

FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

# TruData: A Collaborative Platform for Data Cleaning

Rogério Luiz Araújo Carminé



Mestrado em Engenharia de Software

Supervisor: Professor Carlos Manuel Milheiro de Oliveira Pinto Soares (FEUP)

Second Supervisor: Francisco José Lopes Veiga (Fraunhofer Portugal AICOS)

July 29, 2021



# **TruData: A Collaborative Platform for Data Cleaning**

**Rogério Luiz Araújo Carminé**

Mestrado em Engenharia de Software

July 29, 2021



# Abstract

The growing need for data in many areas (e.g., science, education, health) raises concerns about data quality as it impacts the generation of effective information for decision-making. Data cleaning tasks solve data quality issues; however, they are very time-consuming and unpleasant for a significant number of professionals. Comparative analyzes of commercial and academic solutions indicate a promising direction in reusing elements and collaboration among people who experience similar problems of this type.

This work proposes the TruData platform, which allows reducing time and effort, in addition to increasing the satisfaction in performing data cleaning tasks through reuse and collaboration. To do this, a solution model was developed to support the sharing of data quality requirements and data cleaning operators from multiple platforms (e.g., Python and R), in addition to collecting usage data from people with different roles and from different domains, which experience similar data quality issues. In addition, an application based on the solution components related to the reuse of data cleaning operators was implemented and validated. The validation was carried out through a usability test involving professionals who work in data preparation and compared the performance of data cleaning tasks with and without applying the proposed solution.

The results demonstrate a decline in time and effort and increased satisfaction in performing data cleaning tasks when TruData was applied. They also indicate that it is necessary to prepare data cleaning operators for wide sharing properly, deal with concerns about source code reliability, and provide an effective user experience to choose the more suitable operator according to the application.

**Keywords:** Data Cleaning, Error Detection, Error Repairing, Data Preparation, Data Quality



# Resumo

A crescente necessidade de dados em muitas áreas (por exemplo, ciência, educação, saúde) aumenta as preocupações com qualidade dos dados, pois impacta na geração de informação efetiva para tomada de decisão. As tarefas de limpeza de dados são realizadas para resolver problemas de qualidade de dados, porém, são muito demoradas e desagradáveis para uma parte significativa dos profissionais. Análises comparativas de soluções comerciais e acadêmicas indicam uma direção promissora na reutilização de elementos e colaboração entre as pessoas que passam por problemas semelhantes desse tipo.

Este trabalho propõe a plataforma TruData, que permite reduzir tempo e esforço, além de aumentar a satisfação em realizar tarefas de limpeza de dados através da reutilização e colaboração. Para isso, foi elaborado um modelo de solução para apoiar o compartilhamento de requisitos de qualidade e operadores de limpeza de dados de várias plataformas (por exemplo, Python e R), além de coletar dados de utilização por pessoas com funções diferentes e de diferentes domínios, que passam por problemas de qualidade de dados semelhantes. Além disso, foi implementada e validada uma aplicação baseada nos componentes da solução relacionados à reutilização de operadores de limpeza de dados. A validação foi realizada através de um teste de usabilidade envolvendo profissionais que atuam na preparação de dados, e comparou o desempenho das tarefas de limpeza de dados com e sem aplicação da solução proposta.

Os resultados demonstram uma redução no tempo e no esforço, além do aumento na satisfação para realizar tarefas de limpeza de dados quando o TruData foi aplicado. Eles também indicam ser necessário preparar os operadores de limpeza de dados para um amplo compartilhamento, lidar com as preocupações sobre a confiabilidade do código-fonte e fornecer uma experiência de usuário eficaz para escolher o operador mais adequado de acordo com a aplicação.

**Keywords:** Limpeza de Dados, Detecção de Erros, Reparação de Erros, Preparação de Dados, Qualidade de Dados





# Acknowledgements

First of all, I am very grateful to my supervisor, Professor Carlos Soares, for his support, guidance, and partnership during this challenging journey, in addition to the great example of being a successful person and professional, which has been a great inspiration for me.

I would like to thank my co-supervisor Francisco Veiga, from Fraunhofer AICOS Portugal, for his guidance and friendship, sharing tips and practical experiences that helped me focus on what is important.

I also thank Fraunhofer AICOS Portugal for providing me resources and excellent professionals to support this work, particularly Ana Vasconcelos, for helping me with the usability elements, and Dinis Moreira and Pedro Faria, for helping me to clarify important issues during the development of this work.

Special thanks to the professors and administrative staff of the Masters in Software Engineering, in particular, Professor Ana Paiva and Professor Nuno Flores for their help in many moments, and also to my colleagues for the challenging and unforgettable experiences we spent together, especially to Catarina Gomes and Diogo Melo who became great friends.

I would like to thank Secretaria Municipal de Saúde de Manaus - SEMSA for making this learning experience possible and provide opportunities to contribute to the public health field. Special thanks to my friends Alexandra Muniz, Jean Abreu, Mário Torres, Saymon Souza for the initial support in this journey and, especially, during the most challenging moments.

I also thank Instituto Federal do Amazonas - IFAM for encouraging my professional development, especially to Professor Ribamar Cardoso, who constantly reminded me of the importance of having a master's degree, and also to dear friends Andrea Mendonça, Jucimar Souza, Jorlene Marques, Neila Xavier, and Janduy Medeiros for their encouragement and support at all times.

Special thanks to my friends from the University of Brasília and ProEpi for their usual support and partnership, in addition to providing great experiences and opportunities for contributing to global health, especially to Jonas Brant and Sara Ferraz, for the example of the great people, professionals, and leaders that they are.

I am so grateful to my parents, who were the first teachers in my life and taught me the principles of honesty and hardworking, that guided my way to reach this moment. Special thanks to my mother for her unconditional love and care for me, even very far away from home.

I am really grateful to my wife and children for joining me on this journey, full of challenges and surprises, which have helped us to value our family life more and more. Thank you very much for understanding the countless moments of absence to complete this work.

Finally, I thank all the other colleagues, family members, and friends who in some way supported this 2-year journey, especially to Gleyton Lima for his support as always and for the rich and enjoyable conversations that inspired me with ideas for this work.

Rogério Luiz Araújo Carminé



*“I can only think of so many good ideas in a week.  
Having other people contribute makes my life easier.”*

Martin Fowler



# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Motivation . . . . .	2
1.2	Goals . . . . .	2
1.3	Dissertation Structure . . . . .	3
<b>2</b>	<b>Background</b>	<b>5</b>
2.1	Data Quality Issues . . . . .	5
2.1.1	Data Redundancy . . . . .	5
2.1.2	Data Consistency . . . . .	6
2.1.3	Uniqueness . . . . .	6
2.1.4	Data Completeness . . . . .	6
2.1.5	Data Validity . . . . .	7
2.1.6	Data Accuracy . . . . .	7
2.2	Data Cleaning . . . . .	8
2.2.1	Main Concepts . . . . .	8
2.2.2	Extensibility . . . . .	9
2.2.3	Data Privacy . . . . .	9
2.3	Software Reuse and Collaboration for Data Cleaning . . . . .	10
2.3.1	Software Reuse . . . . .	10
2.3.2	Collaboration . . . . .	11
<b>3</b>	<b>State of the Art</b>	<b>13</b>
3.1	Data Cleaning Tools . . . . .	13
3.1.1	Comparison Based on Main Features . . . . .	13
3.1.2	Comparison based on Data Quality Issues . . . . .	16
3.2	Related Work . . . . .	18
3.2.1	Comparison Analysis of the Related Works . . . . .	21
3.3	Gap Analysis . . . . .	26
<b>4</b>	<b>TruData</b>	<b>29</b>
4.1	Solution Overview . . . . .	29
4.2	Development . . . . .	32
4.2.1	Functional Perspective . . . . .	32
4.2.2	Data Perspective . . . . .	34
4.2.3	Deployment Perspective . . . . .	38

<b>5</b>	<b>Proof of Concept</b>	<b>41</b>
5.1	Methodology . . . . .	41
5.2	Implementation . . . . .	42
5.2.1	Functionalities . . . . .	42
5.2.2	Deployment and Technologies . . . . .	44
5.3	Validation . . . . .	46
5.3.1	Methodology . . . . .	47
5.4	Results . . . . .	48
5.4.1	Participants . . . . .	48
5.4.2	Tasks . . . . .	48
5.4.3	Comments from the Participants . . . . .	53
5.5	Discussion . . . . .	54
5.5.1	Answers to Research Questions . . . . .	54
5.5.2	TruData and Related Works . . . . .	56
<b>6</b>	<b>Conclusions and Future Work</b>	<b>59</b>
6.1	Results . . . . .	59
6.2	Main Contributions . . . . .	60
6.3	Future Work . . . . .	61
<b>A</b>	<b>State of the Art</b>	<b>63</b>
A.1	Empirical Analysis of the Commercial Tools . . . . .	63
<b>B</b>	<b>Proof of Concept</b>	<b>65</b>
B.1	Methodology Overview . . . . .	65
B.2	Dataset . . . . .	65
B.3	Questionnaire . . . . .	66
B.4	Usability Test Description . . . . .	70
B.5	Consent Form . . . . .	78
B.6	Results - Details . . . . .	79
B.7	Tasks - Statistical Analysis Results . . . . .	81
B.8	Original Comments from the Participants . . . . .	82
	<b>References</b>	<b>85</b>

# List of Figures

2.1	Mind map: Main Concepts of Data Cleaning . . . . .	9
3.1	CoClean: Snapshot of the Web UI for data cleaning [30] . . . . .	19
3.2	TDE: Snapshot of the UI for values transformation [17] . . . . .	19
3.3	Autotransform - List of Source and Target Patterns for Data Transformation [23] . . . . .	20
3.4	Potter’s Wheel: Snapshot of the UI with a dataset loaded [35] . . . . .	20
3.5	Mind Map of the Compared Solutions . . . . .	26
4.1	Overview of the solution TruData . . . . .	29
4.2	Representation of data cleaning operator and related cataloging elements . . . . .	30
4.3	Simplified functional view of the solution . . . . .	30
4.4	Representation of data requirement and related elements . . . . .	31
4.5	Association between DQR and DCO assisted by social features . . . . .	31
4.6	Modules of the solution . . . . .	33
4.7	Functionalities of the data requirement module . . . . .	33
4.8	Functionalities of the data cleaning operator module . . . . .	34
4.9	Functionalities of the recommendation support module . . . . .	34
4.10	Functionalities of the administration module . . . . .	35
4.11	Package diagram of the data domain modelling . . . . .	35
4.12	Simplified class diagram of the catalog package . . . . .	36
4.13	Simplified class diagram of the social package . . . . .	37
4.14	Simplified class diagram of the catalog and social packages . . . . .	38
4.15	Simplified class diagram of the social and data analysis packages . . . . .	39
4.16	Deployment diagram of the TruData . . . . .	40
5.1	Overview of the Validation Methodology . . . . .	42
5.2	Functionality - Search for Operators . . . . .	42
5.3	Functionality - Results of the Search for Operators . . . . .	43
5.4	Functionality - View Details of an Operator . . . . .	44
5.5	Functionality - View Details of an Operator - Related Operators and User comments . . . . .	44
5.6	Source code of a Data cleaning Operator in a GIT repository . . . . .	45
5.7	Deployment view of the implementation . . . . .	46
5.8	Tasks comparison based on completion time . . . . .	49
5.9	Tasks comparison based on effort level . . . . .	49
5.10	Tasks comparison based on satisfaction level . . . . .	50
5.11	Number of errors and assistance during the tasks performance . . . . .	50
5.12	Ratings and comments influence on choosing the operator . . . . .	51
5.13	Usage Aspects of the Solution: Easy to Use, Frequent Use, Learn Rapidly . . . . .	51
5.14	Group Comparison - Completion Time Metric . . . . .	51

5.15	Group Comparison - Effort Metric . . . . .	52
5.16	Group Comparison - Satisfaction Metric . . . . .	52
5.17	Group Comparison - Influence of ratings and comments . . . . .	53
5.18	Usage Aspects per Group: Easy to Use, Frequent Use, Learn Rapidly . . . . .	53
A.1	Fictional Dataset of Disease Cases Investigation for Evaluating The Commercial Tools . . . . .	63
B.1	Sample of the Dataset of Disease Cases Investigation for the Solution Validation .	65



# List of Tables

3.1	Comparison of the Data Cleaning Tools based on Main Features . . . . .	14
3.2	Comparison of tools per data quality issues . . . . .	17
3.3	Comparison of the Related Works . . . . .	21
5.1	Contexts that the participants applied data preparation . . . . .	48
B.1	Information about the Participants . . . . .	79
B.2	Completion Time, Errors and Assists per Participant and Task . . . . .	79
B.3	Results - Effort, Satisfaction, and Other Aspects per Participant and Task . . . . .	80
B.4	Comparison based on time, effort and satisfaction metrics . . . . .	81
B.5	Comparison based on completion time metric - Statistical Analysis . . . . .	81
B.6	Comparison based on Effort Level - Statistical Analysis . . . . .	81
B.7	Comparison based on Satisfaction Level - Statistical Analysis . . . . .	81
B.8	Comparison based on Numbers of Errors and Assistance . . . . .	81
B.9	Group Comparison per Completion Time - Statistical Analysis . . . . .	82
B.10	Group Comparison per Effort Level - Statistical Analysis . . . . .	82
B.11	Group Comparison per Satisfaction Level - Statistical Analysis . . . . .	82



# Abbreviations

ADT	Abstract Data Type
AI	Artificial Intelligence
API	Application Programming Interface
CDF	Collaborative Data frame
CRISP-DM	Cross-industry Standard Process for Data Mining
DCO	Data Cleaning Operator
DSL	Domain-specific Language
DQR	Data Quality Requirement
GUI	Graphical User Interface
POC	Proof of Concept
RDBMS	Relational Database Management System
REST	Representational State Transfer
UDF	User-defined Functions
UI	User Interface
UML	Unified Modeling Language
URL	Uniform Resource Locator



# Chapter 1

## Introduction

Digital transformation has increased the need for data in many areas (e.g., industry, commerce, education, healthcare) [19]. Recently, due to the COVID-19 pandemic declared in 2020, the importance of having better data for decision-making has become more notorious [9].

Erroneous data impacts the generation of information, and consequently, decision-making; so it is necessary to ensure data quality. Some organizations have a high level of digitization of their processes; however, this is not yet true in other contexts, which use standalone applications, electronic spreadsheets, or even paper-based forms for computerizing their work processes. For instance, some health organizations in developing countries do not have a well-established digital information system for managing clinical data in health facilities, leading to problems in consolidating data to obtain effective information [21, 28, 45]

To reach a high level of digitization is considered a long-term journey and depends on organizational processes, resources, and cultural aspects [19]. A certain level of data quality is essential for efficient business processes and impacts the digital transformation [36]. Thus, it is important to think of solutions to support data quality in organizations that do not have the elements yet to move forward on digital transformation.

Data quality issues may occur in several applications (e.g., Business Intelligence-BI, Big Data, and Statistical Analysis). Regarding the BI approach, for instance, it may be necessary to integrate databases from different domains (e.g., sales and manufacturing) and then harmonize the data for analysis [16]. As for Big Data applications, there are some data quality issues related to the high volume of data and the variety of data sources, which need solutions for sampling and working distributed [7]. There can be also issues in statistical analysis regarding data anomalies (e.g., missing values, outliers, and inconsistencies), which can impact, for instance, the results of predictive models in data science projects.

Data cleaning tasks are typically performed during data preparation activities to identify and repair errors in the data. There is a variety of tools to support these tasks, such as electronic spreadsheets, database management tools, and statistical tools (e.g., R language<sup>1</sup>, Python<sup>2</sup>, and

---

<sup>1</sup><https://www.r-project.org/>

<sup>2</sup><https://www.python.org/>

Stata<sup>3</sup>) and also specific tools for data cleaning (e.g., Trifacta<sup>4</sup>, Clarity<sup>5</sup> and OpenRefine<sup>6</sup>). Each tool has its feature set related to some aspects (e.g., error detection and repairing techniques, extensibility, reuse, and collaboration).

## 1.1 Motivation

Data cleaning tasks require a lot of effort and are also very time-consuming. It is estimated that data cleaning and other data preparation activities represent 60% of data science tasks, and it is considered unpleasant work by more than 50% of data scientists [34].

Some data cleaning tasks that take a lot of time and effort to complete (e.g., harmonizing the names of municipalities in a country) can be applied in other scenarios, with little or no need for modification. Other tasks may need to involve people of different skills (e.g., domain expert, data analyst, software developer) to complete.

These challenges are common to software engineering, and two approaches that have been used to address them are reuse and collaboration. Software reuse, according to Sommerville [38], can reduce effort, time, and risk in developing new solutions; it is commonly adopted in the software industry as a way to increase the return of investment. Collaboration in software projects reduces not only costs, which is related to time and effort aspects, but also increases innovation [13]; there are examples of collaboration platforms for many applications (e.g., office, software development, project management, and data science).

Some data cleaning tools allow the reuse of operations in several ways (e.g., copy-and-paste and export-and-import methods) and also collaboration among users (e.g., sharing the dataset and/or the script). However, the reuse often seems to occur in an unstructured way, and the collaboration usually appears within the boundaries of a team or organization. These approaches can be applied in a more structured and broader way, enabling people from different locations and with different roles to work together to create data cleaning operations with less time and effort, also reducing the discomfort of performing this type of task.

## 1.2 Goals

This work aims to investigate whether the reuse and collaboration concepts, when applied to data cleaning, can reduce time and effort in carrying out tasks and increase user satisfaction. To achieve this primary objective, this work intends to reach the specific goals:

- Develop a solution model to enable the reuse of data cleaning elements and the broad collaboration among users who experience similar problems;
- Implement a Proof of Concept (PoC) based on the solution model;

---

<sup>3</sup><https://www.stata.com/>

<sup>4</sup><https://www.trifacta.com/>

<sup>5</sup><https://clarity.cloud.tibco.com/>

<sup>6</sup><https://openrefine.org/>

- Validate the application with the target audience.

During the investigation process, which considered *the application of reuse and collaboration concepts on performing data cleaning tasks*, it was raised the following research questions which contribute to the conclusion of this work:

- Is it possible to reduce the completion time on performing data cleaning tasks?
- Is it possible to reduce the effort on executing data cleaning tasks?
- Is it possible to increase the user satisfaction on performing data cleaning tasks?
- Is it necessary to prepare the data cleaning operations for reuse?
- What user experience elements affect the reuse and collaboration on data cleaning elements?

### 1.3 Dissertation Structure

This section presents the structure of the dissertation for better guidance on the topics developed throughout this work.

- Chapter 2 presents essential concepts in the literature on data quality and data cleaning, which will help understand the other topics in this work.
- Chapter 3 presents the solutions available in the market and the academic contributions, more focused on reuse and collaboration aspects, that contributed to data cleaning.
- Chapter 4 presents the proposal for the solution of this work to contribute to the theme of data cleaning, describing the overview and functional and architectural elements of the solution.
- Chapter 5 presents the validation, describing the process and the results obtained.
- Chapter 6 concludes this dissertation, presenting the conclusions obtained and also suggestions for future work.





