Electronics Engineers (IEEE), 8 2021. doi: 10.1109/cec45853.2021.9504895. URL <https://doi.org/10.1109/CEC45853.2021.9504895>.
- Rahul Nadikattu. The emerging role of artificial intelligence in modern society. volume 4, pages 2320–2882, 2016. URL [www.ijcrt.org](http://www.ijcrt.org).
- Jakob Nielsen. Ten usability heuristics, 2005. URL <https://www.nngroup.com/articles/ten-usability-heuristics/>. Accessed: 2022-02-24.
- Jenn Visocky O’Grady and Ken Visocky O’Grady. *A Designer’s Research Manual, Updated and Expanded: Succeed in Design by Knowing Your Clients and Understanding what They Really Need*. Rockport Publishers, 2017. ISBN 1631592629.
- F. Padillo, J. M. Luna, and S. Ventura. A grammar-guided genetic programming algorithm for associative classification in big data. *Cognitive Computation*, 11:331–346, 6 2019. ISSN 18669964. doi: 10.1007/s12559-018-9617-2.
- Thore Reitz, Stephanie Schwenke, Sebastian Hölzle, and Adelheid Gaulty. Usability testing to evaluate user experience on cyclers for automated peritoneal dialysis. *Renal Replacement Therapy*, 7, 12 2021. doi: 10.1186/s41100-021-00340-0.

- Mireia Ribera and Agata Lapedriza. Can we do better explanations? a proposal of user-centered explainable ai. In *IUI Workshops*, volume 2327, page 38. CEUR Workshop Proceedings, 2019.
- Dan Saffer. *Designing for Interaction: Creating Innovative Applications and Devices*. New Riders Publishing, USA, 2nd edition, 2009. ISBN 0321643399.
- Jeff Sauro. 5 ways to interpret a sus score. <https://measuringu.com/interpret-sus-score/>, 2018. Accessed: 2022-04-28.
- Tjeerd Schoonderwoerd, Wiard Jorritsma, Mark Neerincx, and Karel Bosch. Human-centered xai: Developing design patterns for explanations of clinical decision support systems. *International Journal of Human Computer Studies*, 154, 10 2021. ISSN 10959300. doi: 10.1016/j.ijhcs.2021.102684.
- Helen Sharp, Yvonne Rogers, and Jenny Preece. *Interaction Design: Beyond Human Computer Interaction*. John Wiley & Sons, Inc., Hoboken, NJ, USA, 2007. ISBN 0470018666.
- Artem Syromiatnikov and Danny Weyns. A journey through the land of model-view-design patterns. In *2014 IEEE/IFIP Conference on Software Architecture*, pages 21–30, 2014. doi: 10.1109/WICSA.2014.13.
- Jenifer Tidwell. *Designing interfaces: Patterns for effective interaction design*. " O'Reilly Media, Inc.", 2 edition, 2010. ISBN 9781492051961.
- Giulia Vilone and Luca Longo. Explainable artificial intelligence: a systematic review. 5 2020. doi: 10.48550/arXiv.2006.00093.
- Danding Wang, Qian Yang, Ashraf Abdul, and Brian Lim. Designing theory-driven user-centric explainable ai. *Association for Computing Machinery*, 5 2019. ISBN 9781450359702. doi: 10.1145/3290605.3300831.
- Claes Wohlin, Per Runeson, Martin Hst, Magnus C. Ohlsson, Björn Regnell, and Anders Wesslén. *Experimentation in Software Engineering*. Springer Publishing Company, Incorporated, 2012. ISBN 3642290434. doi: 10.5555/2349018.
- Jichen Zhu, Antonios Liapis, Sebastian Risi, Rafael Bidarra, and Michael Youngblood. Explainable ai for designers: A human-centered perspective on mixed-initiative co-creation. volume 2018-August. IEEE Computer Society, 10 2018. ISBN 9781538643594. doi: 10.1109/CIG.2018.8490433.

# Appendix A

## Content Inventory

### Navbar

- Common to all pages in the User and Model areas;
- TRUST-AI logo that redirects the user to the “Projects” page when it is clicked;
- Button to redirect the user to the “Projects” page;
- Button to redirect the user to the “Settings” page;
- Button to logout the user.

### Breadcrumbs

- Common to all pages in the User and Model areas;
- Contain the list of links representing the current page and its “ancestors”.

### Authentication

- TRUST-AI logo;
- TRUST-AI slogan;
- Button to toggle between “Sign In” or “Sign Up”;
- Form and button for signing in;
- Form and for signing up;
- Logos of the project’s partners.

**External Links**

- Logos that serve as links for project partners.

**Projects**

- Field to search projects;
- Button to open "New Project" modal; the modal should have a form for the name and type of the project;
- Table with all the projects owned by the user; each entry should have the name and type of the project, the date when it was last modified, and a button to delete it.

**Algorithms**

- Field to search algorithms;
- Button to open "New Algorithm" modal; the modal should have a field for the name and another to upload the algorithm's content;
- Table with all the algorithms uploaded by the user; each entry should have the name of the algorithms, and a button to delete it.

**Settings**

- fields to edit user's fields such as username, email and password.

**Datasets**

- Field to search datasets;
- Button to open "Upload" modal; the modal should have a field to upload the dataset's file;
- Table with all the datasets uploaded by the user; each entry should have the name of the dataset, a button to delete it, and a button to download it;
- Data visualizer of the selected (clicked) dataset.

**Configurations**

- Field to search configurations;
- Button to open "Create" modal; the modal should have a field to select the algorithm and then the respective fields that can be changed in that same algorithms;
- Table with all the configurations created by the user; each entry should have the name of the configuration, a button to delete it, and a button to duplicate it;
- Parameter visualization of the selected (clicked) algorithm.

### **Sessions**

- Field to search sessions;
- Button to open "New Session" modal; the modal should have a form for the name, dataset, algorithm, and configuration of the session;
- Table with all the sessions created inside the project; each entry should have the name and algorithm of the sessions, its status (e.g., "Created", "Running", "Finished", etc.) the date when it was last modified, and a button to delete it.

### **Bookmarked**

- Table with all the models bookmarked; each entry should have the model's expressions, its metrics and a button to remove it from the bookmarked models.