# Chapter 2

## Background

This chapter introduces general concepts about data quality issues and data cleaning, including software reuse and collaboration, which will be helpful in better understanding the following chapters.

### 2.1 Data Quality Issues

Data quality issues are present in many applications (e.g., Information Security [47], Social Housing [10], Pandemics [9]). These issues can impact the information for decision-making by affecting the accuracy or even providing erroneous results. Hence, it is important to detect and repair them by performing data cleaning operations.

In the following subsections, common data quality issues are described to provide an overview of this topic that will be helpful to understand the data cleaning tools and related works in the further sections.

#### 2.1.1 Data Redundancy

Data Redundancy occurs when a specific value is represented in the computer system multiple times [20]. For instance, a sales system keeps the customer address on multiple records according to the organization's departments (e.g., customer, sales, and delivery records). According to Heuser [20], there are two types of data redundancy: controlled and uncontrolled. The first one occurs when the system is aware of the multiple data representations and ensures their synchronization (e.g., distributed systems, backup, and cache mechanisms). The second usually occurs when the user is responsible for the multiple data representations and does not guarantee their correct synchronization, leading to unnecessary effort on data collection and data inconsistency.

Data redundancy can appear in simple and complex scenarios; it depends on the data architecture decisions in the information system. There are scenarios in which they are acceptable due to the solution design (e.g., Big Data, Distributed System).

### 2.1.2 Data Consistency

According to Askham et al. [4], data consistency is when there is no difference between data representations of an entity according to a definition. For instance, a customer's telephone number is stored in three departments' records (e.g., sales, marketing, logistics), obeying the same data definition.

Data inconsistency occurs when the data representations of the same entity differ from each other [20]; for instance, the user updates the customer's telephone number in the sales record but not in the marketing's record. In order to minimize these problems, it is important to organize the data according to the functional dependency between the entity's attributes.

Functional Dependency is a relation between attributes in a data table, in which the values in one or more attributes determine the values of other attributes [11]. For instance, a personal document number determines other attributes, such as name, address, and telephone. The process of organizing the data according to the functional dependency between attributes to avoid redundancy is called Data Normalization [20]. Once the data is stored in a single location and is not duplicated, it also avoids data inconsistency.

Data inconsistency is a well-known problem relate to database integration [5]. This integration may be required in various applications (e.g., Data Science, BI, Management Reports). In some scenarios, the units of an organization need to send data (e.g., CSV files, spreadsheets) to the central level in order to generate information for decision-making; during this process, data inconsistency issues may arise; for instance, these issues have occurred during the recent events of the COVID-19 pandemic, where it was observed disparity in the data reporting at the state and the national levels in India. [46].

### 2.1.3 Uniqueness

Uniqueness is related to an entity not being stored more than once based on identification criteria. [4]. For example, when recording a person's data, it is necessary to verify if this data has already been stored before, considering the national identification document. This concept is closely related to the concept of data redundancy.

According to Elmasri, R and Navathe [11], with respect to relational databases, the use of primary keys distinguishes one record from others in a data table. A primary key is a field or a combination of fields in the data table that uniquely identifies the record [20]; for example, the social security number or the combination of data fields: name, date of birth, and mother's name.

### 2.1.4 Data Completeness

Data completeness is the measure of the number of data elements that are filled compared to the expected total based on the definition. It is about identifying the missing data in a dataset (i.e., values, tuples, attributes) and representing them according to the correct meaning to minimize the impact on results [12].

There are some types of missing data: unknown and inapplicable [12]. The value is unknown because it was not informed during the data collection (e.g., age of a person); and the value is inapplicable according to the business rules related to the entity (e.g., regarding clinical data, the attribute "pregnancy status" is not applicable to male patients).

Missing data can lead to several problems on the results in many applications. For instance, in the healthcare sector, the absence of data regarding the health conditions of people impacts the effective monitoring of the pandemics [9]. In the financial sector, missing values impact the credit decision for a loan [27]. In the e-commerce sector, incomplete data impacts the effective recommendations of products to a customer [18].

There are some situations related to the data collection processes that can lead to missing data. For instance, many data fields in a form can discourage the user from completing the record. Another example is the use of paper-based forms that can also impact the data completeness due to the lack of an effective mechanism to alert when data is missing. This type of form is also associated with illegible records, which also leads to missing values.

### 2.1.5 Data Validity

Data validity is verifying if the data conform to their definition, such as data types (e.g., number, text, and boolean), format (e.g., number of decimal places, date formats), and ranges (e.g., minimum, maximum and allowed values) [4]. Some examples are: the value regarding date of birth must follow the format "MM/DD/YYYY" (i.e., 2 digits for month, 2 digits for day, and 4 digits for year); the value on a person's gender must be included in a list of allowed values (e.g., male and female); the selling price must be in Euros with two decimal places.

Data validity is related to accuracy, completeness, consistency, and uniqueness, according to Askham et al. [4]. The authors also mentioned that it is possible to obtain consistency without validity or accuracy. For instance, the date "1st October 1980" can also be represented in other formats: "1980-10-01", "10-01-1980" and "01-10-1980"; it represents the same data element (i.e., consistency) but in different formats, which is not valid according to the definition of first value (i.e., "1st October 1980").

### 2.1.6 Data Accuracy

Data accuracy is the measure of how data correctly describes the existing entity [4]. For instance, regarding a healthcare scenario, a doctor collects a patient's data on height and weight that was not measured properly, which can impact the monitoring of chronic illnesses (e.g., diabetes, obesity, and high blood pressure).

Inaccurate data can strongly affect results in some domains due to small differences in measurements (e.g., healthcare, automotive, aerospace). Therefore, there are cases where the data may not be suitable for use if they are not fit the minimum level of accuracy that the applications require [4].

As already mentioned, data quality issues can impact the information for decision-making. Hence, it is necessary to perform data cleaning tasks to ensure data quality in order to obtain effective results. The following section provides an overview of the data cleaning topic.

## 2.2 Data Cleaning

This section covers data cleaning concepts, such as phases, approaches, and techniques, in addition to the aspects of extensibility, data privacy, reuse and collaboration. This section provides a theoretical basis for understanding the following sections, which deal with more specific elements and solutions developed by the industry and academic community.

### 2.2.1 Main Concepts

Data cleaning is the term related to tasks for detecting and repairing errors in the data in order to improve its quality for analysis [22, 48]. In the literature, it is also referred as data cleansing, data scrubbing, and data wrangling [48, 25]. Data cleaning can be applied in multiple scenarios: analytical reports, BI, streaming data, Big Data, statistical projects [48, 8].

In data science projects, for instance, data cleaning tasks are usually performed in the Data Understanding and Data Preparation phases of the CRISP-DM methodology (Cross-industry Standard Process for Data Mining) [51]. Regarding data cleaning, the first phase has tasks for identifying data characteristics and detecting data quality problems, and the second has tasks for repairing to get the data ready for analysis.

The data cleaning activities are usually performed in two phases: error detection and error repairing [8]. The first one is related to identifying the quality problems present in the data; while the second is related to remove or transform the erroneous data.

Regarding error detection techniques, the tasks can be done using quantitative and/or qualitative techniques. The first kind is related to statistical methods, for instance, to deal with outliers, while the second is related to descriptive approaches regarding patterns and constraints to identify the values to repair [7]. A process that helps with error detection is data profiling, which analyzes the structure, content, and other characteristics of the dataset to understand it and its meta-data [1].

Error repairing can be done individually by a human (e.g., user repairs the value in an electronic spreadsheet) or can be automated (e.g., a computer executes data transformation scripts) [8]. Although many data cleaning tasks tend to be automated, humans continue managing the process (i.e., specifying, monitoring, and reviewing) to control the results [8].

Regarding the repairing target (i.e., what to repair), there are solutions to repair the data and also to repair the rules (i.e., when the data is considered true, but the quality rules need to be reviewed) [8]. For instance, an expert identifies that according to the data quality rule, a value is erroneous, but in fact, it is correct, and it represents an existing attribute of a real-world entity; in this case, the expert should revise the data quality rule.

Data cleaning operations can be applied at the data field scope (e.g., to clean values on a data field) or at the data record scope (e.g., to remove duplicates records). The operations performed on

data fields seem to be more prone to reuse due to the common data definition between datasets; for instance, two different datasets can contain similar data fields (e.g., name, date of birth, telephone, and city); on the other hand, data records usually consist in different data structures and definitions across datasets.

To summarize the concepts and terms covered in this subsection and provide a relational view of these elements, it is shown a mind map in Figure 2.1.

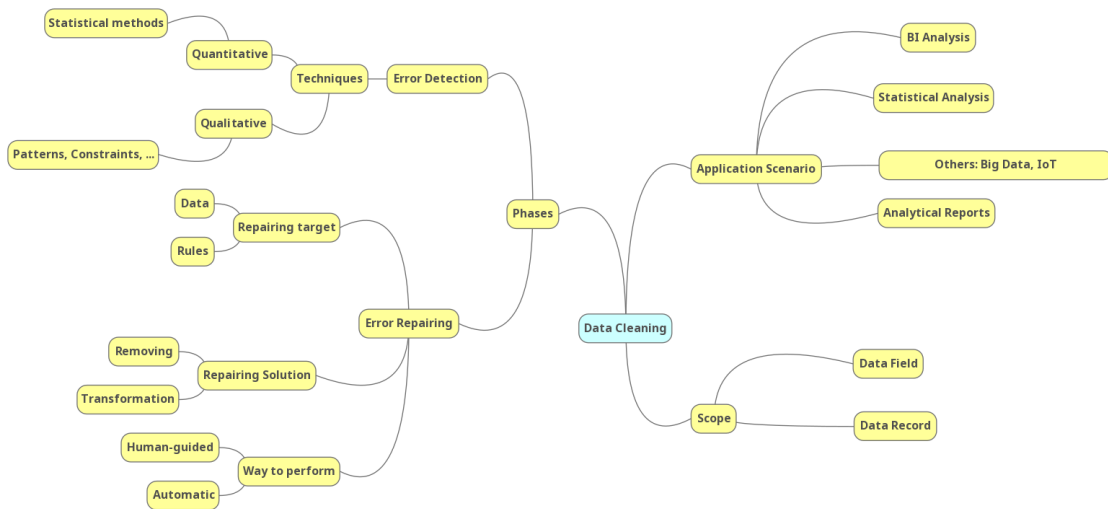


Figure 2.1: Mind map: Main Concepts of Data Cleaning

### 2.2.2 Extensibility

Extensibility is the quality of adding new features to the software product with minimal or no modification on its internal elements [24], which is important for users to adapt the software according to the application's needs. Concerning data cleaning projects, the data structure varies from one dataset to another according to the application; the data cleaning applications must allow adjustments (i.e., to be extensible) according to the scenarios; for instance, in a project about clinical data, the data attribute for the patient's symptom has a list of 7 possible values, and in another similar project, the equivalent attribute considers 9 values; the application should allow the reuse of the data cleaning operations on this attribute from the first project into the second project, without much modification.

### 2.2.3 Data Privacy

Data is considered a very sensible asset in many sectors (e.g., healthcare, finance, government). For instance, a physician must keep confidentiality regarding the treatment of the patient; companies and governments have also classified data that must not be disclosed without proper authorization.

Data privacy ensures that the data shared between parties is only used for its intended purpose, and it will not be disclosed without proper authorization [43]. Data regulations exist in many countries: RGPD (Regulamento Geral de Proteção de Dados)<sup>1</sup> in Portugal, LGPD (Lei Geral de Proteção de Dados)<sup>2</sup> in Brazil, HIPAA (Health Information Privacy)<sup>3</sup> in the United States.

It is common to see data privacy issues regarding data disclosure in many applications: social media [37], IoT [32], e-commerce [15]. Regarding public health, specifically in COVID-19 recent events, for instance, personal data leakage has been identified on government websites [46].

Data can be considered a very sensitive and critical asset for organizations, and activities referring to it must obey compliance rules in order to minimize the risk of non-authorized disclosure. So, data is more likely to be kept inside the organization domain, and actions on sharing data with other external parties can be considered a compliance issue.

Data management solutions must meet the requirements for data privacy to avoid unauthorized disclosure, which can potentially harm individuals and organizations.

## 2.3 Software Reuse and Collaboration for Data Cleaning

There are many possibilities to contribute to the data cleaning field (e.g., Scalability, Streaming, User Experience, Data Privacy) [7, 48]. In this work, the searches were narrowed to the software reuse and collaboration aspects on performing data cleaning operations as a way to find opportunities to contribute.

Some data cleaning operations can be complex and require much effort (e.g., standardize values regarding the locations where a patient lives, remove duplicates without a proper primary key field). In addition, there is a growing demand for data professionals over the years [26]; hence, it is necessary to think of solutions to satisfy the need for data management operations. Software reuse and collaboration aspects can be taken into account for a suitable solution.

### 2.3.1 Software Reuse

According to Sommerville [38], software reuse is a strategy to develop new software based on the use of pre-existent software elements. Some examples of software reuse elements are software libraries and frameworks. The first are collections of resources, such as software functions and graphical components, that are used multiple times during the software development (e.g., Math functions in Java<sup>4</sup>; Bootstrap<sup>5</sup>); the second are collections of concrete and abstract source code that provide standard ways to develop software applications (e.g., Spring Framework<sup>6</sup>, Vue.js<sup>7</sup>). The concept of software reuse has been adopted in industry for many years to reduce effort, time,

---

<sup>1</sup><https://dre.pt/pesquisa/-/search/123815982/details/maximized>

<sup>2</sup><https://www.gov.br/defesa/pt-br/acesso-a-informacao/lei-geral-de-protecao-de-dados-pessoais-lgpd>

<sup>3</sup><https://www.hhs.gov/hipaa/index.html>

<sup>4</sup><https://docs.oracle.com/javase/8/docs/api/java/lang/Math.html>

<sup>5</sup><https://getbootstrap.com/>

<sup>6</sup><https://spring.io/projects/spring-framework>

<sup>7</sup><https://vuejs.org/>

and risk; it provides more quality in software projects, as components used in a project were validated and improved in previous ones, getting maturity over time, according to Sommerville [38].

There are also examples of reuse applied to the data management field. For instance, some international and national databases (e.g., classification of diseases, demographic statistics, local infrastructure) are applied in data management projects to standardize values before generating information. To share these databases, many organizations have adopted the concept of Open Data [6]. Governments and other organizations have adopted the concept to share data among parties, providing more transparency to their actions, and, in some cases, ways to collaborate [6]. Furthermore, there are also initiatives to reuse the knowledge on data cleaning operations, such as an ontology-based solution [2], which will be discussed in the next section.

### 2.3.2 Collaboration

Collaboration provides a way to do things better or allowing to achieve goals that would not be possible by a single individual. When it is applied to data cleaning, it can reduce the effort and time in performing these activities.

In software engineering, collaboration is a common aspect since there are many software projects with many engineers working together [49]. This aspect is present in many software activities (e.g., modeling, coding, and testing), and its application has become broader over the years (i.e., people collaborating from different locations around the world). There is a variety of tools that support the collaboration in this field (e.g., JIRA<sup>8</sup> Confluence<sup>9</sup>, and Github<sup>10</sup>). Some initiatives support the data cleaning process, such as CoClean[30], which will be discussed in the next section.

The list below summarizes some points on the reuse and collaboration applied to data cleaning that can be taken into consideration as opportunities for future initiatives:

- The lack of data experts with skills in data processing, statistical tools, or/and programming languages to satisfy the increasing demand;
- There are data cleaning operations that need a huge amount of effort to do for one single dataset, which can be applied to other datasets;
- A type of data quality problem solved by a professional can be the same as other professionals;
- An data expert can share his/her data cleaning knowledge with non-technical people.

There is a belief that aspects of software reuse and collaboration can also improve the awareness and communication on common data quality issues and provide innovative ways to solve

---

<sup>8</sup><https://www.atlassian.com/software/jira>

<sup>9</sup><https://www.atlassian.com/software/confluence>

<sup>10</sup><https://github.com/>

them in a broader context, involving people from different expertise, knowledge, and locations to work together on data cleaning solutions.



## Chapter 3

# State of the Art

This chapter presents the state-of-the-art of data cleaning solutions related to reuse and collaboration aspects. Some data cleaning tools are presented, as well as related works that have been cited in recent publications, along with three comparative analyses. In the end, a gap analysis is presented to identify opportunities for contributions.

This work carried out the literature review, the solution proposal, and a proof of concept, which implements and validates the solution. The literature review analyzes data cleaning tools available on the market and related works to identify how they address the concepts of reuse and collaboration to obtain the research gap that provides direction for the development of this work.

### 3.1 Data Cleaning Tools

There are tools to support data cleaning activities that are commonly mentioned in recent works, such as Trifacta Wrangler<sup>1</sup>, OpenRefine<sup>2</sup>, and TIBCO Clarity<sup>3</sup> [33, 52, 44]. The term "Data Cleaning Tool" in this work refers to specific tools for data cleaning that exist on the market, regardless of whether they are open source, licensed, SaaS (Software as a service), or free to use. Two comparative analyses are presented; the first is regarding common features of data cleaning activities (e.g., error detection and correction approaches), including aspects of reuse and collaboration, extensibility, and data privacy; the second analysis focuses on specific features for solving data quality issues.

#### 3.1.1 Comparison Based on Main Features

This subsection presents a comparison of the data cleaning tools based on the features: error detection, error repairing, extensibility, data privacy, reuse, and collaboration. Table 3.1 shows

---

<sup>1</sup><https://www.trifacta.com/>

<sup>2</sup><https://openrefine.org/>

<sup>3</sup><https://clarity.cloud.tibco.com/>

the comparison that is discussed in this section.

Item	Feature Group	Feature	Trifacta Wrangler	OpenRefine	TIBCO Clarity
1	Data Validation	Data Profiling	Detailed	Simplified	Detailed
2		Meta-data Definition	-	-	X
3		Patterns for Validations	X	-	-
4		Validation Report	-	-	X
5	Data Repairing	Predefined repairing actions	X	X	X
6		Remove records	X	X	X
7		Transform values	X	X	X
8		Identify issues for human analysis	X	X	X
9	Extensibility	Define operations by programming	X	X	X
10		Programming Language	Wrangle Language	GREL, Python, and Clojure	GREL
11	Reuse	Reuse operations in the same project	X	X	X
12		Reuse operations among projects	X	-	-
13		Import / Export operations	X	X	X
14		Reuse among different platforms	-	Platforms compatible with GREL, Python and Clojure	Platforms compatible with GREL
15	Collaboration	Allow multiple users	X	-	-
16		Share data among users	X	-	-
17		Share operations among users	X	-	-
18	Data Privacy	Where data is processed	External	Local	Local (Premium version); External
19		Access Control	X	-	X

Table 3.1: Comparison of the Data Cleaning Tools based on Main Features

**Error Detection** It refers to detecting errors and anomalies in the data based on an evaluation criterion (e.g., constraint, rule, pattern, or expression), statistical methods, and also through human analysis.