### **Training**

- Form to edit training settings such as dataset, algorithm and configuration;
- Button to start training;
- Table with the best solution of each generation; each entry should have the generation number, the solution's expression and the generated metrics;
- Chart with the metrics evolution throughout the different generations; this chart should have a filter for the metrics;
- Visualization of the logs that are generated by the algorithm.

### **Filter / Compare**

- Table with all the generated solutions in a session (or more); each entry should have the expression's id, the solution's expression, the generated metrics and a button to bookmark it; this table should be possible to order using the metrics;
- Scatter-plot with all the solutions represented; two drop-down buttons to selected which metrics appears on the X and Y axes;
- Bar chart with all the solutions represented; this chart should have a filter for the metrics;
- Sliders for each of the metrics that allow the user to filter the represented solutions;
- Drop-down button that allow the user to add more sessions to be compared.

### **Evaluate**

- Arrow buttons that allow the user to visualize the filtered expressions, one by one;
- Tree visualization of the solution; should be possible to zoom-in and out;
- Text visualization of the solution; this content should be editable;
- Button to update the solution after it was edited;
- Table containing the metrics for the solution represented and general statistics about the session metrics;
- Bar chart containing the metrics for the solution represented and general statistics about the session metrics; this chart should have a filter for the metrics.



## Appendix B

# Usability Study Information

### TRUST-AI Framework – **Usability Studies**

Hi there! Thank you for taking the time to read this. We are inviting you to participate in a usability study whose goal is to **gather feedback from end-users**, both qualitative and quantitative. This feedback will help **validate the current status of the TRUST-AI Framework** interfaces by **assessing its usability** and helping **identify the weaknesses and strengths** of the platform. This process is an essential step to guarantee that the development of the application heads in the right direction and maintains a high level of user acceptance.

We'll start by introducing you to some concepts about the project and then ask you to perform some tasks using the application. Once the tasks have been completed, we'd like you to **answer some questions regarding feedback** using a *Google Form*. The whole process should take around **45 to 60 minutes**.

#### Requirements:

- Computer with a microphone, internet connection and a web browser installed (preferably *Google Chrome*)
- Access to Google Meet (or another similar platform) so that the session can be coordinated and guided.
- Consent to screen sharing.

#### Background Context

*TRUST-AI Framework* is an **XAI system** developed within the European project TRUST-AI, which seeks to take a step towards the next generation of AI, **resorting to symbolic learning models**. These models are transparent and produce results that are easily interpreted and manipulated by humans, promoting **interaction between humans and AI**, which reinforces the trustworthiness of the latter.

Although still in development, the framework aims at allowing its users to **create XAI projects, upload datasets, select a model** that fits the problem (classification, regression, and prescription), **train and adjust it**, and **visualize the results** graphically and textually. Furthermore, the application will be customized and validated within real-world use-cases such as the retail sector (dynamic time slot pricing) and the health sector (cancer treatment scheduling).

The utility of this approach involves providing users with a **transparent AI system capable of preventing undesirable behaviors** by revealing information that allows the adjusting of different models. The application can be helpful in various sectors where human control and trustworthiness are needed in its AI systems.

#### Demo

A demonstration video of the application can be found [here](#). This will help you gain a general understanding of the application and its main capabilities.

#### Instructions / Features

In case you want to know more about the details of the framework's interfaces, before stepping into the tasks, feel free to read [this document](#) where you can visualize the website's sitemap and the main features of each section.

#### Scheduling

Please feel free to book a slot [here](#).

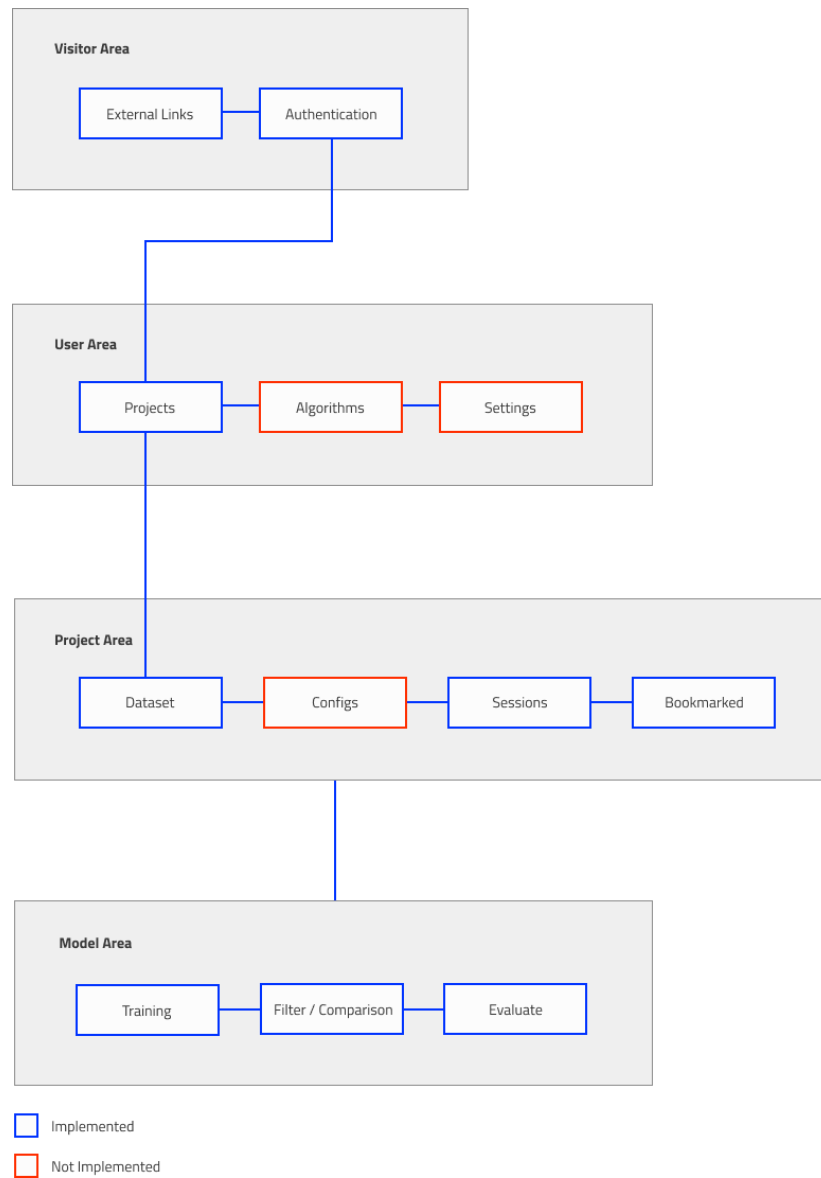
*That's it for now! See you soon.*

## Appendix C

# Usability Study - TRUST Features

## TRUST-AI Framework – Sitemap and Features

## Sitemap



## Features / Sections Descriptions

### Visitor Area

#### **Homepage**

- *sign in*
- *sign up*

### User Area

*Page with sections for user's projects manipulation.*

#### **Projects**

- *List, search, create and delete projects*

### Project Area

*Page with sections regarding any information and necessary data for sessions to be run.*

#### **Datasets**

- *List, search, upload, visualize (text only) and delete datasets*

#### **Sessions**

- *List, search, create and delete sessions*

#### **Bookmarked**

- *Visualize bookmarked expressions*
  - *Expressions are copied to the clipboard on click*

### Model Area

*Page with sections that allow users to visualize a model's training, evaluate it and compare it. Each section has a modular tile-based design with different visualizations and is configurable for each user.*

#### **Training**

- *Update some parameters such as algorithm and dataset*

- *Start training*
- *Table with best model for each generation*
  - *Expressions are copied to the clipboard on click*
- *Chart with metrics evolution*
- *Logs output by the algorithm*

**Filter / Comparison**

- *Table with all the expressions of one or more sessions*
  - *Can be ordered according to each metric*
  - *Expressions are copied to the clipboard on click*
- *Bar chart with metrics comparison*
- *Scatter plot with metrics comparison*
- *Filters for every metric*
- *Dropdown to add/remove sessions being*

**Evaluate**

- *Shows the expressions that were filtered*
- *Code editor to apply changes for the expression*
  - *suggestions for variables and operators*
  - *syntax highlighting*
  - *parsing errors*
  - *latex visualization*
  - *copy expression to clipboard (in latex format)*
- *Tree visualization*
  - *zoom in/out*
  - *pan x/y*
- *Metrics table with statistics about the session*
- *Bar chart with the different metrics*

- *Start training*
- *Table with best model for each generation*
  - *Expressions are copied to the clipboard on click*
- *Chart with metrics evolution*
- *Logs output by the algorithm*

### ***Filter / Comparison***

- *Table with all the expressions of one or more sessions*
  - *Can be ordered according to each metric*
  - *Expressions are copied to the clipboard on click*
- *Bar chart with metrics comparison*
- *Scatter plot with metrics comparison*
- *Filters for every metric*
- *Dropdown to add/remove sessions being*

### ***Evaluate***

- *Shows the expressions that were filtered*
- *Code editor to apply changes for the expression*
  - *suggestions for variables and operators*
  - *syntax highlighting*
  - *parsing errors*
  - *latex visualization*
  - *copy expression to clipboard (in latex format)*
- *Tree visualization*
  - *zoom in/out*
  - *pan x/y*
- *Metrics table with statistics about the session*
- *Bar chart with the different metrics*

## Appendix D

# Usability Study Interviewer Script

### TRUST-AI Framework - **Usability Studies Script**

Hello and thank you for your availability to participate in this study. As you may have read, the goal of this study is to evaluate the current state of the TRUST-AI framework, namely its interfaces, which were developed by me in the scope of my master dissertation.

I will now share a document with you (

■ TRUST AI - Usability Studies Guide.pdf ), that details the things that we are about to do now, but in the meantime, I will also give a brief explanation about it.

So, in this session, I will start by sharing my screen and perform a small demonstration of the platform so that we can familiarize ourselves with it. Although you have been seeing its progress in our monthly meetings. Then I will ask you to be the one sharing the screen and perform 4 different tasks that are described in the document that I shared. After that, I will also ask you to answer a google form which will allow us to obtain the necessary feedback for the study,

#### ***\*Perform Demonstration\****

##### **1 - Create Project:**

Name: Pathfinder

Type: Prescriptive

##### **2 - Show datasets**

##### **3 - Create Session:**

Name: Demo Session

Dataset: Locations\_2.csv

Algorithm: TSP

Config: TSP Config 2 (serve para mudar parâmetros do algoritmo mas ainda está em desenvolvimento e não tem relevância para este estudo)

##### **4 - Train Model:**

Table with the best model of each generation

Metrics evolution in the form of a chart

Logs that are output by the chosen algorithm

#### 5 – Filter Model

Table with all individuals of all generations (can be ordered and bookmarked)

Left: filter by metrics

Scatterplot with the models metrics

Bar chart with the metrics comparison

We can also add more sessions here but we will leave that for later

#### 6 – Evaluation

Expression in text and tree format

Table with metrics for each model and some general statistics

Can add, remove, replace and rearrange windows and the layout stays saved for

each user.

We also have the table data represented in a bar chart

We can edit the expression with the help of some suggestion features.

Can copy the expression in latex format

Can also bookmark

#### 7 – Bookmarked page

**\*Share website\*:** <https://trustai-interfaces.herokuapp.com/>

Now, I'll ask you to share your screen, and let's start the tasks. In the meantime, I want to highlight that you can be completely at ease. After all, this study isn't made to evaluate you, but the platform. So, you can really feel free to ask questions and ask for help. Also, if you feel comfortable, I ask you to try to verbalize your thoughts during the tasks, so we can gather the most information possible.

#### **\*Take notes\***

Having said this, you can now read the first task, ask if you have any questions, and then start working on it.

Great job. Now that the tasks are finished, I will ask you to fill out the form that is mentioned in the document after the tasks



([https://docs.google.com/forms/d/e/1FAIpQLScfAkzydySSzVqCQ3PTJBsC9\\_u9iGfcYGxIwWmf8z\\_LelMe2w/viewform?usp=sf\\_link](https://docs.google.com/forms/d/e/1FAIpQLScfAkzydySSzVqCQ3PTJBsC9_u9iGfcYGxIwWmf8z_LelMe2w/viewform?usp=sf_link)). Take your time and feel free to ask any questions. Once again, I ask you to be as honest as you can and, in the end, feel free to give some verbal feedback as it will be really welcome too.