TIBCO Clarity has a mechanism of data validation based on schema definition, which allows generating a validation report to identify values that do not meet the criteria based on a definition of data validity criteria, such as type (e.g., text, number, and date), length for text fields, data range,

and format. Trifacta Wrangler allows defining patterns for detecting errors by using its own DSL (Domain-specific Language) called Wrangle Language.

As for data profiling, Trifacta Wrangler and TIBCO Clarity provide a detailed data profiling report that identifies the number of mismatched values, missing and unique values, which depends on the field data type and defined patterns for detection. OpenRefine has a simplified report based on facets (i.e., a mechanism for selecting values in a dataset to perform data cleaning operations) applied to data fields, which shows the unique values and the number of records they appear.

**Error Repairing** Once errors are detected in the error detection phase, it is necessary to perform actions for removing, transforming, or even selecting the records for careful human analysis later.

All three compared tools provide options to repair the data based on predefined repairing actions (i.e., removing or transforming values), and also marking the records for a more careful analysis.

**Extensibility** There are situations where the validation and repairing operations available in the tools are not enough for specific domains. Hence, it was observed if the tools allow specifying new data cleaning operations based on programming languages. Users usually define their functions (i.e., UDF – User Defined Functions) according to application needs.

The tools allow the definition of specific data cleaning operations through programming languages. Trifacta Wrangler allows coding operations in Wrangle Language (i.e., Trifacta DSL). OpenRefine and TIBCO Clarity allow defining specific operations in GREL (i.e., General Refine Expression Language), but OpenRefine also allows coding operations in Python and Clojure programming languages.

**Reuse** During the data cleaning activities, users usually define some domain-specific operations to detect and repair the data. It is observed if the tools allow reusing these operations using copy-and-paste and/or export/import functionalities, in addition to a systematized way to reuse.

All three compared tools allow reuse operations in the same data analysis project. Also, the import and export of operations were possible in all analyzed tools. The reuse of operations among different projects in a systematized way is available in Trifacta.

Regarding the reuse of data cleaning operations among different platforms, it was observed that tools allow the reuse among platforms that use the same language (e.g., Python, GREL). For instance, OpenRefine and TIBCO Clarity can reuse operations from each other, once they execute operations in GREL. OpenRefine also allows reusing code in Python and Clojure languages from compatible platforms.

**Collaboration** Regarding the aspect of collaboration in this analysis, it was considered: sharing the data to clean or sharing the data cleaning operations to perform in different datasets.

Trifacta Wrangler is the only one out of the compared tools that allowed the collaboration among users to share the data to clean and the operations; the users must be authenticated in the

organization profile and have permission to access the data cleaning project. OpenRefine and TIBCO Clarity are considered single-instance applications, and they do not provide authentication and access control for systematic collaboration.

**Data Privacy** The collaboration among users can bring data privacy concerns in case the data is shared without proper authorization. It was observed if the tool can expose the users' data when performing data cleaning operations (e.g., data processing locally or remotely, access control during data sharing operation).

In order to clean the data using Trifacta Wrangler or TIBCO Clarity, the user needs to send the data to an external server, crossing the organization's boundaries. Although those two applications provide access control, this situation can make some data privacy concerns arise. TIBCO Clarity has a premium version that can be installed on the organization's server. OpenRefine runs locally and does not need to send data to other servers; although it was developed in web technologies, it is executed in a web application server that runs on the local machine.

### 3.1.2 Comparison based on Data Quality Issues

This section compares the data cleaning tools, already mentioned in the previous section, based on data quality issues, which are present in chapter 2. It was defined some types of these issues that the tools should handle during the data cleaning activities.

To perform a more informative analysis of the selected tools, besides analyzing their technical specifications, a simple empirical analysis of their functionalities was carried out based on a simplified and fictional dataset regarding case investigation in an outbreak monitoring. This dataset is tabular data in CSV format, which is available in the appendix, Figure A.1.

Table 3.2 presents the comparison of the tools based on six types of data quality issues that are described next.

**Missing Values** It was observed if the tools allow to define a data field as mandatory to alert if there are missing values.

All compared tools were capable of detecting missing values, and they also allow repairing by removing the records or transforming to a specific value. TIBCO Clarity was also capable of defining a specific data field as mandatory using meta-data (i.e., based on schema definition), which supported to focus on the specific ones instead of verifying all the data fields.

**Type-mismatch Values** The tools can provide basic data types (e.g., text, number, boolean) or domain-specific data types (e.g., person's nationality, person's gender, city of residence), which can be defined by users and reused in other situations. Once the data field is associated with the data type, the tool can verify if values are in the corresponding data domain.

All tools could define preset data types (e.g., telephone number, person's name), and they also allow detecting mismatched values. The feature of repairing using regular expression was present in all tools, but Trifacta Wrangler identifies patterns in values supported by AI algorithms.

Item	Data Quality Issue	Feature	Trifacta	OpenRefine	Clarity
1	Missing Values	Define a field as mandatory	-	-	X
2		Detect missing values	X	X	X
4		Repair by removing the record	X	X	X
5		Repair by setting to a specific value.	X	X	X
6	Type-mismatch Values	Define preset data types (e.g., number, text)	X	-	X
7		Define specific data types	-	-	X
8		Detect mismatched values	X	X	X
9		Repair using regular expression or patterns	X	X	X
10		Repair using AI support	X	-	-
11	Incorrect dates	Detect date type and format	X	X	X
12		Repair by transforming into preset date formats	X	X	X
13		Repair by transform-by-example method	X	-	-
14	Out-of-range Values	Use of Mapping tables	X	-	X
15		Repair by replacing values	X	X	X
16		Repair by Transform-by-example Method	X	-	-
17	Duplicate Records	Detect duplicate records by the unique fields	X	X	X
18		Repair by removing the record	X	X	X
19	Functional Dependence Violation	Define rules of functional dependence	X	X	X
20		Detect records that violate the defined rules	X	X	X

Table 3.2: Comparison of tools per data quality issues

In addition, Wrangle Language (i.e., Trifacta DSL) allows the extensibility of specific repairing programmatically using patterns.

**Invalid Dates** Regarding a data type to represent dates in a dataset, it is usually necessary to deal with different formats, locations, and time zones. It is observed if the tool allows defining these characteristics and then detect mismatched values.

All tools detect the data types for dates, and they also allow repairing by transforming into preset date formats. Trifacta Wrangler also allowed applying the transform-by-example method that was useful to repair when there are many formats of dates present in the data field.

**Out-of-range Values** This issue occurs when values are not within a set of allowable values. In this work, it was preferred to distinguish the issues of type mismatch from out-of-range values, as

the first is related to the data structure, and the second focused on the data content. It was analyzed the tools' mechanisms of error detection and repairing related to this issue.

All compared tools have functionalities for replacing erroneous values. Trifacta Wrangler and TIBCO Clarity support the use of mapping tables to standardize values in data fields. On OpenRefine, the mapping-tables functionality assisted by UI (User Interface) was not identified, but it could be done programmatically. Trifacta Wrangler provided transform-by-example functionality, which reduces the effort by applying the repair operations in more than one record.

**Duplicate Records** In order to analyze how the tools deal with duplicated values, it was observed if it is possible to define one or more fields as unique identifiers for detecting duplicate records. As for the repairing action, it was observed if the tools allow removing the records or selecting them for a human analysis later.

All three tools allow detecting duplicates, but TIBCO Clarity allows defining more than one data field to combine and use for duplicates detection. The data profiling option on Trifacta Wrangler and TIBCO Clarity has information about uniqueness for the data field and how many records have the specific value. A common repairing option in all compared tools was to remove one of the duplicates.

**Functional Dependency Violation** It was observed if the tools allow defining constraints based on functional dependency between fields (e.g., pregnancy status and person's gender) and then detect the records that do not obey these constraints. As for the repairing action, it was observed if the tool selects the records for human analysis.

All compared tools supported the detection of the relationship between fields in the dataset using programming scripts. It is possible to detect and select the records and then decide the action to repair (e.g., replace the erroneous values or remove the records).

## 3.2 Related Work

This section presents related works from academia and industry in the data cleaning field concerning collaboration and reusing aspects mentioned in recent publications, aiming to identify the state of the art, directions, and opportunities to contribute.

The following works usually focus on specific issues to solve that differ from tools available in the market, which are more mature and typically allow tackling various situations of the data cleaning tasks.

**CoClean [30]** The solution supports the collaboration on data cleaning activities by allowing users to clean the same dataset. It provides a Python library called Collaborative Data frame (CDF). The collaboration starts when the owner of the dataset shares a link with the users. Power

users (i.e., expert users) can collaborate by accessing the dataset and define data cleaning operations in Python language through the library API. The solution also provides a Web UI (Figure 3.1) for the non-technical users, which can collaborate by doing the manual cleaning.

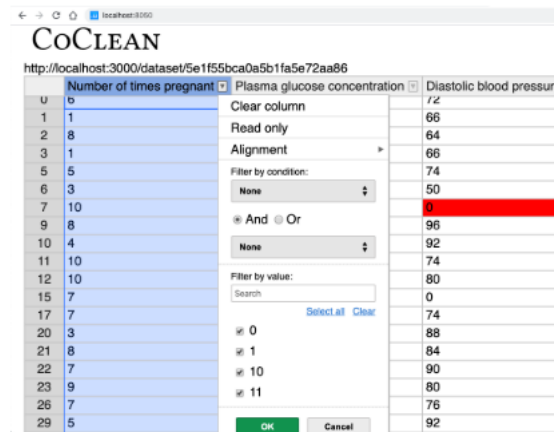


Figure 3.1: CoClean: Snapshot of the Web UI for data cleaning [30]

**Transform-Data-by-Example [17]** Transform-Data-by-Example (TDE) focuses on data transformation to solve data quality problems (e.g., Standardizing values). The solution receives a few pairs of input/output examples and synthesizes programs based on transformation logics that exists in source code repositories (e.g., GitHub). The Figure 3.2 shows an example of TDE UI.

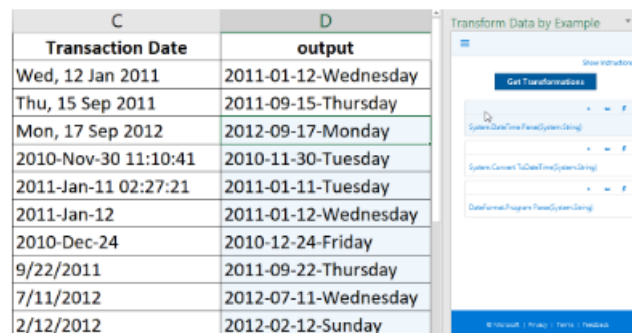


Figure 3.2: TDE: Snapshot of the UI for values transformation [17]

**Auto-Transform [23]** The solution allows transforming data based on input/output of data patterns, without having to specify examples, as in TDE. It can define these patterns based on the analysis of existing paired tables (e.g., data tables in wiki pages). Figure 3.3 shows examples of patterns used in the data transformations.

**Potter's Wheel [35]** The tool provides an interactive mechanism for data cleaning tasks, which allows the user to apply data repairing operations gradually and see the results on the screen

TBP-id	Source-pattern ( $P_s$ )	Target-pattern ( $P_t$ )	( $T$ )
TBP-1	<letter>{3}. <digit>{2}, <digit>{4}	<digit>{4}-<digit>{2}-<digit>{2}	...
TBP-2	<digit>{3}) <digit>{3}-<digit>{4}	<letter>{3}-<digit>{3}-<digit>{4}	...
TBP-3	(<digit>+<num>'<letter>{1}, <digit>+<num>'<letter>{1})	<letter>{1}<digit>+<num>' <letter>{1}<digit>+<num>'	...
...	...	...	...
TBP-7	<digit>{4}/<digit>{2}/<digit>{2}	<letter>{3} <digit>{2}	...
TBP-8	<num> kg	<num> lb	...
TBP-9	<num> lb	<num> lb <num> oz	...
...	...	...	...
TBP-15	<num> kg	<num> 公斤	...
TBP-16	<letter>+ de <digit>{4}	<digit>{4}	...
...	...	...	...

Figure 3.3: Autotransform - List of Source and Target Patterns for Data Transformation [23]

at once. It allows the user to specify specific data domains, which allows detecting mismatched values based on checking of constraint violations. It comes with default data domains (e.g., String, Integer, Money), and the users can define custom ones as needed. The Figure 3.2 shows an example of TDE UI.

Delay	Carrier	Source	Dest..	Date	Day	Dept_Sch	Arr_Sch
-12	TWA	JFK	STL	1997/10/17	F	17:30	19:18
0	TWA	ORD	STL	1997/07/28	M	12:25	13:36
-26	TWA	JFK to MIA		1998/12/04	F	09:40	12:40
-3	TWA	JFK to MIA		1997/12/30	Tu	07:30	10:36
-5	TWA	ORD	STL	1997/06/08	Su	15:05	16:17
2	TWA	JFK	MIA	1998/09/21	M	07:25	10:25
3	TWA	ORD	STL	1998/07/02	Th	11:20	12:30

Figure 3.4: Potter's Wheel: Snapshot of the UI with a dataset loaded [35]

**Frictionless Data [14]** This project provides a set of products (i.e., tools, specifications, best practices guides) focused on helping data professionals deal with data preparation. For this comparison analysis, it was considered the product Table Schema, which allows defining a data schema for the dataset that can be used later for data validation.

**An Ontology-based Methodology for Reusing Data Cleaning Knowledge [2]** This solution is a methodology for defining and reusing data cleaning operations based on ontology concepts and technologies. The authors proposed a conceptual layer and a concrete layer. In the first layer, the user, as a Domain Expert, defines the domain of interest and the conceptual elements for data cleaning operations (i.e., vocabulary and rules) using an ontology (RDF/OWL). In the second layer, as an IT Expert, the user defines the concrete operations for data cleaning, taking into account the technology used to implement the data source (e.g., MySQL, MongoDB). For the sake of brevity, this work will be referred to as the ontology-based solution.

**Federated Data Cleaning [29]** Federated Data Cleaning (FedClean) is an application protocol that allows the data cleaning in edge nodes regarding IoT applications without centralizing or



Tools	Error Detection	Error Repairing	Extensibility	Usability	Reuse	Collaboration	Data Privacy	Platform
CoClean [30]	Human-Guided	Human-Guided	Not Applicable	Non-technical User, Technical User		Collaboration	Data Sharing	Web UI, Python
Auto-Transform [23]	-	Transform-by-patterns approach	Error Repairing, Data Patterns source	Technical User	Reuse of Transformation		Local Data Processing	-
Transform Data by Example (TDE) [17]	-	Synthesized transformation logics from various sources	Error Repairing, Source Code Repository	Non-technical User	Reuse of Transformation	-	Remote Data Processing	MS-Excel (add-in)
Potter's Wheel [35]	Data Domains (Constraints)	Human-Guided	Error Detection, User Defined Domains	Non-technical User	Reuse of Error Detection and Error Repairing	-	Local Data Processing	Desktop GUI (Java Application)
Frictionless Data (Table Schema) [14]	Table Schema (Constraints)	-	User Defined Types (Constraints)	Technical User	Reuse of Error Detection	-	Local Data Processing	Libraries (Python, JavaScript, Java, and more 6 platforms)
An Ontology-based Methodology for Reusing Data Cleaning Knowledge [2]	Vocabulary, Rules, Code	-	Support Various Domains and platforms	Technical User	Reuse of Error Detection	-	Share Data Cleaning Knowledge, Not Data	-
Federated Data Cleaning [29]	Abnormal Values using AVF	Human-Guided	Not Applicable	Technical User	-	Collaboration of nodes	Data sharing with privacy preservation	Application Protocol

Table 3.3: Comparison of the Related Works

exposing the data to clean, ensuring data privacy in the process. This solution uses boolean shares (i.e., values based on XOR operation to share a variable secretly) of the collected data in the edge points, which are sent and processed on non-colluding servers (i.e., servers that run a protocol to obtain the results without revealing the inputs); it is used AVF (Attribute Value Frequency), which is an outlier detection technique, to detect the abnormal values without exposing them.

### 3.2.1 Comparison Analysis of the Related Works

This section presents a comparison of the related works regarding common features of the data cleaning process (e.g., error detection and error repairing), and the aspects of reusing data cleaning elements and collaboration among users, in addition to data privacy, target audience, and available platforms. Table 3.3 shows the comparison, which is described in the following subsections.

#### 3.2.1.1 Error Detection

As mentioned in Chapter 2, error detection is related to detecting anomalies or errors in the data according to a criterion. It is usually related to the application, with specific data elements to be validated (e.g., data fields: date of symptom onset, and patient status), but there are also general data elements that can be reused in other contexts (e.g., data fields: name, gender, date of birth).

Some solutions have different characteristics concerning error detection. CoClean [30] considers the human-guided approach for error detection, although it also allows automating through programming scripts; and Federated Data Cleaning [29] detects abnormal values (e.g., outliers) without exposing the original values.

Other solutions (Potter's Wheel [35], Frictionless Data [14], ontology-based solution [2]) have similar characteristics regarding using rules to detect data quality issues in the application. These elements support the automation of common issues, allowing the user to focus on more specific situations. Potter's wheel detects discrepancy by applying data domains to the values (e.g., number, text, word); the values that violate the data domain rules are pointed out. Frictionless Data allows defining table schema, where each data field has a set of constraints. The Ontology-based solution uses abstract and concrete elements for error detection (i.e., vocabulary, rules, source-code).

Some solutions do not deal specifically with error detection. Auto-transform [23], and TDE [17] are focused on data transformation (i.e., one of the forms of data repairing), and do not provide features for detecting errors (e.g., error reporting)

It was observed that the solutions do not automate completely the operations for detecting errors. The user continues to be in charge of perceiving specific data quality issues to be solved.

### 3.2.1.2 Error Repairing

As already mentioned, error repairing is typically performed after error detection. The former can also be very related to the application, thus some solutions (CoClean [30], Potter's Wheel [35], Federated Data Clean [29]) consider the user-guided approach during the moment of repairing to apply specific operations according to the domain at hand.

Some compared solutions are based on data transformation to fix the erroneous values: Auto-Transform [23] allows automating the data repairing activity based on transformation patterns, and TDE [17] uses an approach based on synthesized transformation programs from a variety of sources (e.g., source code repositories, mapping tables).

Frictionless (Table Schema)[14] and the ontology-based solution [2] do not consider data repairing as part of their scope, and they cover only error detection. The latter mention that it is possible to deal with data repairing using ontology-based methodologies, which was pointed out as future contributions.

Regarding the analyzed solutions, although there is a direction to develop automated tools for data repairing, other solutions consider the human-guided approach for obtaining more effective results since there are specific situations that need more careful analysis performed by humans.

### 3.2.1.3 Extensibility

As data cleaning activities are performed on datasets according to the application, they should support user-defined data types and scripts (i.e., customized operations) to make the operations suitable for the application at hand.