On my side that is everything. Once again, thank you for your time and availability. If you're interested we can share the final results of the study by email.

# Appendix E

## Usability Study Guide - Pilot Sessions

### TRUST-AI Framework - **Usability Test Guide**

Hi there! Thank you for taking time out of your day to participate in this test, whose goal is to **gather feedback from end-users**, both qualitative and quantitative. This feedback will help **validate the current status of the TRUST-AI Framework** interfaces by **assessing its usability** and helping **identify the weaknesses and strengths** of the platform. This process is an essential step to guarantee that the development of the application heads in the right direction and maintains a high level of user acceptance.

If you wanna know more about the context of the study and the platform, feel free to check this [document](#).

#### Tasks

We're now **ready to start the test**. Before we begin, I'd like to remind you that **we aren't testing you today**. We're testing the TRUST-AI Framework. So if something isn't working, don't worry! It should be a problem with our software and not something you've done wrong. There are no wrong answers here. We also encourage you to **verbalize your thoughts**, if you feel comfortable, **ask questions**, and, if needed, ask for help.

We'd like you to **be as honest as possible**. If you feel like something doesn't make sense or is not working right, please tell us. You're not going to hurt our feelings, so don't worry about that.

Here is a list of the tasks we would like you to complete:

#### **Task 1 - Training Visualization**

In this task, we want you to **run a session and visualize the training evolution**, using the Dataset "locations\_1.csv" and the algorithm "ECJ TSP" (prescriptive problem). For that, you'll have to do some preliminary steps such as creating a project.

**Task 2 – Filter & Compare Results**

In this task, we want you to **use the results from the previous task and compare them** to the session named “Comparison Session”. Can you tell which session resulted in an expression with the best fitness? What about cost? Try to point out those expressions and make use of the filters to get there faster.

**Task 3 – Edit & Copy**

**Find the tree for the expression** with the best fitness and make some changes to it (e.g., add a constant or a variable at the end). **Bookmark the expression and copy its Latex format to your clipboard.**

**Task 4 – Do it All**

Using the Dataset “patients.csv” and the algorithm “EAS” (predictive problem), find the best expression according to fitness and the best according to cost.

Before you begin, we kindly ask you to open the [feedback form](#) and fill out the first section that serves as a support to better characterize the participants of the study.

**Feedback**

While you perform each task, we ask that you fill out the [feedback form](#) as there is a set of 6 questions for each specific task, that will help us assess each task’s workload. After finishing the tasks there is one more set of 10 questions that will be used to evaluate the overall usability of the tool and its adequacy.

You will also have a place for you to share any qualitative feedback that you want to give, regarding each task and anything that you may want to point out.

*That’s it! Thank you for your participation.*

## Appendix F

### Pilot Sessions - Questionnaire Answers

Table F.1: Pilot Sessions - Participants Characterization

participant	1	2	3	4	5	6
<i>What is your age?</i>	17 - 24	25 - 34	25 - 34	35-44	17 - 24	17 - 24
<i>What is your gender?</i>	male	male	male	male	male	male
<i>academic background</i>	informatic engineering	industrial engineering	industrial engineering	computer science	computer science	marketing
<i>academic level</i>	bachelor	phd	phd	phd	master	bachelor
<i>how comfortable do you feel using artificial intelligence algorithms and techniques?</i>	4	3	4	1	4	1
<i>how comfortable do you feel using explainable artificial intelligence algorithms and techniques?</i>	2	3	3	1	2	1
<i>how comfortable do you feel using genetic programming algorithms?</i>	3	4	4	1	1	1
<i>for how many months (full-time) have you used gp algorithms?</i>	0	10	3	0	0	0

Table F.2: Pilot Sessions - Task 1 Nasa TLX Scores

Participant	mental	physical	temporal	performance	effort	frustration
1	2	3	1	1	1	0.4
2	1	1	1	1	1	1
3	1	1	1	2	1	1.5
4	1	1	1	1	1	1
5	2	1	3	1	2	1.8
6	4	3	3	2	3	2.1

NASA TLX Scores range from 1 to 10 (lower is better)

Table F.3: Pilot Sessions - Task 2 Nasa TLX Scores

participant	mental	physical	temporal	performance	effort	frustration
1	2	2	1	1	1	1
2	2	1	2	1	2	2
3	1	1	1	1	1	2
4	1	1	1	1	1	1
5	3	1	3	1	2	1
6	3	2	2	2	2	1

NASA TLX Scores range from 1 to 10 (lower is better)

Table F.4: Pilot Sessions - Task 3 Nasa TLX Scores

participant	mental	physical	temporal	performance	effort	frustration
1	4	2	1	3	2	3
2	2	1	2	1	2	2
3	1	1	1	1	1	1
4	1	1	1	1	1	1
5	1	1	1	1	1	1
6	2	2	2	2	2	1

NASA TLX Scores range from 1 to 10 (lower is better)

Table F.5: Pilot Sessions - Task 4 Nasa TLX Scores

participant	mental	physical	temporal	performance	effort	frustration
1	1	2	1	1	1	1
2	2	1	2	1	2	2
3	1	1	1	1	1	1
4	1	1	1	1	1	1
5	3	1	2	1	2	1
6	4	3	3	2	2	3

NASA TLX Scores range from 1 to 10 (lower is better)

Table F.6: Pilot Sessions - SUS Results

participant	1	2	3	4	5	6
<i>I think that I would like to use this system frequently.</i>	4	5	5	5	5	4
<i>I found the system unnecessarily complex.</i>	1	1	1	1	1	4
<i>I thought the system was easy to use.</i>	4	5	5	5	5	5
<i>I think that I would need the support of a technical person to be able to use this system.</i>	1	2	1	1	1	2
<i>I found the various functions in this system were well integrated.</i>	4	4	4	5	5	4
<i>I thought there was too much inconsistency in this system.</i>	1	1	1	1	1	2
<i>I would imagine that most people would learn to use this system very quickly.</i>	4	5	4	5	5	3
<i>I found the system very cumbersome to use.</i>	1	2	1	1	1	2
<i>I felt very confident using the system.</i>	5	5	4	5	4	3
<i>I needed to learn a lot of things before I could get going with this system.</i>	2	1	3	1	2	1
<b>SUS Score</b>	87,5	92,5	87,5	100	95	70

Answers range from 1 to 5.