CoClean [30] and FedClean [29] are solutions focused on collaboration among resources involved in the data cleaning process, and so they do not provide a set of operations to be extended, although they allow the use of other programming tools (e.g., PANDAS<sup>4</sup>) to support data analysis. Thus, the characteristic of extensibility was considered "Not Applicable" for these solutions.

---

<sup>4</sup><https://pandas.pydata.org/>

Auto-transform [23] allows extensibility by receiving sources of data patterns (e.g., Wikipedia tables) to "learn" and synthesize a pattern of data transformation. TDE [17] allows connecting to different code libraries and then indexing the transformation operations available in there. In both cases, there is the possibility of adding new transformation logic by providing additional repositories.

Potter's wheel [35] supports a user-defined domain (i.e., data constraint) which extends the available set of discrepancy detection constraints. The ontology-based solution [2] supports various domains (e.g., Finance, Health) and platforms (e.g., SQL, MongoDB); it allows defining specific elements (i.e., vocabulary, rules, code) for error detection according to the application. In both solutions, it is observed that the user has a high level of extensibility to support the application at hand.

Extensibility is a common feature in most compared solutions; given that data cleaning operations are usually related to specific applications' datasets, these operations need to be developed or adapted to meet the application's needs.

#### 3.2.1.4 Usability

It was identified the target audience of the solutions, which can be a technical-user or a non-technical user. A technical-user is the one who is able to apply programming languages and/or data management knowledge to perform data cleaning operations. A non-technical user is the one who is able to clean the data using only basic skills in web and office applications (e.g., spreadsheets).

The solutions more suitable for technical users are Auto-Transform [23], Frictionless Data [14], Ontology-based solution [2], and Federated Data Cleaning [29], since it is necessary technical skills (e.g., programming language, data management, application protocol) to apply them. The solutions CoClean [30], TDE [17], and Potter's Wheel [35] are more suitable for non-technical users because they allow cleaning data using a graphical user interface (GUI). CoClean can be used by technical users as well because it provides an API to develop solutions using a programming language (i.e., Python).

It was observed that the majority of the analyzed solutions focused on technical users. These solutions allow the use of programming languages and software components to create new solutions that usually solve common quality issues present in the data. However, it is also important to provide solutions for non-technical users to allow cleaning more-specific issues and the engagement of others in the data cleaning process (e.g., domain expert).

#### 3.2.1.5 Reuse

As mentioned, reuse can help to reduce the effort on defining data cleaning operations given that some data quality issues (e.g., missing values, out-of-ranged values) are common in different applications. This criterion aims to identify data cleaning solutions that provide a structure or mechanism to reuse operations.

Some solutions seem to allow only the reuse of data transformation to perform the error repairing operations: Auto-transform [23] and TDE [17]. The former can reuse and synthesize transformations based on input/output data patterns, which the authors define as Transform-by-Patterns (TBP); the latter allows the reuse of transformation logic from the existing base of source code.

Potter's wheel [35] allows the reuse of both types of data cleaning operations (i.e. Error Detection and Error Repairing). This solution provides data domains (e.g., Strings, Integer, Money) for error detection based on constraint validation and also common actions (e.g., Merge, Format, Split) for error repairing.

Some solutions only allow the reuse of error detection logic. Frictionless Data (Table Schema) [14] allows the reuse of error detection based on data schema validation. The ontology-based solution [2] was designed to allow the reuse of error detection and error correction logic across domains and platforms (e.g., SQL and Non-SQL databases). This solution is more focused on error detection, although the authors mentioned the intention of future works to develop the error correction part.

The solutions CoClean [30] and Federated Data Cleaning [29] do not focus on reusing data cleaning logic, and they address issues on collaboration among elements involved in the data cleaning process; nonetheless, they do not make the reuse unfeasible, which can be addressed by combining these tools with others. It was observed that the solutions cover the needs of reusing data cleaning operations (i.e., error detection or error repairing) partially. Few solutions allow the reuse of error repair logic, but only using data transformation. Only one of them covered the reusability of both types of operations.

### 3.2.1.6 Collaboration

This criterion regards the possibility of users collaborating on cleaning data in order to reduce the individual effort and time. Only two analyzed solutions allow the collaboration on the data cleaning process: CoClean [30] and Federated Data Cleaning [29].

CoClean [30] allows collaboration by sharing the data to clean with other users. Multiple users can perform the data cleaning operations on the data through an API, using a version control mechanism that allows the dataset owner to decide what updates in data will be applied to the final result. The solution also provides a way for lay-users to collaborate through a graphical interface, similar to an electronic spreadsheet.

Federated Data Cleaning [29] allows the collaboration of edge nodes (i.e., devices) to clean data locally, preserving data privacy. This solution takes a different direction than other approaches that clean data in a centralized node, which can expose the data during the process.

Regarding the other solutions, a well-developed or systematized mechanism was not identified to support the collaboration in data cleaning. CoClean [30] addresses the collaboration aspect in a feasible and structured way; however, since the solution focuses on the interaction among users involved in the process, sharing data can bring some data privacy concerns, mainly when dealing with sensitive data (e.g., clinical data).

### 3.2.1.7 Data Privacy

As it was already mentioned, a concern that usually comes to mind when there is collaboration among people to do data cleaning operations is privacy preservation, especially when dealing with sensitive data. This criterion aims to identify features of data privacy present in the analyzed solutions.

Some solutions address data privacy issues by avoiding sharing data with other users. The solutions Auto-Transform [23], Potter's Wheel [35], and Frictionless Data [14] allow the data cleaning operations to be performed locally without sending the data to an external application server.

Other solutions allow performing data cleaning using different nodes. TDE [17] uses a back-end service deployed on a cloud platform for performing the data cleaning, which can bring some data privacy concerns once part of the data is sent to the service; the paper regarding the TDE solution does not address further details about data privacy. CoClean [30] allows sharing the data to clean with other users by sharing a URL to access, which can bring data privacy concerns, especially when sensitive data is used.

Regarding the solution based on ontology [2], the related paper does not address data privacy issues explicitly. The solution deals with data cleaning knowledge, not data to clean, which can be considered a promising approach to reduce the risk in solutions that involves collaboration. The possibility of sharing operations and other definition can be a feasible contribution to reduce the risk in data exposing.

FedClean [29] allows the detection of abnormal values without centralizing or exposing the data. The study was done considering the IoT domain and outlier detection technique, but it seems that it is not easily applicable to other domains and other types of data quality issues.

It was observed that most of the solutions do not maturely address the data privacy aspect. Some solutions avoid sharing data to reduce the risk of unauthorized exposure; others do not consider security mechanisms when data is transmitted between the parties; only one solution defines means of privacy preservation during the data cleaning process. Something that drew attention was the possibility of sharing knowledge in data cleaning instead of the data itself to reduce the effort and time on data cleaning activities, reducing the risk of unauthorized data exposure.

### 3.2.1.8 Available Platform

This criterion considers if the solutions are broadly available to use and in which type of artifact they are provided (e.g., web and desktop applications, programming language libraries, and application protocols).

The following solutions are available on desktop and/or web platforms: CoClean, TDE, and Potter's wheel. The first has a graphical interface on a WEB platform and an API library; the second was made available as an Excel add-in; the third was developed as a Java desktop GUI application.

Regarding the solutions available through libraries of programming languages and application protocol, the following were identified: Frictionless Data (Table Schema), CoClean, and Fed-Clean. The first is available in 10 different platforms (e.g., Python, R, JavaScript, and Java); the second is available as a Python library and as a Web Application, as already mentioned; and the third through an application protocol.

Some solutions do not refer to the targeted technologies: The ontology-based solution and Auto-transform. The first is a proposed methodology; the specific technologies to implement were not described in the related paper. Auto-transform was not identified as a product available for use.

The majority of the solutions are provided as artifacts for end-user (i.e., web or desktop applications). Other solutions are available as software developing components that can be used for creating new solutions.

### 3.3 Gap Analysis

The analyzed data cleaning solutions were grouped into data cleaning tools and related works. The former supports many types of data cleaning tasks and is widely available to users. The latter support specific types of tasks, and not all are available for use. The related works were classified according to the most relevant aspect they address, as shown in Figure 3.5.

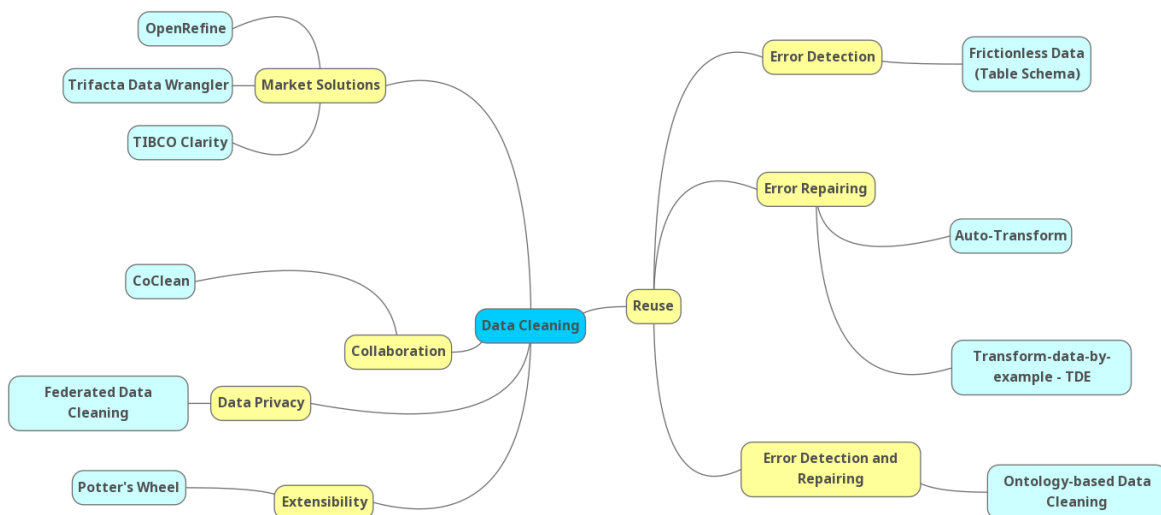


Figure 3.5: Mind Map of the Compared Solutions

Regarding the reuse aspect, the tools allow the sharing of data cleaning scripts in various ways (e.g., copy-and-paste and export/import methods), and one of them allows systematic reuse within a workgroup. The direction of the ontology-based solution seems promising; however, it would be necessary to carry out studies on implementations in different scenarios to assess usability.

Regarding the collaboration aspect, a promising direction is collaboration in sharing data cleaning operation scripts broadly, not restricted to the domain of the organization or workgroup, which could increase the possibility of reusing these scripts within a more significant number of people.

Finally, given there are some kinds of data quality issues that are common across domains and applications, it can be pointed out a possible direction to a broader collaboration on reusing data cleaning operations with people working together, regardless of organizations, roles, and expertise levels. That kind of collaboration focused on operations, not data, can also contribute to avoiding data privacy issues once the data is not shared among parties.





# Chapter 4

## TruData

According to the state of the art regarding the data cleaning field presented in the previous chapter, there is a possibility to contribute through a solution to support the reuse of data cleaning operations and the collaboration between professionals involved in the process. This chapter presents the proposed solution, describing the overview and software modelling perspectives

### 4.1 Solution Overview

TruData consists of a platform to support the broad reuse of data cleaning operators (i.e., programming script for cleaning data) from different platforms (e.g., Python, R, and Excel), data types (e.g., text, numbers, dates), domains (e.g., healthcare, finance, education), making the data cleaning tasks more efficient and effective, and also more pleasant. The solution catalogs data cleaning operators developed by a user and makes them available to others through the Internet. Users provide usage data (i.e., ratings, comments), which can be used to improve operators and optimize search results. Figure 4.1 shows the overall idea of the solution.

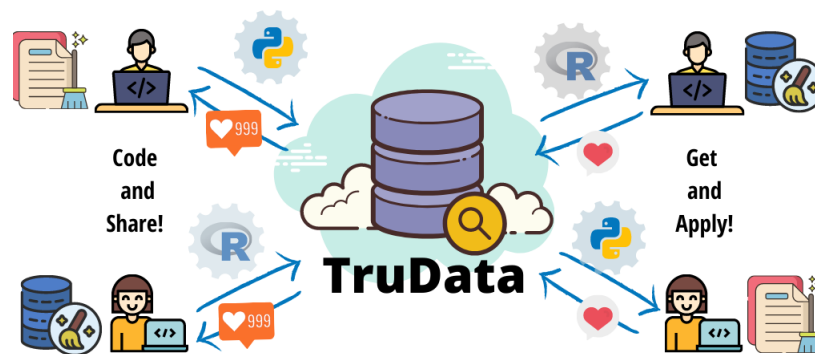


Figure 4.1: Overview of the solution TruData

The cataloging process is done by a contributor, providing the address of the source code repository where the operator is located and also defining related elements on data cleaning and application domain, such as data type (e.g., Text, Number, Date), platforms (e.g., Python, R, Excel,

OpenRefine) and data quality anomalies (e.g., Outliers, Out-ranged-values). Figure 4.2 shows a representation of the operator and its related elements.

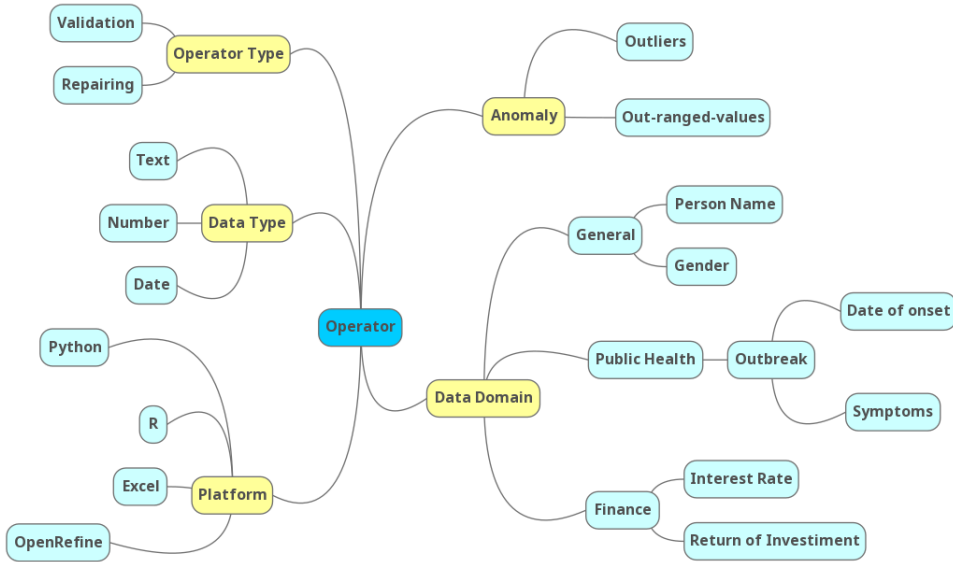


Figure 4.2: Representation of data cleaning operator and related cataloging elements

After cataloging, the operator is available on the platform for consumers to obtain and apply in data cleaning tasks. The simplified view of cataloging and getting an operator is shown in Figure 4.3. The following steps describe the process: The contributor catalogs the operator in the platform (step 1); the consumer search for an operator (step 2); the platform makes a request to the code repository (step 3), which in turn, respond with operator and other information (e.g., an example of use) (step 4); finally, the platform sends the operator to the consumer (step 5).

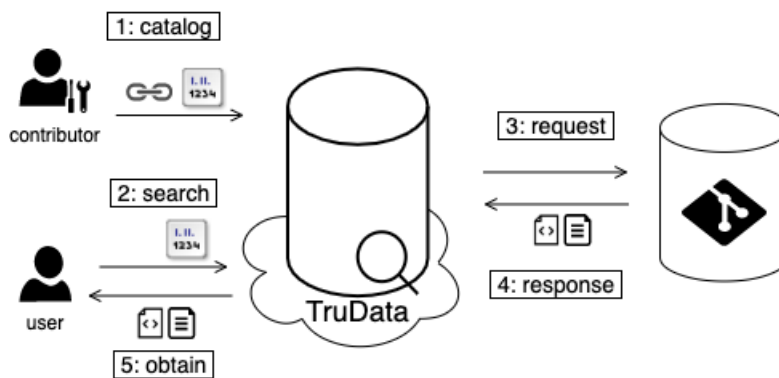


Figure 4.3: Simplified functional view of the solution

A Data Cleaning Operator (DCO) can detect errors or repair values in a dataset to satisfy Data Quality Requirements (DQR), which was also considered in TruData, representing the need for data quality according to the application domain. For instance, regarding the public health domain, a data field related to disease symptoms must be filled only with the numerical values: 0 and 1.

Figure 4.4 shows a representation of the data requirement and its related elements, which are the data type (e.g., text, number, and dates) and data domain (e.g., general, public health, and finance).

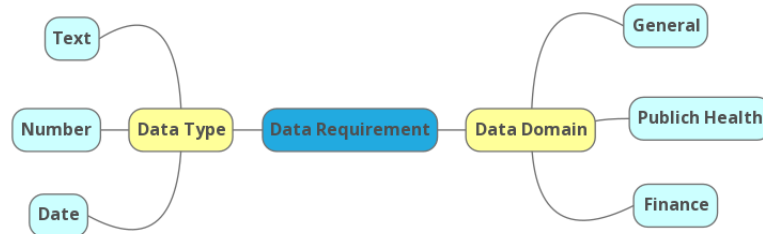


Figure 4.4: Representation of data requirement and related elements

The solution also considers the social aspect of sharing data cleaning operators by encouraging collaboration between people to build and improve them. For instance, a user that obtains an operator also evaluates and comments about the user experience, so others can consider it before deciding to use this operator.

One of the solution's main features is the broad collaboration between users to define a set of related operators (DCO) to satisfy the need for data quality (DQR). Making an analogy with a game of blocks, the blocks (i.e., DQR and DCO) can be associated with each other to achieve a goal: to create a solution that solves a data quality problem according to the application. The community creates and manages the associations between the elements (i.e., DQR and DCO), that is, users indicate and evaluate the associations, which can be strong or weak, depending on the result of the evaluation.

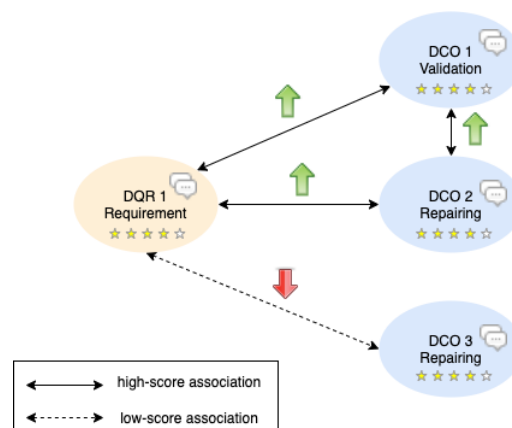


Figure 4.5: Association between DQR and DCO assisted by social features

A user can also indicate an association between similar or complementary operators. For instance, an operator that identifies errors is related to another one that repairs these errors, building a chain of elements involved in cleaning the values of a particular data domain (e.g., person gender, patient symptoms, patient location). Other users can evaluate the associations, providing a

numerical mark (from 1 to 5) that allows calculating the average score; associations with a high score are considered more relevant than others.