SUS Items 2, 4, 6, 8, 10 are negatively worded: lower represents higher perceived satisfaction.

# Appendix G

## Usability Study Guide - Main Sessions

### TRUST-AI Framework - **Usability Test Guide**

Hi there! Thank you for taking time out of your day to participate in this test, whose goal is to **gather feedback from end-users**, both qualitative and quantitative. This feedback will help **validate the current status of the TRUST-AI Framework** interfaces by **assessing its usability** and helping **identify the weaknesses and strengths** of the platform. This process is an essential step to guarantee that the development of the application heads in the right direction and maintains a high level of user acceptance.

If you want to know more about the context of the study and the platform, feel free to check this [document](#).

#### Tasks

We're now **ready to start the tests**. Before we begin, I'd like to remind you that **we aren't testing you today**. We're testing the TRUST-AI Framework. So if something isn't working, don't worry! It should be a problem with our software and not something you've done wrong. There are no wrong answers here. We also encourage you to **verbalize your thoughts**, if you feel comfortable, **ask questions**, and, if needed, ask for help.

We'd like you to **be as honest as possible**. If you feel like something doesn't make sense or is not working right, please tell us. You're not going to hurt our feelings, so don't worry about that.

Here is a list of the tasks we would like you to complete:

#### **Task 0 (Setup) - Training Visualization**

In this task, we want you to start by **creating a project**. Then you should **create and run a session and visualize the training evolution**, using the Dataset "**locations\_1.csv**" and the algorithm "**TSP**" (**prescriptive problem**) with the config "**TSP Config 1**".

**Task 1 – Filter & Compare Results**

In this task, we want you to **use the results from the previous task and compare them** to the session named “First Session”. After that:

- bookmark the expression with **the best (highest) fitness**
- **using the scatterplot, highlight the solution with the least cost**
- bookmark the expression with the **cost equal to 53105**
- see how many expressions need **more than 215s (time)**.
- see how many expressions have a **size higher than 21**, a **cost under 58000** and a **distance under 8700**, and bookmark the **one with the lowest delay**.
- **using the bar chart, compare only the fitness and cost** of the filtered solutions.

**Task 2 – Edit & Evaluate**

In this task we want you to find some expressions, make a few changes to one of them and see the results.

- **find the tree for the expression** with the **highest fitness**
- **make some changes to it** (e.g., add a constant or a variable at the end)
- **copy its text to your clipboard**
- **find the tree for one of the expressions** with the **lowest cost**
- using the **if operator** *change the expression* so that if the **time is higher than 50**, the expression with the highest fitness should be used, otherwise, the one with the lowest cost). Note that the if operator has the following format *if(condition, expression\_if\_condition\_equals\_true, expression\_if\_condition\_false)*
- see **what is the height of the resulting tree** and the **number of children of the first node**.

**Task 3 – Do it All**

Let's **create a project** to predict the possibility of patients suffering from strokes (**predictive problem**). Using the Dataset “**patients.csv**” and the algorithm “**EAS**”, **create a new project**.

- **between the expressions with the highest fitness, find the one with the smallest size and bookmark it.**
- **find the expressions with a size lower than 22.8**



- **identify the expressions that are non-dominated (Pareto efficient) and bookmark them.** A solution is non-dominated if none of the objective functions (metrics) can be improved in value without degrading some of the other objective values. Beware in this problem we want to maximize the fitness in this problem
- **using any expression with size 13, change it** (using the **If operator**) to stay the same, if the **age is higher than 20** and the value **isParent is True** using the **AND Operator**, otherwise the expression should be **bmi\*gender**. Note that the *and* operator has the following format **(condition\_1) AND (condition\_2)**.
- see **what is the height of the resulting tree** and the **number of operators**.

### Feedback

Now, we kindly ask you to open the [feedback form](#) and fill out the first section that serves as a support to better characterize the participants of the study. For each task, we ask that you fill out their respective sections (6 questions for each one), which will help us assess each task's workload.

**The questionnaire is completely anonymous.**

After finishing the tasks there is one more set of 10 questions that will be used to evaluate the overall usability of the tool and its adequacy.

You will also have a place for you to share any qualitative feedback that you want to give, regarding each task and anything that you may want to point out.

*That's it! Thank you for your participation.*

## **Appendix H**

### **Main Sessions - Questionnaire Answers**

Table H.1: Main Sessions - Participants Characterization

What is your age?	25 - 34	25 - 34	17 - 24	25 - 34	25 - 34	17 - 24	25 - 34	25 - 34	17 - 24	45-54	17 - 24	25 - 34
What is your gender?	Male	Male	Male	Female	Female	Female	Male	Male	Male	Male	Male	Male
What is your academic background (e.g. computer science, industrial engineering)?	Industrial Engineering	Industrial Engineering	Industrial Engineering	Mechanical Engineering	Mathematics	Electronics Engineering	Industrial Engineering	Industrial Engineering	Industrial Engineering	Computer Science	Electronics Engineering	Computer Science
What is your academic level (e.g. bachelor, master, phd)?	PhD	Master	Master	Master	PhD	Master	Master	Master	Master	PhD	Master	Master
How comfortable do you feel using Artificial Intelligence Algorithms and Techniques?	4	4	4	3	4	4	5	5	3	5	5	5
How comfortable do you feel using eXplainable Artificial Intelligence Algorithms and Techniques?	3	4	3	4	1	4	5	4	3	4	4	5
How comfortable do you feel using Genetic Programming Algorithms?	4	5	4	4	2	4	5	4	3	3	4	5
For how many months (full-time) have you used GP Algorithms?	0	0	0	3	1	4	4	5	5	6	6	80

Table H.2: Main Sessions - Task 1 Nasa TLX Scores

participant	mental	physical	temporal	performance	effort	frustration
1	2	1	1	2	2	2
2	4	1	6	2	3	2
3	4	3	3	9	5	1
4	7	4	5	3	7	3
5	2	2	5	9	3	2
6	3	1	1	8	3	2
7	3	1	3	2	2	1
8	2	2	8	2	2	1
9	1	1	3	1	1	5
10	5	4	5	2	3	1
11	6	6	5	9	5	2
12	7	1	4	2	2	2

NASA TLX Scores range from 1 to 10 (lower is better)

Table H.3: Main Sessions - Task 2 Nasa TLX Scores

participant	mental	physical	temporal	performance	effort	frustration
1	2	1	1	3	3	2
2	2	1	5	1	2	1
3	6	3	3	9	4	1
4	7	3	3	8	7	3
5	3	2	4	9	3	2
6	3	1	1	9	4	1
7	3	1	2	9	2	1
8	2	2	8	2	2	1
9	1	1	3	1	1	5
10	5	4	5	2	3	1
11	6	3	2	9	3	2
12	2	1	1	1	1	1

NASA TLX Scores range from 1 to 10 (lower is better)

Table H.4: Main Sessions - Task 3 Nasa TLX Scores

participant	mental	physical	temporal	performance	effort	frustration
1	2	1	2	2	3	3
2	2	1	5	1	2	1
3	6	3	3	8	5	1
4	7	3	3	8	7	3
5	2	2	5	9	3	2
6	1	1	2	10	2	1
6	3	1	2	2	2	1
7	2	2	8	3	2	1
8	1	1	3	1	1	5
10	5	4	4	2	2	2
11	4	2	2	9	2	2
12	2	1	2	2	1	2

NASA TLX Scores range from 1 to 10 (lower is better)

Table H.5: Main Sessions - SUS Results


participant	1	2	3	4	5	6	7	8	9	10	11	12
<i>I think that I would like to use this system frequently.</i>	5	4	5	3	4	4	5	5	2	4	4	5
<i>I found the system unnecessarily complex.</i>	1	1	1	1	1	2	1	1	2	1	1	2
<i>I thought the system was easy to use.</i>	4	5	5	4	5	4	5	5	4	5	4	4
<i>I think that I would need the support of a technical person to be able to use this system.</i>	2	2	1	2	1	1	1	1	3	1	2	1
<i>I found the various functions in this system were well integrated.</i>	4	3	5	4	4	4	5	4	2	4	4	4
<i>I thought there was too much inconsistency in this system.</i>	1	2	1	2	1	2	1	1	4	1	2	2
<i>I would imagine that most people would learn to use this system very quickly.</i>	5	4	5	4	4	4	4	5	1	4	3	2
<i>I found the system very cumbersome to use.</i>	1	2	1	2	1	2	1	1	2	1	2	1
<i>I felt very confident using the system.</i>	4	5	5	4	4	4	5	5	3	5	3	4
<i>I needed to learn a lot of things before I could get going with this system.</i>	3	1	2	2	2	2	1	2	1	2	4	1
<b>SUS Score</b>	85	82,5	97,5	75	87,5	77,5	97,5	95	50	90	67,5	80

Answers range from 1 to 5.

SUS Items 2, 4, 6, 8, 10 are negatively worded: lower represents higher perceived satisfaction.

## Appendix I

# TRUST-AI - Workshop Presentation




**TRUSTAI**

TRANSPARENT, RELIABLE  
& UNBIASED SMART TOOL

## TRUST-AI Framework

Interfaces

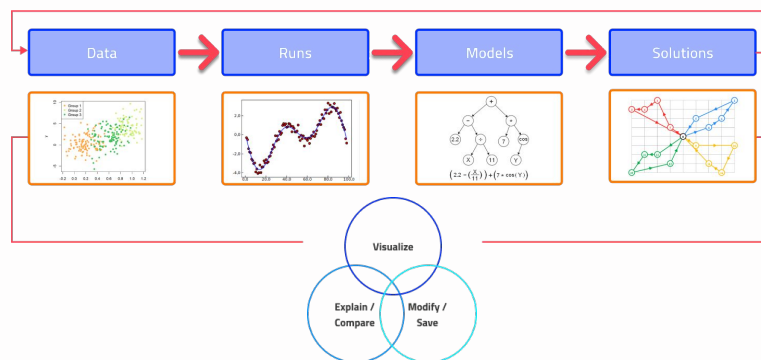
29/06/2022



**Agenda**

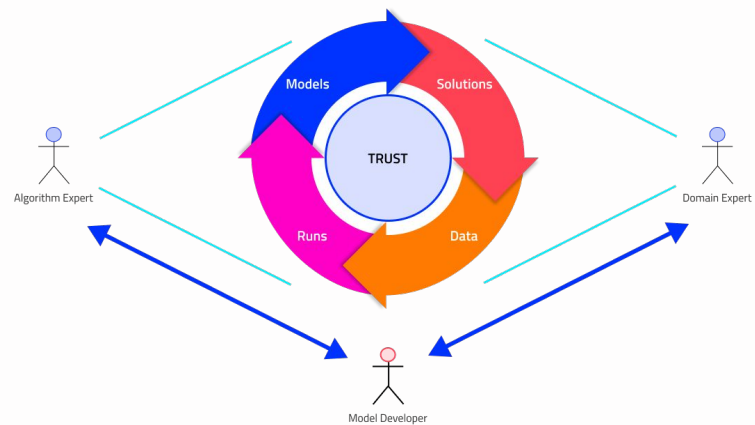
1. Main Workflow
2. Users of the System
3. Demonstration
4. Exercise