Figure 4.5 shows an example of possible associations between elements. The Data Quality Requirement element (DQR1) is associated with 3 data cleaning operators: the first one that validates (DCO1) and the other two that repair the data (DCO2, DCO3). In associations in which DQR1 is associated with DCO1 and DCO2, it was assessed as positive (i.e., high-score association) by users, but the association with DCO3 was assessed as negative (i.e., low-score association). Associations between operators are also shown; the association between DCO1 and DCO2 was assessed as positive (i.e., high-score association).

The association between operators can also be assisted by recommendation systems using artificial intelligence since the operators' meta-data, evaluations, and associations can serve as a dataset for model generation. Thus, users can receive recommendations on the relevant operators related to their application.

Regarding the actors who interact with the system, three profiles were identified: user, contributor, administrator, and recommendation system. The first consumes data requirements and operators; the second provides and manages the data cleaning elements (i.e., data requirement and operators); the third manages users, and general settings (e.g., data types, anomaly types, and data cleaning platforms); the latter provides recommendation capabilities to the platform.

## 4.2 Development

### 4.2.1 Functional Perspective

The solution's features were organized into four modules: data requirement, data cleaning operator, recommendation system support, and administration. The first contains the functionalities for dealing with data requirements; the second includes functionalities related to data cleaning operators; the third contains functionalities for supporting the recommendation of operators; finally, the administration module contains functionalities for general settings of the platform. Figure 4.6 illustrates the organization of the functional modules of the solution based on the notation of *UML Use Cases Diagram*<sup>1</sup>. These modules are explained in further detail in this section.

Regarding the data requirement module, as shown in Figure 4.7, users can search, view the details, evaluate, and post comments regarding the data requirement. Only contributors can manage (i.e., create, update, and remove) a data requirement in the platform. Users can also associate a requirement with data cleaning operators, and this association can be assessed as valid or not by others.

The data cleaning operator module, as shown in Figure 4.8, contains similar functionalities from those in the data requirement module (e.g., create, search, remove, evaluate, and post comments about an operator). The user can also obtain operators for applying on his/her data cleaning

---

<sup>1</sup><https://www.omg.org/spec/UML/>

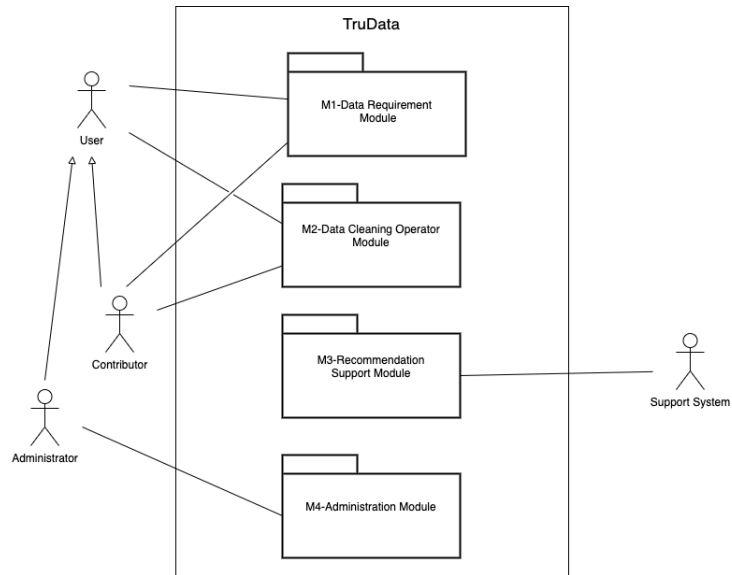


Figure 4.6: Modules of the solution

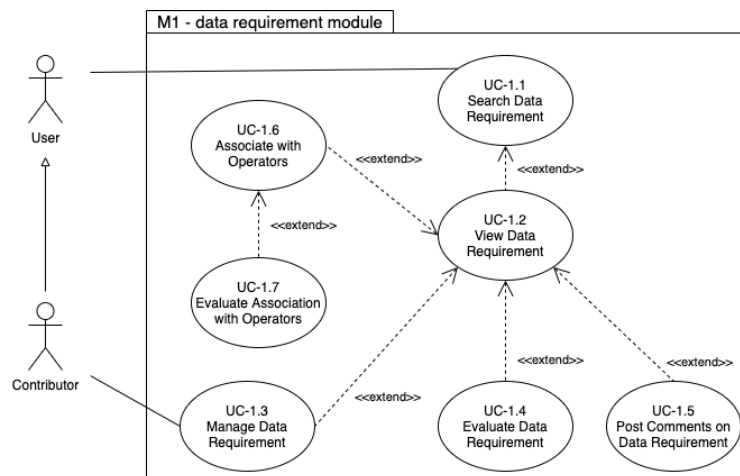


Figure 4.7: Functionalities of the data requirement module

tasks. It is also possible to associate the operator with data requirements and other operators that the user considers related; other users can collaborate to evaluate this association.

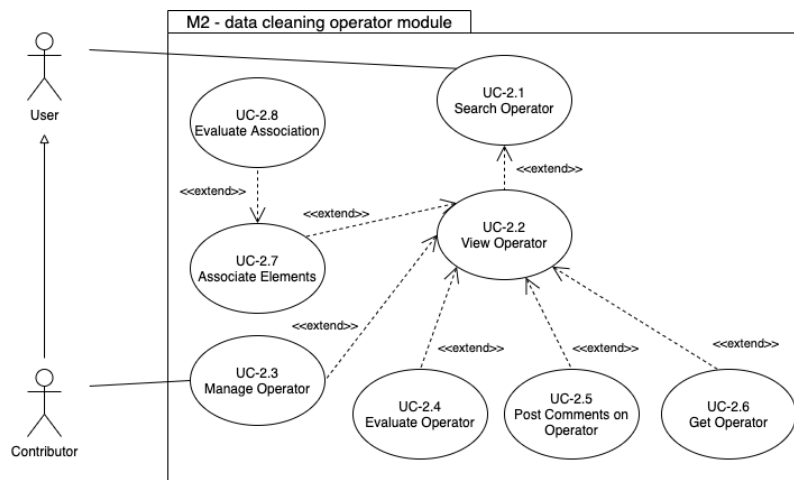


Figure 4.8: Functionalities of the data cleaning operator module

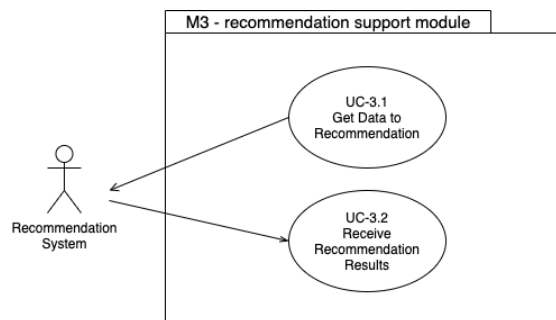


Figure 4.9: Functionalities of the recommendation support module

The recommendation support module, as shown in Figure 4.9, contains functionalities for generating recommendations of related operators to the user. A recommendation system interacts with the solution to obtain the data for applying recommendation algorithms and then send the results back to the platform.

In the administration module, as shown in Figure 4.10, the administrators manage the primary data that support the cataloging of the elements (i.e., data requirements and data cleaning operators): type of operator, data types, data domains, tags, and anomaly types. They can also manage the users and obtain access history regarding these elements.

## 4.2.2 Data Perspective

In this section, domain data modeling is presented using the UML notation<sup>2</sup>. The *package diagram* was used to show an overview of the elements involved. *Class diagrams* are designed to describe the most significant parts of this modeling.

<sup>2</sup><https://www.omg.org/spec/UML/>

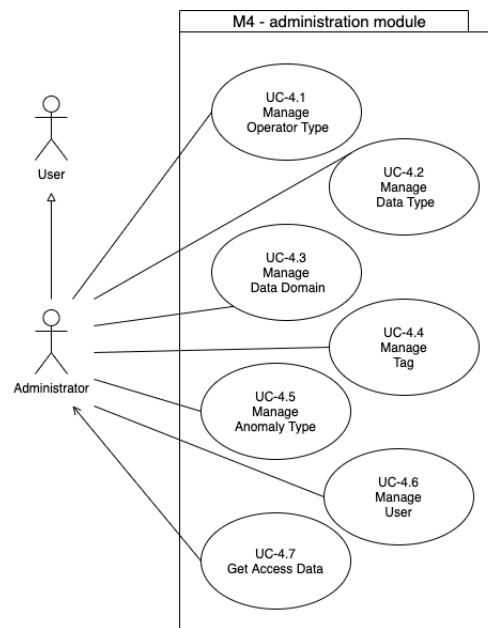


Figure 4.10: Functionalities of the administration module

The data domain classes were organized into three packages: catalog, social and data analysis, as shown in Figure 4.11. The first contains the elements used for cataloging operators and data requirements, such as data type, anomaly, and platform; the second package contains the social elements used in the platform, such as user, association, evaluation, and comment; the third contains elements for access history and recommendation system. The catalog and data analysis packages depend on the social package due to the social approach of the solution, which is explained later in this section.

#### 4.2.2.1 Catalog Package

In the catalog package, as shown in Figure 4.12, the classes *Operator* and *DataRequirement* represent the main concepts of cataloging. They can be associated with the classes: *DataType*, *Domain*, *Tag*, *OperatorType*, *Platform*. These classes have the following attributes in common: name (i.e.,

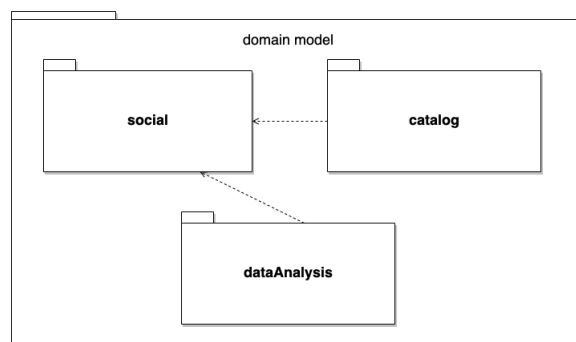


Figure 4.11: Package diagram of the data domain modelling

textual identifier) and description (explanation of the element in more detail); for the sake of simplicity, these attributes have been omitted from the diagram.

The class *Operator* (Figure 4.12) has the attributes: name (textual identifier), description (explanation of the operator in more detail), *sourceCodeURL* (URL of the operator's source code), and example (URL of the example of use). This class has associations with other classes for representing the operator according to the definition in this solution: *DataType*: an operator has only one data type (e.g., Number); *Domain*: an operator has only one data domain (e.g., Healthcare); *Tag*: an operator can have one or more tags (e.g., Symptom, Fever); *OperatorType*: an operator has only one type (e.g., Validation); *Platform*: an operator has only one platform (e.g., Python)

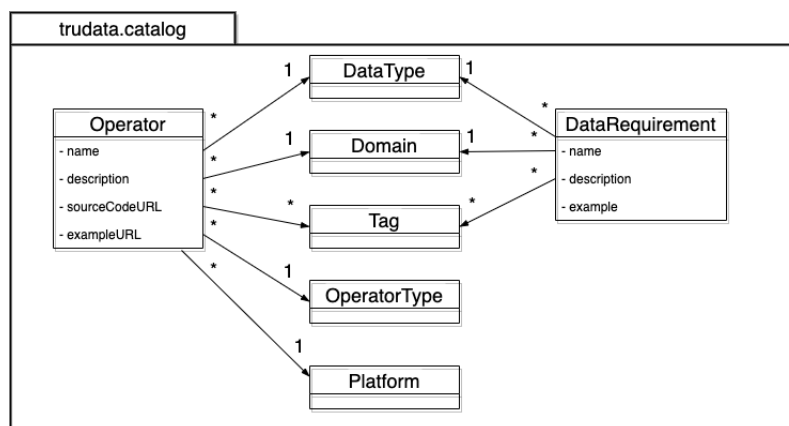


Figure 4.12: Simplified class diagram of the catalog package

The class *DataRequirement* (Figure 4.12) has the attributes: name, description, and example; being similar to those of the operator class; however, the example attribute, in this class, contain the content and not the reference, as in the operator class. This class has associations with other classes for complementing the data requirement meaning in this solution: *DataType*: a requirement has only one data type (e.g., Number); *Domain*: a requirement has only one data domain (e.g., Healthcare); *Tag*: a requirement can have one or more tags (e.g., Symptom, Fever); It can be noticed that these classes also have associations with the class *Operator*, which will be helpful in associations between operators and requirements managed by users and also supported through recommendation tools.

#### 4.2.2.2 Social Package

The package *Social* (Figure 4.13) contains the elements to represent the collaborative aspects of the solution, such as evaluation, comments, and associations between the data cleaning elements (e.g., operators and data requirements), which are carried out by users in different roles on the platform.

The class *User* (Figure 4.13) has the attributes: name, email, password, and active (i.e., represents if the user is active or not in the platform). This class has an association with the class *Role*, which represents the distinct roles of a user (e.g., administrator, contributor, and consumer).



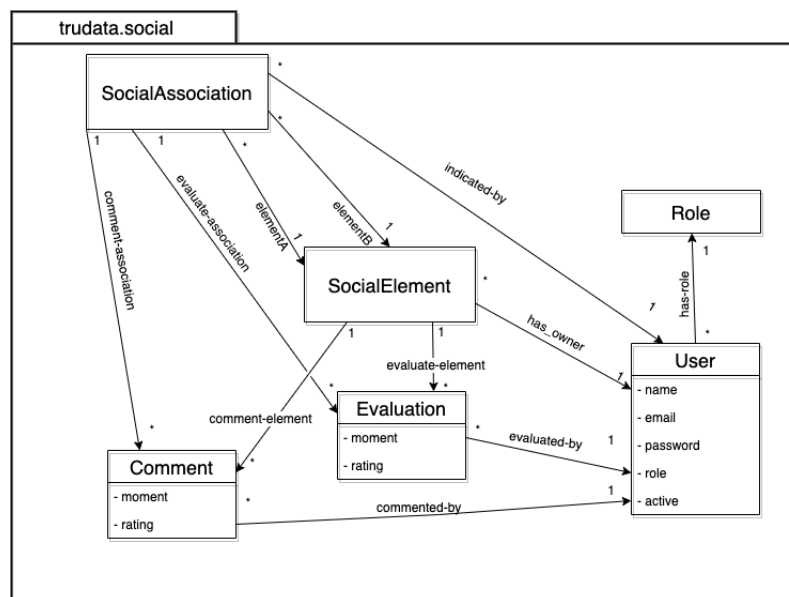


Figure 4.13: Simplified class diagram of the social package

The class *SocialElement*, in Figure 4.13, represents the social aspects of the data cleaning elements (i.e., operators and data requirements) that will be provided to them through inheritance associations. This class is associated with the class *User*, representing the element’s owner, and also with the classes *Evaluation* and *Comment* that express many evaluations and comments a social element can have.

As shown in Figure 4.13, the class *SocialAssociation* represents the association between social elements (e.g., data requirement, operator) defined by a contributor. He/she associates, according to his/her perception, an operator with a data requirement and also with others operators. Once the association is defined, other users can evaluate and post comments about it.

#### 4.2.2.3 Relation between Catalog and Social Packages

Defining the packages *Catalog* and *Social* at first has prepared the elements for internal reuse, avoiding the definition of other similar classes, which facilitates the development and the maintenance of the product.

As shown in Figure 4.14, the classes *DataRequirement* and *Operator* have an inheritance association with the class *SocialElement*, which represents the base class for social aspects. As a result, the first and second classes inherit the attributes and associations from the third one, acquiring features of evaluations, comments, and associations between them.

#### 4.2.2.4 Data Analysis Package

The data analysis package contains classes to represent the access history regarding the data cleaning elements (i.e., data requirement, operator), as well as the recommendation support provided by external tools.

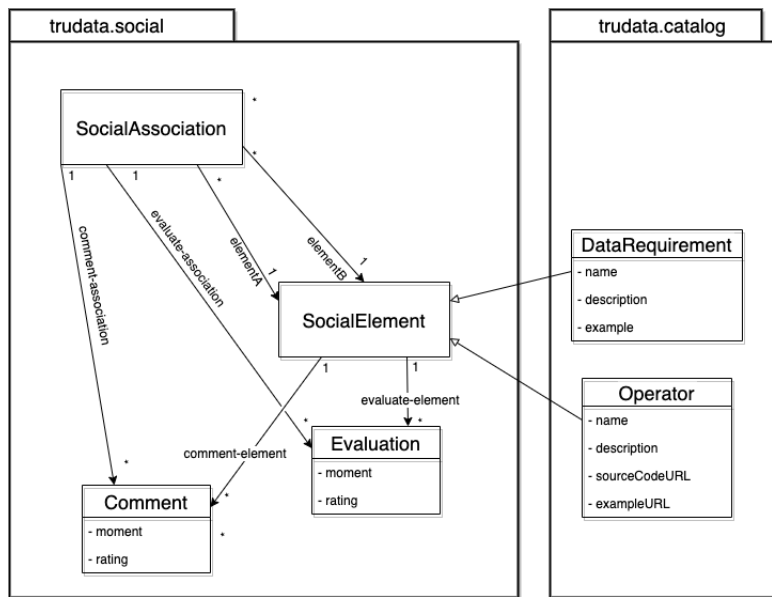


Figure 4.14: Simplified class diagram of the catalog and social packages

As shown in Figure 4.15, the class *Access* represents the user's access data to operators and data requirements. This class contains the attributes: *moment* (when the user access the element) and *type* (access type: *view* or *get* the element). These data support the generation of statistics and data analysis.

The class *Recommendation* (Figure 4.15) represents the recommendations on the elements made by an external recommender system, represented by the class *Recommender*. The former has associations with the class *SocialElement*, which is the base class of the data requirement and operator subclasses; in one association, an instance of the class *SocialElement* represents the source item (i.e., which the recommendation is for); in the other association, an instance represent the results (recommended item); thus, one source element can have many recommended (or result) elements according to the type of use (*view*, *get*, *associate*).

### 4.2.3 Deployment Perspective

This section presents the deployment view of the solution, describing the elements involved and the relationships between them.

The solution operates on the web platform, involving elements of the client-side, the platform, and the source code repositories. As a way to deploy the solution's artifacts, it was proposed to use cloud-computing services and container technology, nonetheless, it can also run on-premises. No specific technologies have been defined to serve the artifacts due to the fact the solution can be implemented using more than one existing technology for each type of artifact, leaving the implementer of the solution to decide the most appropriate ones according to the context. The Figure 4.16 shows the organization of these elements.

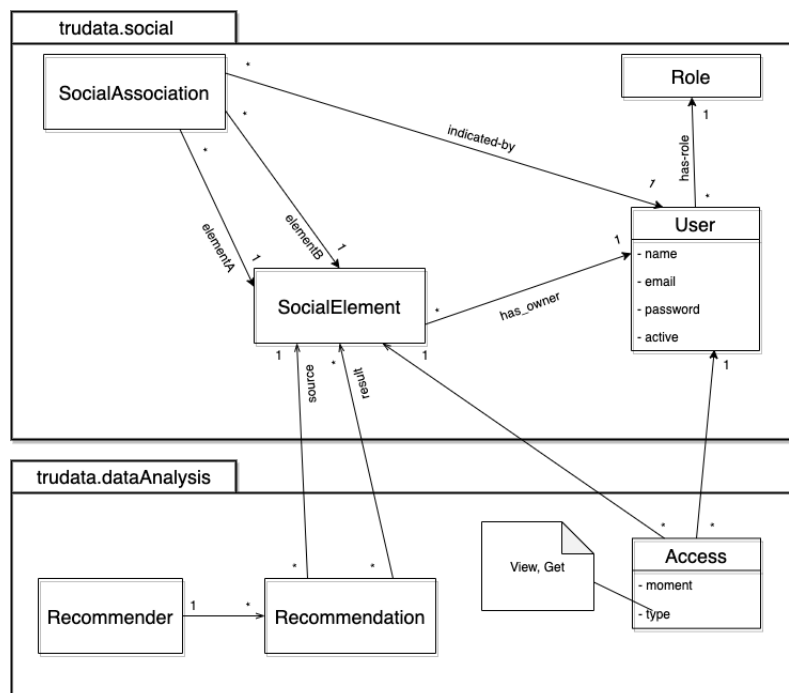


Figure 4.15: Simplified class diagram of the social and data analysis packages

In the Figure 4.16, the central element (*Container Host*) represents the execution node of the TruData platform and the front-end client application. Both artifacts are served by means of containers, with a specific image for each of them, containing the basic technology necessary to serve. The TruData web service artifact is served using images of containers with backend technologies (e.g., Java, NodeJS, DotNet). The web application artifact, which consumes the TruData web service, is served in a container image containing frontend technologies (e.g., Angular, React). The TruData web service depends on the TruData database element located on the database node.

The node Database hosts the database schema of the platform. Relational Database Management System (RDBMS) was considered due to the associations between the data elements according to the solution design. Regarding the association between the Container Host and Database nodes, only one instance of each node is related to each other.

One instance of the backend node can relate to multiple instances of the source code repository node, since an operator's source code is hosted in the contributors' repositories. The solution considers the use of GIT repositories or websites (web location). GIT repositories support version control and distributed work, which is important for collaboration among users in developing data cleaning operators.

The node of the user client machine hosts a web browser to access the client application (TruData Web App). Several instances of client machines can access the frontend node, where the web application is hosted. The solution also interacts with other applications, such as recommendation systems.

Many recommendation nodes can access the platform on the backend node via the web service API. The recommendation service can be hosted on a cloud computing platform and can be

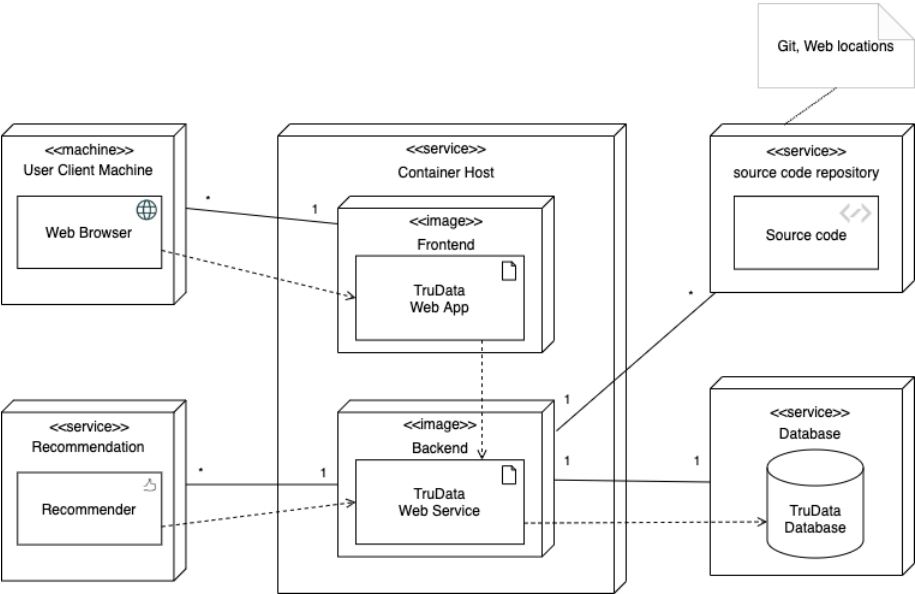


Figure 4.16: Deployment diagram of the TruData

implemented in technologies commonly used for this purpose (e.g., R Language, Python).

# Chapter 5

## Proof of Concept

This section presents the implementation and validation, describing the methodology and the results obtained. The following sections describe the decisions about the scope, components, and technologies used, in addition to the evaluation method. In the end, the results are presented and discussed.

### 5.1 Methodology

The scope of the TruData solution model includes features related to the following contexts: data requirements, data cleaning operators, data analysis, and platform administration. This Proof of Concept (PoC) implements and validates part of the data cleaning operator module in order to analyze the experience of obtaining and applying an operator.

Regarding the usage scenario, it is expected that the users find the data cleaning operators on the TruData platform and apply them to data cleaning tasks. This scenario is considered essential to assess whether the proposed reuse of this type of element is feasible. Hence, the following features were considered to be implemented based on the TruData solution model:

- Search for data cleaning operators;
- View details on a selected data cleaning operator;
- Get the source code of a data cleaning operator.

The implementation considers the back-end (i.e., web service platform) and front-end (client application) components, using current technologies in the industry. Later, the software artifacts were deployed in cloud-computing services.

To evaluate the implemented solution, usability tests were conducted based on a comparison of reusing a data cleaning operator provided by TruData with other approaches chosen by the participants, measuring effectiveness, efficiency, and satisfaction, in addition to obtaining important perceptions from the target audience.

Figure 5.1 provides an overview of the methodology of this proof of concept, which is described in more detail in the following sections.

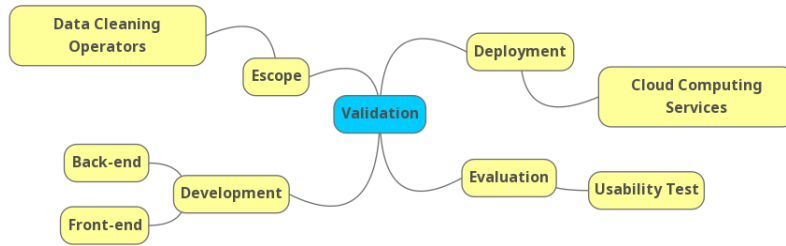


Figure 5.1: Overview of the Validation Methodology

## 5.2 Implementation

This section presents the functionalities implemented for validation, describing the user interfaces, the deployment components, and also the chosen technologies.

In order to demonstrate the user experience considering the features mentioned above, others were also implemented at the backend level (i.e., web service) to enable the experimental setup of the usability tests, such as the posting of data cleaning operators, the association of these operators, in addition to visualize comments and ratings, and other basic features.

### 5.2.1 Functionalities

According to the scope defined for this PoC, the functionalities were implemented on the web platform, aiming to be easy and intuitive to use for the target audience.

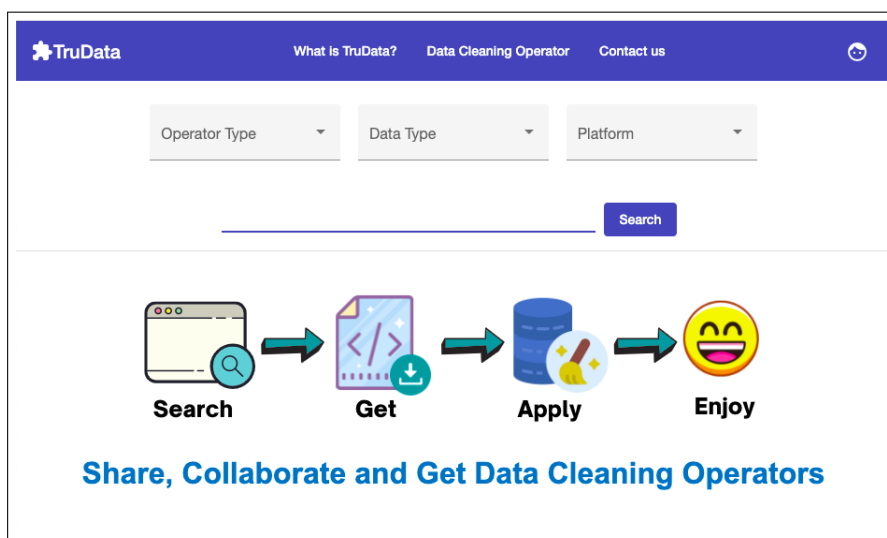


Figure 5.2: Functionality - Search for Operators

### 5.2.1.1 Search for Data Cleaning Operators

This functionality, as shown in Figure 5.2, allows the user to search for operators based on criteria such as operator type (e.g., validation, repairing), data type (e.g., text, number, date), and implementation platform (R Language, Python). The user can combine specific search criteria, for instance, an operator for repairing values on a textual data field that was developed in the Python platform, and can be applied to the health data domain.

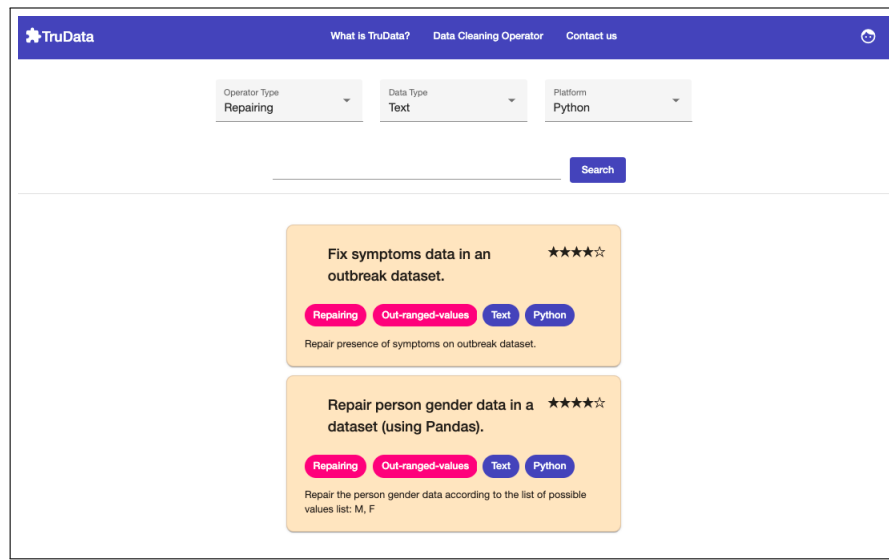


Figure 5.3: Functionality - Results of the Search for Operators

As shown in Figure 5.3, the platform responds by showing a list of operators that meet the criteria, containing other information, such as data domain, data anomaly type and average user ratings. So, the user can choose which operator is most applicable for their usage scenario.

### 5.2.1.2 View Details on a Selected Data Cleaning Operator

In this functionality, as shown in the Figure 5.4, the user sees details about the selected operator, such as classification elements (e.g., data type, data anomalies, data domain), description, usage example, and the source code of the data cleaning operator. The source code can be downloaded or copied to the clipboard, as well as the operator's usage example. This functionality (Figure 5.5) also shows related operators and social features, such as user ratings and comments.

### 5.2.1.3 Get the Source Code of a Data Cleaning Operator

This functionality allows the users to get the source code of a data cleaning operator from the repository and then apply it to their data cleaning tasks. As it was already mentioned, the source code can be obtained by downloading a file or applying the copy-and-paste method, as shown in Figure 5.4; moreover, it is also possible to access the source code directly on the git repository (Figure 5.6), where it is originally stored.

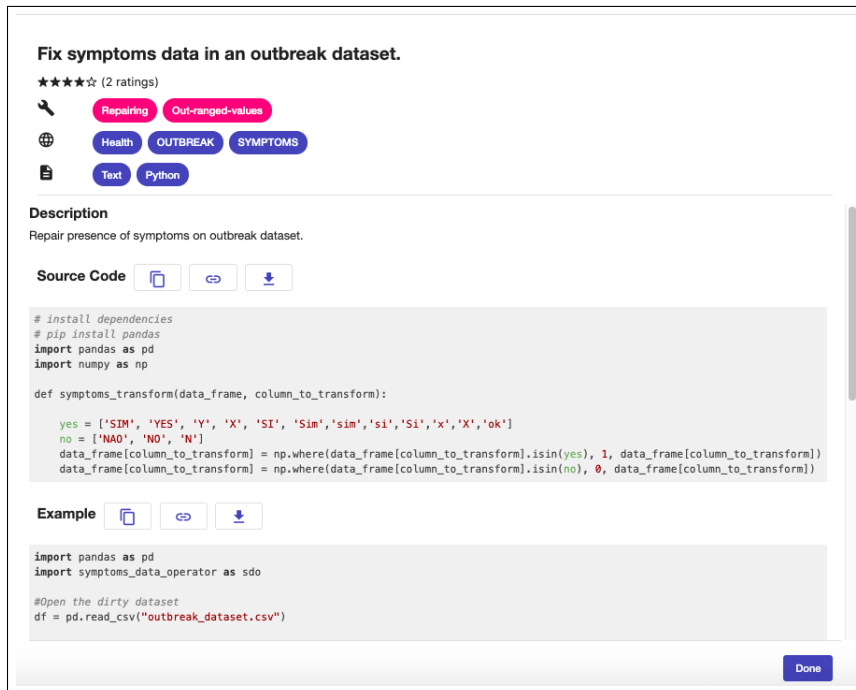


Figure 5.4: Functionality - View Details of an Operator

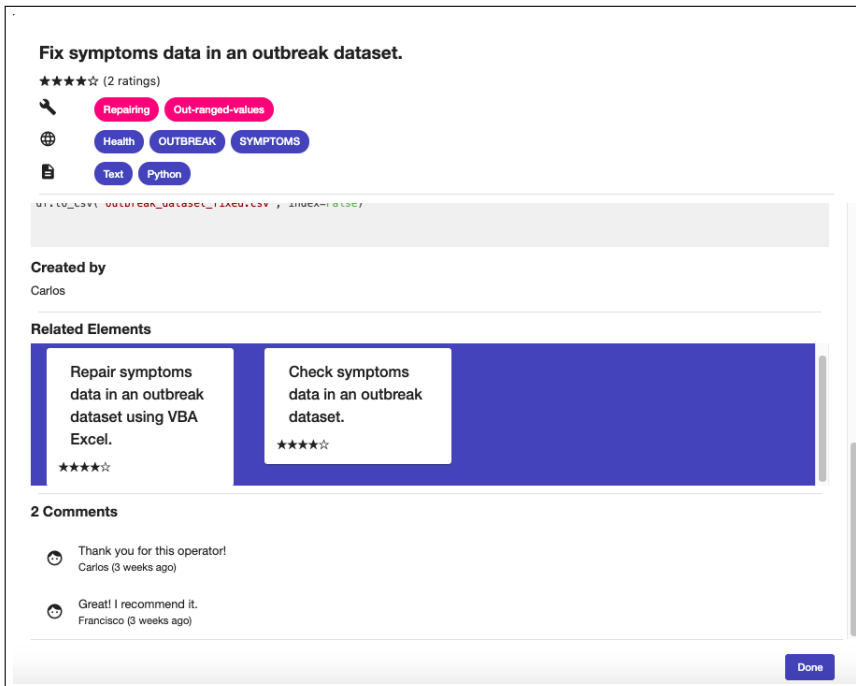
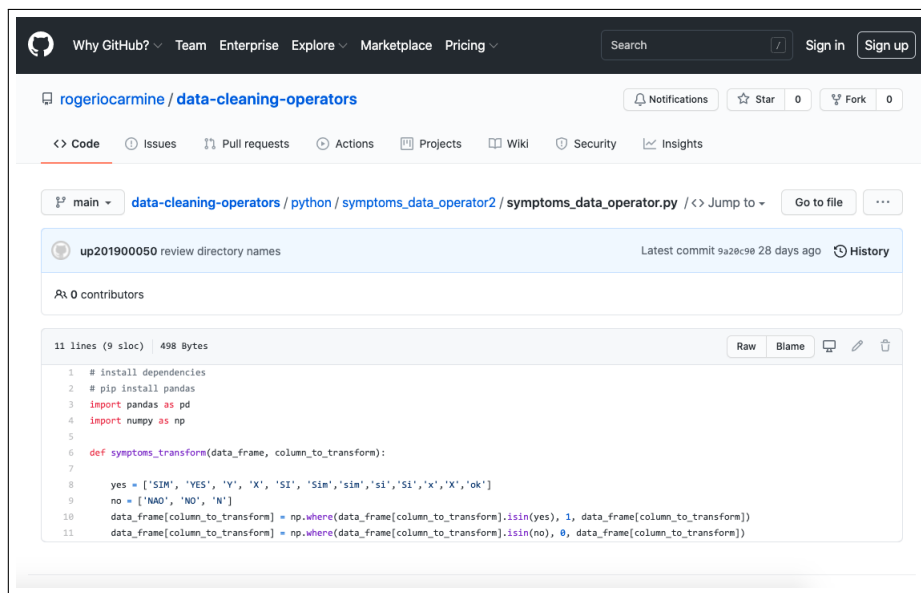


Figure 5.5: Functionality - View Details of an Operator - Related Operators and User comments

### 5.2.2 Deployment and Technologies

This section presents the deployment view of the software artifacts developed to validate the Tru-Data solution model, mentioning the technologies applied in the construction.





```

1 # install dependencies
2 # pip install pandas
3 import pandas as pd
4 import numpy as np
5
6 def symptoms_transform(data_frame, column_to_transform):
7
8     yes = ['SIM', 'YES', 'Y', 'X', 'SI', 'Sin', 'sin', 'si', 'S', 's', 'X', 'ok']
9     no = ['NAO', 'NO', 'N']
10    data_frame[column_to_transform] = np.where(data_frame[column_to_transform].isin(yes), 1, data_frame[column_to_transform])
11    data_frame[column_to_transform] = np.where(data_frame[column_to_transform].isin(no), 0, data_frame[column_to_transform])

```

Figure 5.6: Source code of a Data cleaning Operator in a GIT repository

In order to implement the central node (Container Host), the Docker<sup>1</sup> container technology was applied to build two container images: one for the backend component and another one for the frontend component. Regarding the first component, the Java<sup>2</sup> and the Spring technologies<sup>3</sup> were used to develop a web service accessible through a REST API. Regarding the second component, the Angular framework<sup>4</sup> was used to create the web application that consumes the web service provided in the backend component.