## Main Workflow



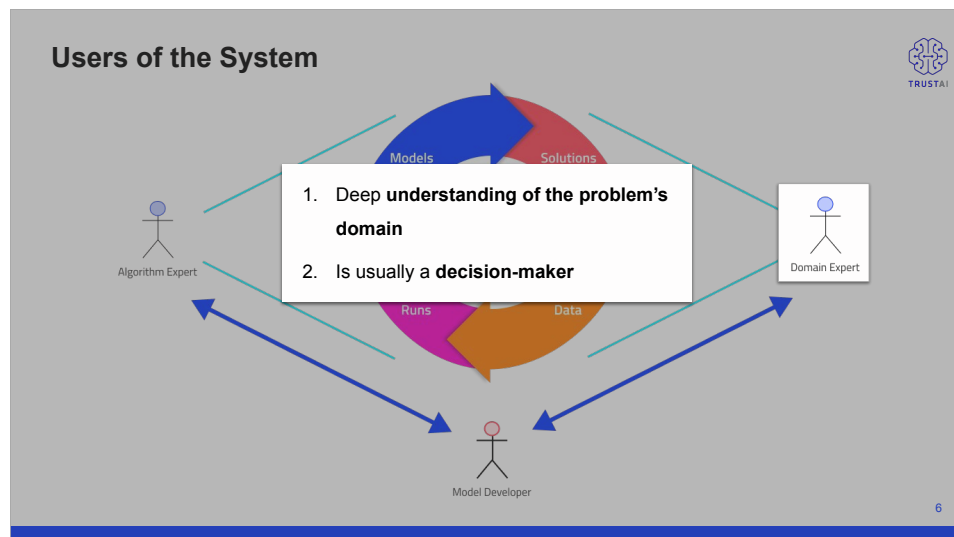
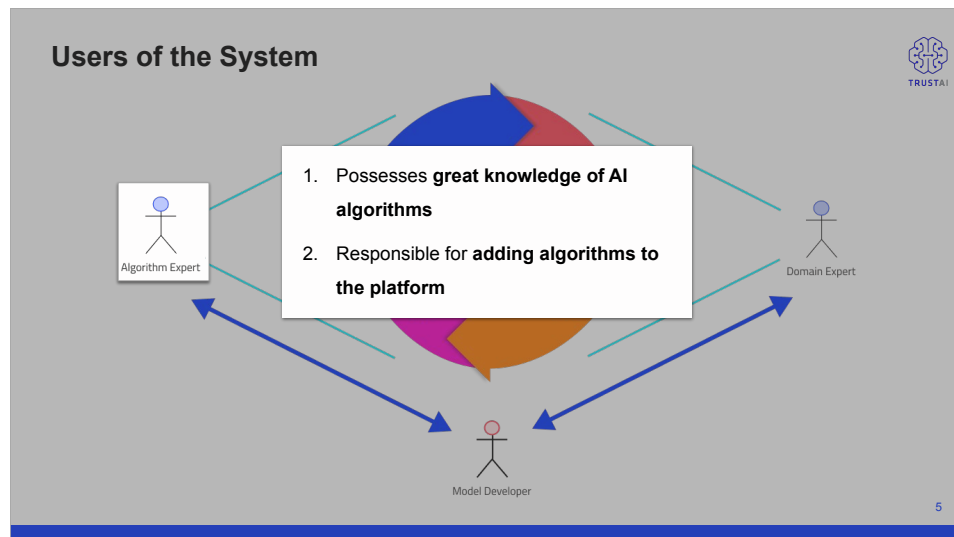
3

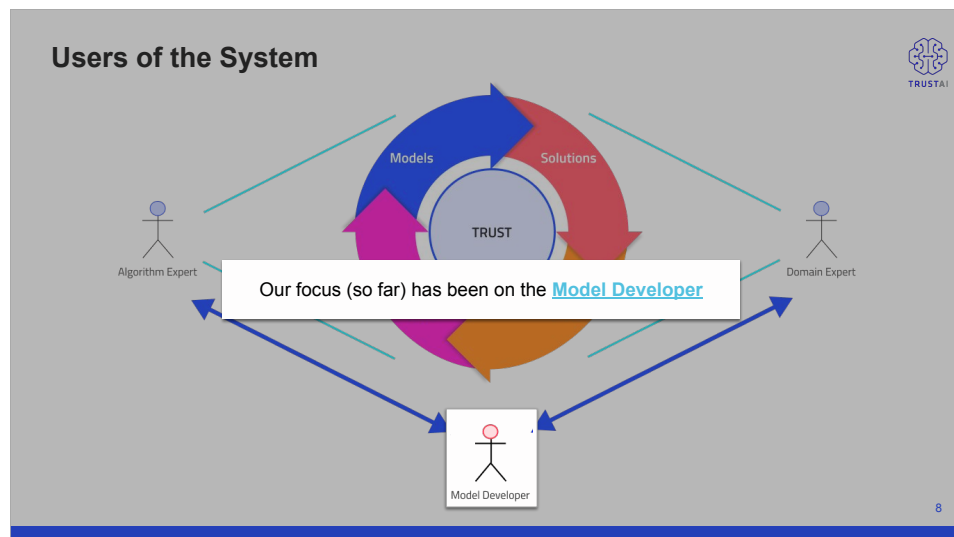
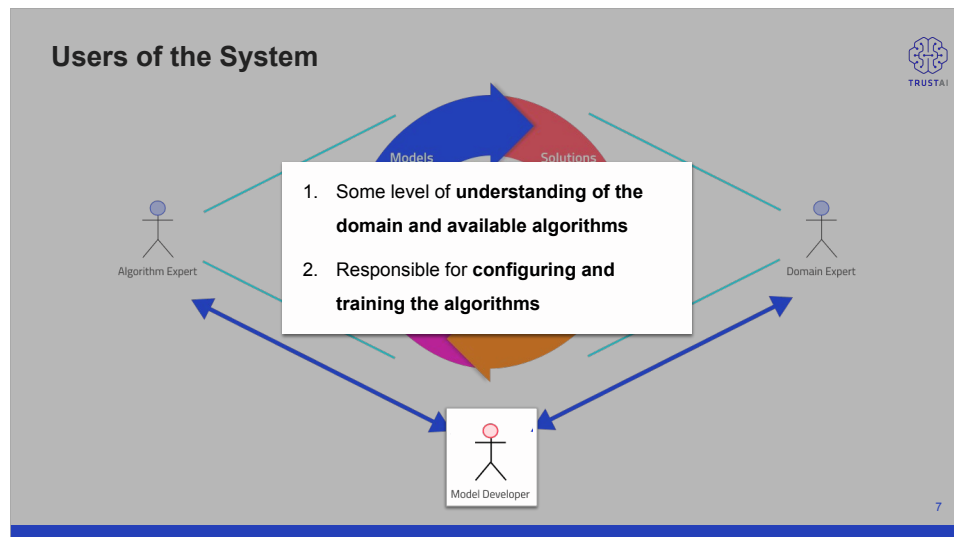
## Users of the System