Regarding the application of the Spring technology in the backend component, the following packages were used:

- Spring Boot: This package allows the use of web servers to serve in the applications without having to install or configure other additional services [40];
- Spring Framework: This package allows the use of essential features of Sprint technology that facilitate the development of applications, such as those related to data access, component dependencies, and architectural patterns [42];
- Spring Data: This package contains features that provide additional data management elements in the application, such as JPA (Java Persistence API) and Hibernate. Regarding integration of the application and relational database systems, these two elements together reduce the effort of mapping common data entities present in both [41].

<sup>1</sup><https://www.docker.com/>

<sup>2</sup><https://www.java.com/>

<sup>3</sup><https://spring.io/>

<sup>4</sup><https://angular.io/>

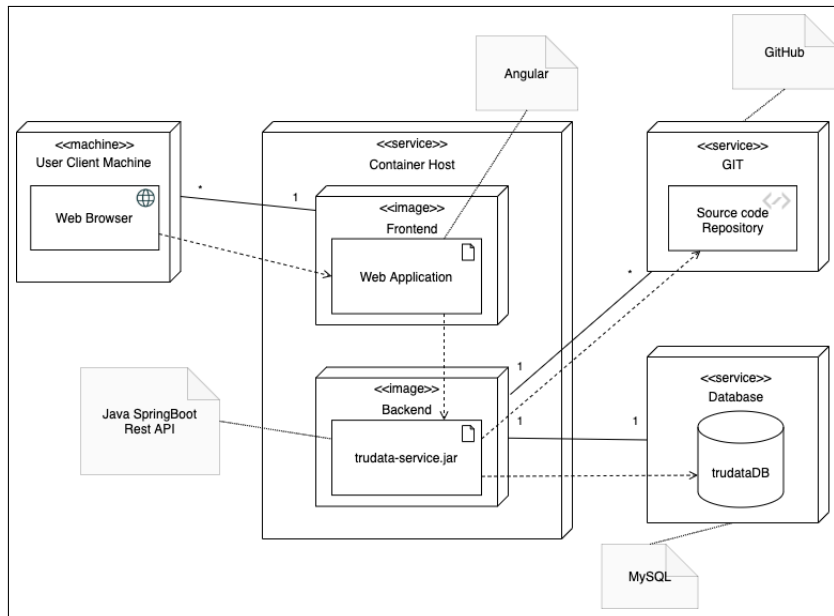


Figure 5.7: Deployment view of the implementation

The frontend component was implemented using Angular technology, which allows reusing many elements that reduce the effort in building web applications, such as visual components, application state management, API consumption. According to the Angular website [3], the platform facilitates the creation of single-page applications and includes a component-based framework, a collection of well-integrated component libraries, and tools to support the application lifecycle.

Regarding the database component, MySQL technology was used to build a relational database to store the data managed by the platform's web service. According to the MySQL documentation [31], it is an open-source relational database management system that is fast, reliable, scalable, and easy to use.

Regarding the source code repository, it was created a repository in GitHub, which is a repository service provider based on Git technology. According to Spinellis [39], Git is a distributed version control system that allows local and remote management, which is commonly available on software development platforms. Although the TruData solution model also considers other types of repositories, GIT repositories were chosen for the implementation of TruData, because it includes version control features and support distributed work, which is aligned with the proposal of the solution model.

### 5.3 Validation

This section describes the process of validating the implemented artifacts with the target audience, describing the testing technique and test methodology.

### 5.3.1 Methodology

This work intends to assess whether the proposed solution reduces effort and time and increases user satisfaction when performing data cleaning tasks. According to Wichansky [50], the usability testing technique is used to assess user performance and product acceptance. Therefore, this technique was considered to evaluate whether effort, time, and satisfaction factors are affected when TruData is applied in comparison to other methods.

The objective of the test is to perform data cleaning tasks on a given dataset to assess the effort, time, and satisfaction factors when applying TruData, and get feedback from participants.

Seven people were invited to participate in the tests. These people work in organizations related to education and research, and carry out data preparation activities in their daily routine concerning the various applications (e.g., data science, analytical reporting, and BI). The participants were arranged into two groups according to their previous experience: Group A - Python Users and Group B - Excel Users. Participants dealt with data cleaning scripts developed in the technology according to their group.

The dataset provided is a simplified list of fictitious case investigation records simulating a disease outbreak response process. It is a tabular data CSV file with 200 rows and 11 data fields, which contains erroneous values placed on purpose for the participants to detect and repair these values. They received detailed information about the dataset to perform the data cleaning tasks according to the test instructions. A sample of the dataset is provided in Annex B.2.

Also, as part of the experimental setup, the TruData application was previously configured with some data for the test scenario: users, data cleaning operators, comments, and ratings, in addition to other basic settings; some of these operators were used by the participants during the tests.

Participants performed three tasks: the first was a task without using the proposed solution (i.e., the participants must use other methods); the second was a task applying TruData with user instructions; the third was a task applying the proposed solution without instructions. During the tests, the completion time was measured, as well as whether the participant fully completed the task, in addition to the number of errors and assists provided by the facilitator. An error is when the participant performs an action that does not contribute to the goal of the task, and assistance is when the facilitator intervenes to help the participant accomplish the task.

The participants received a document with the detailed information to perform the test, which is available in Annex B.4, along with the consent form, in Annex B.5. They also filled out a questionnaire containing questions about the user experience, focusing on the factors of effort and satisfaction according to participant perception for each task. In addition to these questions, the questionnaire contained others on sociodemographic profile and on the ease and applicability of the solution. The majority of questions are based on the 5-point Likert scale, and some are open questions. This questionnaire is available in Annex B.3.

## 5.4 Results

This section presents the test results, describing the participants, the tasks according to the evaluation criteria, in addition to other defined aspects and the feedbacks obtained.

It is important to analyze the results carefully, considering that the number of samples is small and not random, and the initial data (e.g., data cleaning operators, evaluations, and comments) has been loaded in the experimental setup.

### 5.4.1 Participants

Seven participants performed the test, four (57.1%) from group A (Python) and three (42.9%) from group B (Excel). The average age is 31.7 years old; the minimum age was 25, and the maximum was 45 years old. Regarding gender, five (71.4%) were men, and two (28.6%) were women.

It was also asked about IT skills and in what contexts they perform data preparation tasks. Regarding IT skills, three participants (43.1%) are software developers, and four participants (57.1%) are advanced users; the majority (71.4%) have more than two years of experience in data preparation, and the others (28.6%) have between 1 and 2 years. Regarding the contexts in which they perform data preparation tasks (a participant could indicate more than one), the answers were diversified and are presented in Table 5.1. The data in detail is available in Annex B.6.

Context	Qty	Percentage
Business Intelligence (BI)	2	28.60%
Big Data	2	28.60%
Data Science	4	57.10%
Software Testing	3	42.90%
Deployment of Information System	4	57.10%
Results Validation	1	14.30%
Management Report	1	14.30%

Table 5.1: Contexts that the participants applied data preparation

### 5.4.2 Tasks

Statistical analysis was performed, considering the average, standard deviation, minimum value, and maximum value per task; these values in detail are available in Annex B.7. Detailed data on participants and tasks are available in Annex B.6.

All the participants completed all proposed tasks. In the first task, participants were free to choose the method for cleaning the data; the majority used automated functions based on mapping tables; the other party performed manually by selecting the erroneous values and then repairing them. Next, the results will be presented according to the metrics used: completion time, effort level, satisfaction level, numbers of errors and assists, among others.

Regarding the completion time, as shown in Figure 5.8, task 1 (without using the Trudata) had the highest average (9.33 minutes), followed by the tasks in which the participants used the

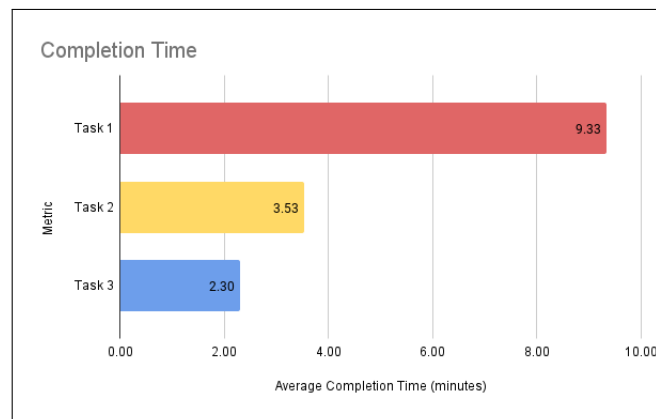


Figure 5.8: Tasks comparison based on completion time

proposed solution, task 2 (3.53 minutes) and task 3 (2.30 minutes). It is observed that the tasks performed using TruData had the least time compared to the task in which different methods were applied.

Regarding effort level, the results are shown in Figure 5.9, in which score one means the minimum effort and score five means the maximum effort; task 1 had the highest average (2.57 points), followed by task 2 (1.14 points) and task 3 (1.00 point). It is observed that the tasks performed using TruData had a lower level of effort compared to the task in which other methods were applied.

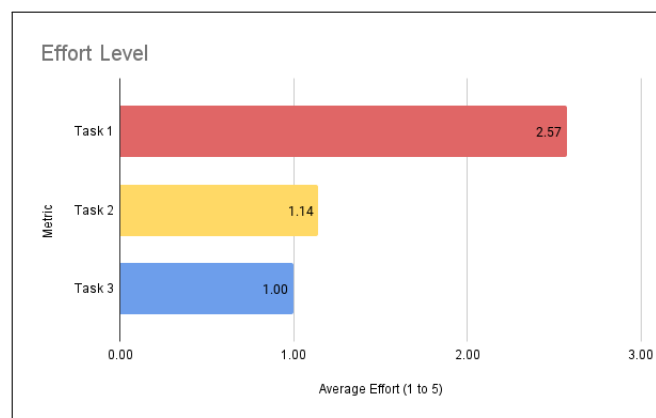


Figure 5.9: Tasks comparison based on effort level

On satisfaction level, as shown in Figure 5.10, where score one means the most unpleasant and score five means the most pleasant; task 1 had the lowest average (2.29 points), followed by task 3 (4.14 points) and task 2 (4.29 points). It is observed that the tasks performed using TruData had a higher level of satisfaction compared to the tasks in which other solution was used.

The results on error and assistance are shown in Figure 5.11. Task 1 had the highest averages (2.20 errors and 2.60 assists), followed by task 2 (1.17 errors and 1.80 assists), and task 3 (1.00 error and 1.25 assists). There is a more significant decrease in these values from task 1 (without

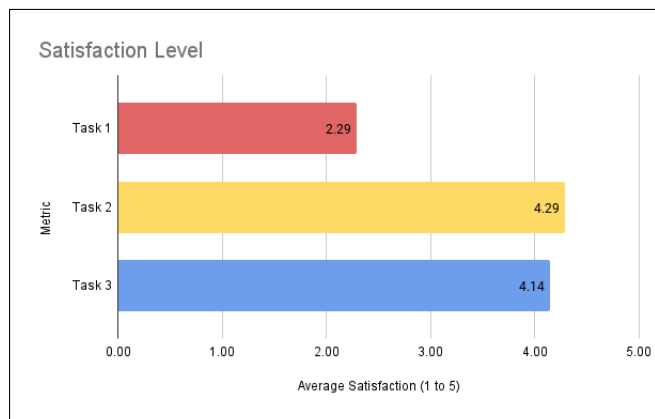


Figure 5.10: Tasks comparison based on satisfaction level

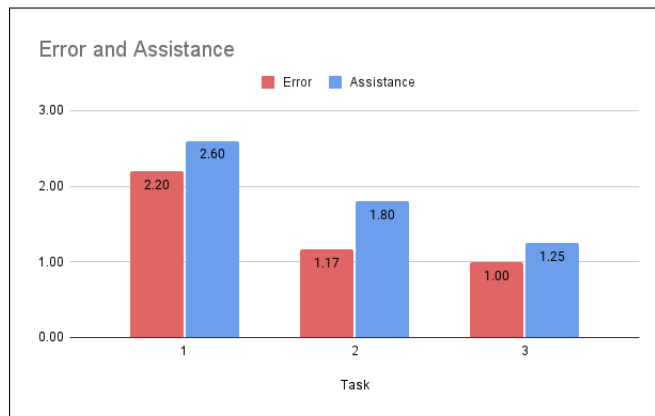


Figure 5.11: Number of errors and assistance during the tasks performance

using the TruData solution) to task 2 (using the proposed solution) and a smaller decline from task 2 to task 3 (also using the solution).

It was also asked whether the ratings and comments of other users present in the application influenced the choice of operators. The chart in Figure 5.12 presents the number of answers per score, considering that one point means no influence at all and five means extremely influential. The majority (4 participants) responded that these social elements influenced the choice, 1 participant scored indifferently, and 2 participants scored that they had no influence.

When asked about usage aspects of the TruData (i.e., frequent use, easy to use, learn rapidly) participants responded with a high level of concordance, with a concentration in 4 and 5 points, as shown in Figure 5.13. It is also noted that no participant scored less than 4 points.

The results were also analyzed considering the two groups of users: A (Python Users) and B (MS-Excel Users). There is four participants from group A, representing 57.14%, and three from group B, representing 42.86%. The results are detailed next by completion time, effort and satisfaction metrics.

Regarding the completion time per group, the results are presented in Figure 5.14, which shows results in an average completion time. In task 1, group A took longer to complete the task

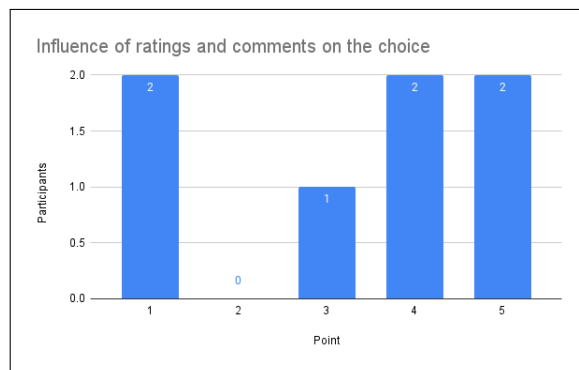


Figure 5.12: Ratings and comments influence on choosing the operator

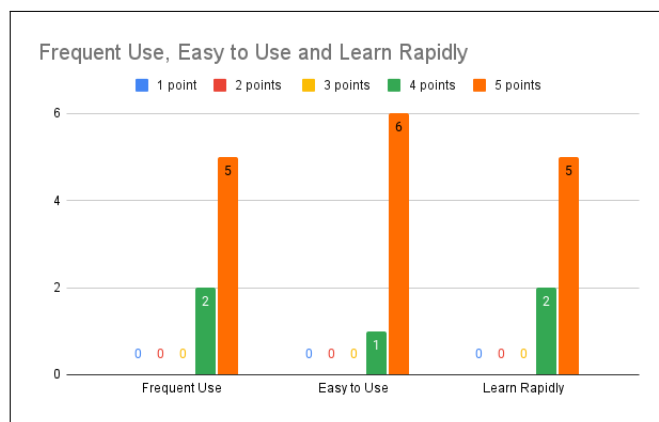


Figure 5.13: Usage Aspects of the Solution: Easy to Use, Frequent Use, Learn Rapidly

(9.96 minutes) than group B (8.48 minutes). In task 2, group A took less time (2.85 minutes) than group B (4.42 minutes). In task 3, group A also took less time (2.04 minutes) than group B (2.64 minutes) to complete the task.

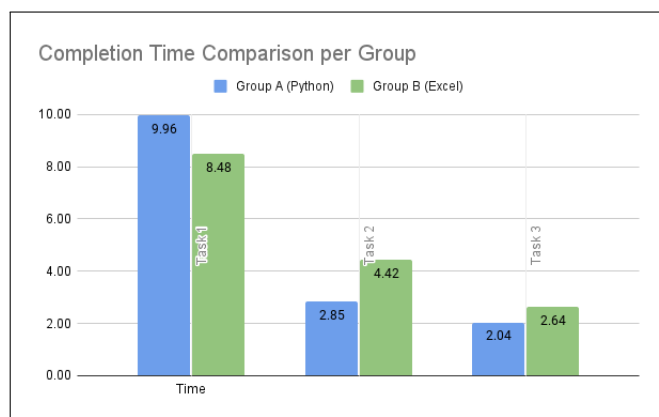


Figure 5.14: Group Comparison - Completion Time Metric

On effort level per group, the results are presented in Figure 5.15, which shows results in average of effort level. In task 1, group A had a lower average (2.25 points) compared to group

B (3.00 points). In task 2, group A also considered less effort (1.00 point) than group B (1.33 points). Finally, in task 3, both groups considered the same level of effort (1.00 point).

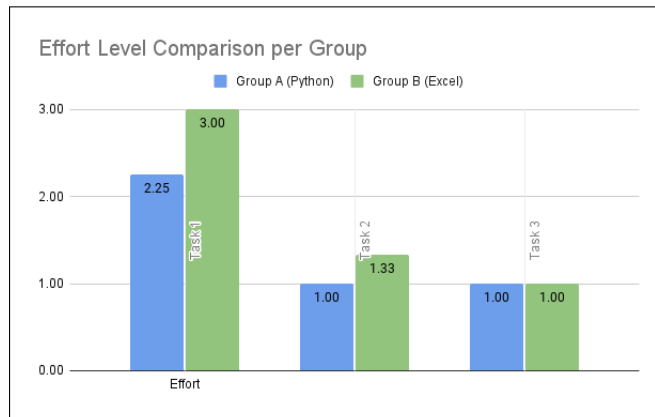


Figure 5.15: Group Comparison - Effort Metric

About satisfaction level per group, the results are presented in Figure 5.16, which shows results in average satisfaction level. In task 1, group B had a higher level of satisfaction (2.67 points) compared to group A (2.00 points). In task 2, group B also had a higher result (4.67 points) than group A (4.00 points). Finally, in task 3, group B also continues with the highest level of satisfaction (5.00 points) than group A (3.50 points).

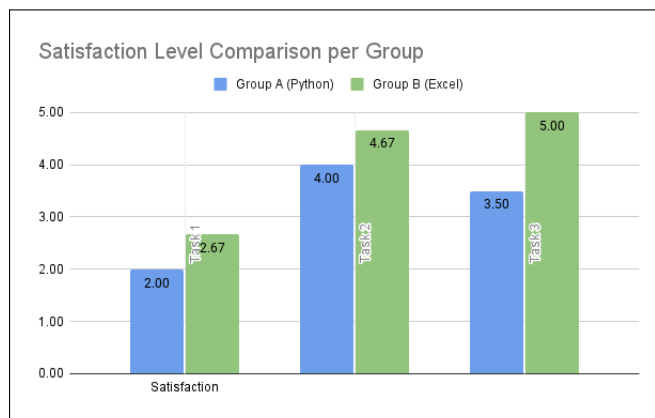


Figure 5.16: Group Comparison - Satisfaction Metric

Regarding the influence of ratings and comments from other users on the choice of a data cleaning operator, the group participants responded very similarly, as shown in Figure 5.17, with the same number of responses in points 1, 4, 5. One participant from group A responded that he/she is indifferent (point 3).