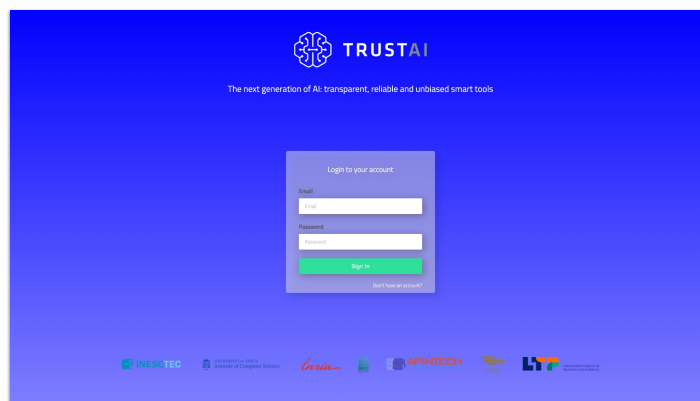
4







## Demonstration



9

## Exercise



1. Join your team (see table on the right)
2. Open *Google Remote Desktop* website in your browser
3. Log in using:  
Email: **demo.<team-number>.trust.ai@gmail.com**  
Password: **trust.ai.demo**
4. Choose TRUST-AI (PIN: 123456, Ubuntu password: 1234)
5. Open [TRUSTAI - Workshop Exercise](#) from your desktop
6. Perform the tasks mentioned in the document
7. Fill out the [feedback form](#) you'll find in the document

Team	Members		
1	Marc Schoenauer	Nikos Sakkas	Nuno Marques
2	Eduard Barbu	Costas Daskalakis	Sérgio Castro
3	Marharyta Domnich	Nikitas Sakkas	Francisco Maia
4	Dazhuang Liu	Christina Chaniotaki	Bernardo Almada-Lobo
5	Marco Virgolin	Teresa Bianchi Aguiar	Daniela Fernandes
6	Peter Bosman	André Morim	Luís Guimarães
7	Tanja Alderliesten	Francisco Amorim	Tanju Cataltepe
8	Evi Sijben	Pedro Amorim	Manolis Christoforou

Feel free to ask for help

10



**TRUSTAI**

TRANSPARENT, RELIABLE  
& UNBIASED SMART TOOL

# TRUST-AI Framework

Interfaces

29/06/2022

## Appendix J

# TRUST-AI - Workshop Exercise Guide

### TRUST-AI Framework - **Workshop Exercise**

Hi there! Thank you for taking time out of your day to participate in this exercise, whose goal is to **disseminate the TRUSTAI -framework** and **gather feedback**, both qualitative and quantitative. This feedback will help **validate the current status of the TRUST-AI Framework** interfaces by **assessing its usability** and helping **identify the weaknesses and strengths** of the platform. This process is an essential step to guarantee that the development of the application heads in the right direction and maintains a high level of user acceptance.

Before we start the exercise, feel free to check out a video demonstration of the application [here](#). We have also prepared [a document](#) where you can visualize the website's sitemap and the main features of each section.

#### Tasks

We're now **ready to start the tests**. Before we begin, I'd like to remind you that **we aren't testing you today**. We're testing the TRUST-AI Framework. So if something isn't working, don't worry! It should be a problem with our software and not something you've done wrong. There are no wrong answers here. We also encourage you to **verbalize your thoughts**, if you feel comfortable, **ask questions**, and, if needed, ask for help.

We'd like you to **be as honest as possible**. If you feel like something doesn't make sense or is not working right, please tell us. You're not going to hurt our feelings, so don't worry about that.

Here is a list of the tasks we would like you to complete:

#### **Task 0 (Setup) - Training Visualization**

In this task, we want you to start by **creating a project**. Then you should **create and run a session and visualize the training evolution**, using the Dataset

“dataset” available in your desktop and the algorithm Job Shop Scheduling Problem (JSSP) which is a **prescriptive algorithm**. [Here, you can find a short description of the JSSP algorithm.](#)

### Task 1 – Filter & Compare Results

In this task, we want you to **use the results from the previous task** to perform the following:

- bookmark the expression with **the best (highest) fitness**.
- **using the scatterplot, highlight the solution with the highest flowtime**.
- bookmark the expression with the **tardiness equal to 10.18**.
- see how many expressions need **more than 215s (flowtime)**.
- see how many expressions have a **size lower than 25.5, a flowtime under 50.27, and fitness higher than 0.23**, and bookmark the **one with the lowest tardiness**.
- **with the same filters applied identify the expressions that are non-dominated (Pareto efficient) and bookmark them**. A solution is non-dominated if none of the objective functions (metrics) can be improved in value without degrading some of the other objective values. Beware that in this problem we want to **minimize the fitness** and **minimize size**.
- **using the bar chart, compare only the fitness and tardiness** of the filtered solutions.

### Task 2 – Edit & Evaluate

In this task we want you to find some expressions, make a few changes to one of them and see the results.

- **remove all filters**.
- **find the tree for the expression** with the **highest fitness**.
- **make some changes to it** (e.g., add a constant or a variable at the end).
- **copy its text to your clipboard**.
- **find the tree for the expressions** with the **lowest tardiness**.
- using the **if operator**, *change the expression* so that if the **slack is lower than 2** and the **WinQF is higher than 50**, the expression with the highest fitness should be used, otherwise, the one with the lowest tardiness. Note that the **if operator** has the following format **if(condition, expression\_if\_condition\_equals\_true, expression\_if\_condition\_false)** and the **and operator** has the following **(condition\_1) AND (condition\_2)**.

- see **what is the height of the resulting tree** and the **number of children of the first node**.

### Feedback

Now, we kindly ask you to open the [feedback form](#) and fill out the first section that serves as a support to better characterize the participants of the study. For each task (“Task 1 – Filter & Compare Results” and “Task 2 – Edit & Evaluate”), we ask that you fill out their respective sections (6 questions for each one), which will help us assess each task’s workload.

**The questionnaire is completely anonymous.**

After finishing the tasks there is one more set of 10 questions that will be used to evaluate the overall usability of the tool and its adequacy.

You will also have a place for you to share any qualitative feedback that you want to give, regarding each task and anything that you may want to point out.

*That’s it! Thank you for your participation.*