Regarding the usage aspects (i.e., frequent use, easy to use, and easy to learn) the average of the answers in each aspect was calculated. The groups had a close average, from 4.5 to 5.0 points in all these aspects, as shown in the Figure 5.18.



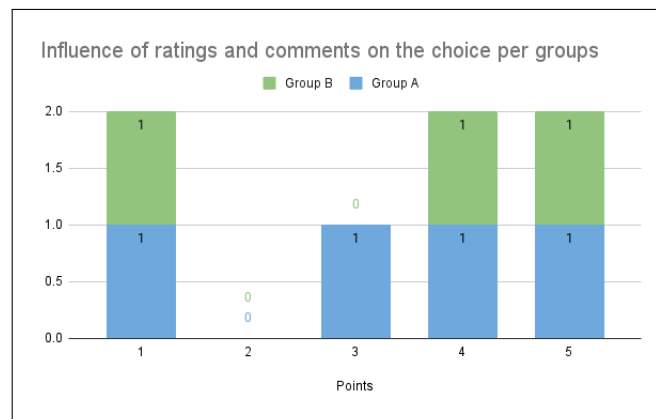


Figure 5.17: Group Comparison - Influence of ratings and comments

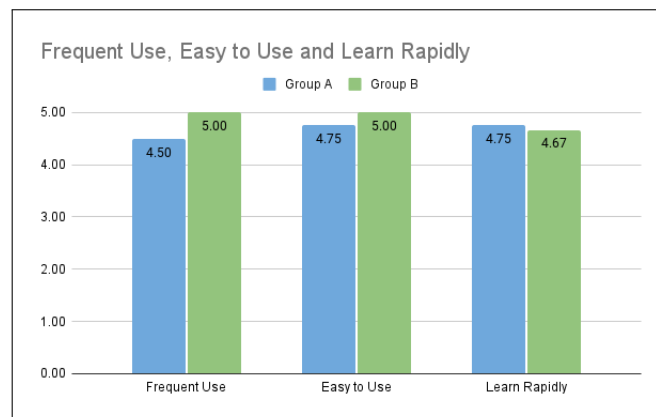


Figure 5.18: Usage Aspects per Group: Easy to Use, Frequent Use, Learn Rapidly

### 5.4.3 Comments from the Participants

Participants also commented on their experience of applying TruData, which is related to general perceptions about the application and suggestions for improvement. These comments have been compiled and described below. The original ones are available in Annex B.8.

- Most useful when the problem is more complex: The tool's benefit is more perceived to solve more complex problems. Users use tools daily to solve common and more straightforward issues. Users would use the TruData to find solutions not available in the tools they already use.
- Excel users were uncomfortable using source code to clean data: For Excel users, using source code for methods to fix data issues has made them uncomfortable. They typically use options available in Excel itself without having to deal with implementation details.
- Improve the highlight on user ratings and comments: Some users did not notice the user ratings and comments elements. They said it could be better highlighted.

- Platform-generated function package: TruData could generate a function package that could be downloaded and installed into the users' tools.
- Provide sample dataset: The platform could provide the sample dataset to test the data cleaning operator, so the user could have a common scenario to evaluate and learn to use the operators instead of using it in their datasets first.
- Execution of operators on the platform itself: Instead of downloading operators to apply in the user tool, the user could perform data cleaning tasks on the TruData platform.
- Application in other domains and other platforms: The platform could be applied in other fields, like image processing, and also in different platforms like C++;
- Operator Source-code Certification: Since dealing with source code may bring some concern about them being malicious, the operators could have some security certification.
- Operator suggestions according to search terms: The platform could suggest operators according to the terms placed in the search field.

## 5.5 Discussion

This section discusses the results presented above based on the research questions raised in Chapter 1, which are related to the aspects of reuse and collaboration to improve time, effort and satisfaction on performing data cleaning tasks. Finally, it presents a comparison between TruData and the related works present in Chapter 3.

The experiments exclusively evaluated the scenario of search and application of these operators by consumers. In the experimental setup, data cleaning operators were previously cataloged in TruData, simulating the contributions of users who experienced similar data quality problems; so, the consumers (participants) could get and apply the operator according to the task.

In task 2 e 3, reusing the operators available in the application compared to applying a method without reusing a specific operator for the case (task 1) positively impacts the results. In task 1, most participants had to prepare data structures for mapping the values (e.g., dictionary, table) that supported the error correction. In tasks 2 and 3, this element was already available inside the data cleaning operator in TruData.

It seems that the decrease in numbers of errors and assistance in tasks was influenced by reuse, especially when comparing tasks 1 and 2, which has a higher decrease. Between tasks 2 and 3, the usability elements of the implementation contributed to the decrease, as users learned how to use the application.

### 5.5.1 Answers to Research Questions

This subsection answers the research questions based on the results obtained. **Considering the application of reuse and collaboration concepts:**

**Is it possible to reduce the completion time on performing data cleaning tasks?** Yes, the results show a decrease in completion time when TruData is applied. The same happened with the analysis carried out per user groups, which remained consistent; comparing the completion time on the same task, participants from group A (Python Users) have more benefit from the solution compared to the group B (Excel Users) due to the greater time decrease (i.e., in task 1, the participants used a different method from tasks 2 and 3, when it was applied TruData).

In task 1, group B (Excel Users) had a shorter time than group A (Python Users). Group B did not apply scripts to solve the problem in this task, but only visual components and formulas in the Excel environment, which seems to facilitate this type of task in comparison to applying source code by group A; however, the difference is small, and more studies are needed. In tasks 2 and 3, group A seems to carry out activities that involve source code more easily than group B, which may have influenced the shorter time of this group.

**Is it possible to reduce the effort level on executing data cleaning tasks?** Yes, the results show a reduction when the TruData is applied. The same occurred consistently in groups. Participants from group B (Excel) perceived more of the effort of the data cleaning tasks when TruData is not applied than participants from group A (Python). When the proposed solution is used, the perception of effort is closely the same in both groups.

In task 1, some participants from group B (Excel Users) corrected errors manually, which may have increased their effort perception compared to group A (Python Users), who used programming scripts to accomplish the task. In tasks 2 and 3, the difference of effort level was minimal when groups are compared.

**Is it possible to increase the user satisfaction on performing data cleaning tasks?** Yes. The results show that TruData increases the satisfaction level in tasks where it was applied compared to the task where another method was used; the same occurs in the comparison by group. Satisfaction is also related to how the user interacts with the solution; considering that the application was evaluated as very positive in usability aspects, it can be influenced the satisfaction criterion.

It seems that Group B had slightly greater satisfaction in all tasks compared to Group A due to the friendly visual components of Excel, which facilitates and reduces the hassle when performing these types of tasks, especially regarding the faster visual feedback after correcting the values. Nonetheless, this difference is minimal, and it is necessary to conduct other studies for more accurate results.

**Is it necessary to prepare the data cleaning operations for reuse?** Yes. There is an effort to prepare a data cleaning operator for sharing with the community. For instance, the operator must have a clear interface (i.e., API), with a name stating the purpose and needed parameters, besides creating good descriptions and usage examples. In this study, the operators were previously developed and made available in the TruData application as part of the experimental setup.

It is also important to consider if the data cleaning operators are prepared for reuse in various scenarios. Contributors who develop these operators need to be aware of cohesion and coupling details, ensuring that the operator performs a clear and specific function and has all dependencies associated.

### **What user experience elements affect the reuse and collaboration on data cleaning elements?**

Some elements influenced the user experience during the data cleaning tasks: use of terms commonly adopted by the data cleaning domain, simplicity of the solution, evaluation elements (ratings and comments). The solution presents terms commonly used in the data cleaning domain (e.g., data quality anomaly, data type, data domain, and platforms), and avoids that the user needs to acquire other skills to apply the reuse and collaboration concepts (e.g., others languages or specifications). The results indicate that users consider TruData easy to use and learn, and they would use it often. In addition, TruData also decreased the number of errors and assists when applied to data cleaning tasks.

The elements of ratings and comments were essential to show that operators have already met users' needs, reducing concerns about the application of operators developed by third parties. However, some results suggest that it is necessary to invest more in this type of resource to distinguish the most suitable operators for each user application.

A concern has been identified regarding dealing with source code for data cleaning. They mention that it is necessary to understand the implementation details of the operators before applying the operators in their context. Suggestions regarding executing the operator on the platform, providing a dataset for testing, and reliability assessment of operators can reduce this concern.

## **5.5.2 TruData and Related Works**

This section relates TruData and the contributions to the data cleaning field regarding reuse and collaboration described in Chapter 3. These contributions cover specific aspects of the data cleaning field, not covering jointly the aspects of reuse and wide collaboration, which is the proposal of TruData. The proposed solution can be used along with these contributions to improving the performance of data cleaning tasks.

Regarding the reuse aspect, TruData can support the repository of specification or implementation (source code) of operators developed in the tools: Frictionless Data [14], Potter's Wheel [35] and the ontology-based solution [2]. In addition, TruData can add sharing and wide collaboration functionalities during the development of operators in these solutions.

Regarding the ontology-based solution [2], the TruData proposal differs from this solution because it uses a more straightforward approach (i.e., less formal approach) to have more simplicity in the process; that is, it is not necessary to know ontology concepts and specification languages to allow the reuse of data cleaning elements. However, the TruData approach may make the data cleaning elements more susceptible to ambiguity than the ontology-based solution. This trade-off could be the object of investigation in the future.

The tools Transform-Data-by-Example (TDE) [17] and Auto-Transform [23] focus on transforming values to solve data cleansing issues. These tools use source code or data table repositories (value pairs) to provide data transformation functions; in this case, TruData could leverage the use of these solutions, serving as input to provide access to the repositories where transformation functions (or operators) and data tables are stored.

CoClean [30] covers the collaboration aspect of performing data cleaning tasks, allowing other users to collaborate to clean the same dataset, which can raise data privacy concerns in some scenarios. By contrast, TruData is based on sharing operators, not data, as a way to allow the broader collaboration from users who experience similar data quality issues. Nevertheless, it is also possible to use TruData along with CoClean in data cleaning tasks; for example, Multiple users can get operators available in TruData and then perform data cleaning tasks in the same dataset simultaneously using CoClean.

One of the most relevant aspects of FedClean [29] solution is data privacy during the collaboration between devices, because it identifies data quality issues using collaboration between servers to detect abnormal values without exposing them. TruData, in turn, is designed to deal with data cleaning operators and other elements (e.g., data requirements) rather than the datasets to be cleaned, to avoid the risk of improper data exposure. It is noticed that the FedClean solution aims to detect abnormal values; TruData, however, can allow the use of many others types of operators and still meet data privacy requirements.



## Chapter 6

# Conclusions and Future Work

The importance of data quality for obtaining effective information has become increasingly evident, motivating the development of data cleaning solutions. However, data cleaning tasks consume a lot of time and effort, and are considered unpleasant by many professionals. This work investigates whether the application of reuse and collaboration concepts widely in carrying out data cleaning tasks can reduce time and effort, and increase satisfaction in this type of task.

This work carried out the literature review, the solution proposal, and the proof of concept. The literature review analyzed data cleaning tools available in the market and related works to identify how they addressed aspects of reuse and collaboration, to identify the research gap that guided this work. The solution proposal presented the TruData concept, considering the functional, data, and deployment perspectives. The proof of concept implemented a key part of the TruData (i.e., reuse of data cleaning operators) and then validated it through a usability test with the target audience. Finally, the results were described and discussed for the conclusion of this work.

The validation considered the scenario of reusing data cleaning operators; the user searches for an operator and then applies it to the data cleaning task. Participants repaired a simplified and fictitious dataset regarding an outbreak investigation. Before carrying out the experiments, the data cleaning operators were already available on the platform, representing contributions from users.

### 6.1 Results

This work concludes that TruData, through the reuse of data cleaning operators and collaboration between users who experience similar data quality problems, can reduce time and effort, and increase satisfaction when performing data cleaning tasks. The results indicate that tasks with another method applied took an average of 9.33 minutes, while tasks using TruData took an average of 2.91 minutes (i.e., a reduction of 68.78%). Regarding the level of effort, the results show a decrease of 58.33%, tasks with another method had 2.57 effort points, and those with TruData

had 1.07 points. Regarding satisfaction, an increase of 84.38% was observed; a task with another method had 2.29 points of satisfaction, while the task with the proposed solution had 4.21 points.

It is important to prepare data cleaning operators for reuse and catalog them properly, so the users can find the applicable operators for their cases. The solution based on simplicity and common terms in the data cleaning domain positively impacted the results.

Regarding the usability aspects, the results show that all users consider the solution easy to use and that it could be frequently applicable in their contexts. The social elements (i.e., ratings and comments) influenced to get and apply data cleaning operators, indicating if the operator is reliable; some feedbacks suggest investing in these features to improve the user experience.

As limitations of this work, it can be pointed out items regarding validation scope, usage scenarios, and sampling, which are presented below:

- Regarding the validation scope, the solution was designed, and some key components (i.e., regarding reuse of operators) were validated; in the future, we plan to develop and validate the remaining components.
- About the usage scenarios, only positive ones were considered in the tests; that is, the operator is cataloged on the platform, and it completely solves the data quality problem of the new user. It is necessary to evaluate other scenarios (positive and negative) in future work.
- Concerning the tasks performed in the tests, they are related to the same type of data anomaly (out-of-range values), which may have influenced the completion time. Test with different anomalies is necessary.
- Regarding the significance of the results, the experiments were performed with a small number of participants selected from a couple of organizations. A larger variety of participants from diverse contexts selected randomly should be considered.

The TruData solution can be applied in various contexts, such as science, education and health-care, allowing reuse and wide collaboration (i.e., through the Internet) in the development and application of operators in different platforms, such as: Python, R, OpenRefine, and Frictionless. In relation to the other solutions compared in this work, TruData can be used along with most of them to leverage the data cleaning experience.

TruData is an innovative proposal that can be considered a basis for other future initiatives to support data cleaning tasks. These initiatives can respond to the increasing need for data with quality which positively affects the information for decision-making.

## 6.2 Main Contributions

The main contributions of this work are:

- A solution model that enables the reuse of data cleaning elements and broad collaboration among professionals who experience similar data quality issues;



- A Proof of Concept (PoC) of this model focused on operator reuse, which validates the scenario of obtaining and applying data cleaning operators;
- A comparative analysis of data cleaning tools available in the market, highlighting reuse and collaboration characteristics;
- A comparative analysis of related works from academia and industry regarding reuse and collaboration in data cleaning tasks;

### **6.3 Future Work**

Future initiatives can be arranged into the following dimensions: development and evaluation of other solution components, number of operators in the catalog, improve sampling, representation of data cleaning elements, and user collaboration features.

Regarding the development and evaluation of other solution components, it can be pointed out the data requirements concept, the data cleaning elements recommendations (requirements and operators), and the usability of posting data cleaning elements by the contributors.

The number of operators can be increased, considering other data types, domains and platforms, as well as carrying out experiments with a larger number of participants, randomly selected, in order to obtain more accurate results.

Some works propose other ways to represent data cleaning elements (e.g., ontologies) to reduce ambiguity. It would be interesting to compare TruData with these solutions in the context of broader users' collaboration.

The collaboration ways in TruData can evolve. It can be developed features regarding gamification, bargaining chips, and challenges motivating users to create new solutions to existing problems of data quality.



# Appendix A

## State of the Art

### A.1 Empirical Analysis of the Commercial Tools

case ID	name	mother name	document ID	gender	pregnant	date of birth	address	neighborhood	city	district/state	date of attendance	symptoms date of onset	fever	sore throat	dry cough	tiredness	headache	difficulty breathing	loss of taste or smell	case status	outcome	closing date
1	John Smith	Mary Anne Smith	CC0000001	M	N	01-01-1960	Av. Dr. Renato Azeredo, 834	Canaiá	Sete Lagoas	Minas Gerais	02/03/2020	01/03/2020	YES	YES	Y	Y	N	Y	Y	CONF.	RECOVERED	15/03/2020
2	Mary Smith	Cleo Santanna	CC0000002	F	Y	02/31/1961	Avenida Dr. Renato Azeredo, N 834	CANAA	Sete Lagoas	Minas	04/03/2020	04/03/2020	SIM	SIM	N	Y	Y	N	Y	CONFIRMED	RECOVERED	16/03/2020
3	Pedro Silva	Mathilde Soares	CC0000003	M	N	02/29/1962	Av. Contorno, 46	Tibira	Curvelo	MG	04/03/2020	01/03/2020	Y	Y	Y	Y	N	Y	YES	DISCHARGED		20/03/2020
4	Pedro Silva	Mathilde Soares	CC0000003	M	N	02/29/1962	Av. Contorno, 46	Tibira	Curvelo	MG	04/03/2020	01/03/2020	Y	Y	Y	Y	N	Y	Y	DISCHARGED		20/03/2020
5	Joana Silva		CC0000004	Masculino	Y	01/50/1963	Av. Maceió, SN	Adrianópolis	Manaus	Amazonas	04/04/2020	01/03/2020	YES	YES	NON	YES	YES	N	Y	DISCHARGED		20/03/2020
6	João Loureiro		CC0000005	Feminino	Y	01/01/1964	Av. Maceió	ADRIANOPOLI S	MANAU	AM	04/03/2020	02/03/2020	Y	Y	NO	Y	Y	N	Y	DISCHARGED		25/03/2020
7	Maria Loureiro	Filomena Martins	CC0000006	F	N	1965-01-01	Rua Aires de Almeida, 9999	S. FRANCISCO	MNS	Amazonas	04/03/2020	02/03/2020	1	1	Y	Y	N	Y	Y	DISCHARGED		25/03/2020
8	Maria Loureiro	Filomena	CC0000006	F	N	01-01-1965	Rua Aires de Almeida, 9999A	São Francisco	Manaus	AMAZONAS	04/03/2020	02/03/2020	0	0	Y	Y	N	Y	Y	DISCHARGED		25/03/2020
9	Márcio Santos		CC0000007	Masc	N	01/01/1966	Rua Sena Madureira, SN	Jardim Gramacho	Duque de Caxias	RJ	05/03/2020	03/03/2020	YES	YES	N	Y	Y	NO	Y	CONFIRMED	RECOVERED	
10	Fernanda Santos		CC0000008	Fem	N	01/01/1967	Rua Sena Madureira	Jd. GRAMACHO	Rio de Janeiro	RJ	05/03/2020	03/03/2020	Y	Y	Y	Y	N	Y	Y	DISCHARGED		25/03/2020
11	Lucas Santos	Fernanda Santos	CC0000009	M	N	01/01/2000	Rua S Madureira, SN	GRAMACHO	Duque de Caxias	RIO DE JANEIRO	06/03/2020		N	N	YES	Y	N	N	N	CONFIRM	RECOVERED	NO
12	Joseph Goodman		CC0000010	Homem	N	01/01/1968	Rua Paulo Coelho, 1111	Bouganville	Sete Lagoas	MG	06/03/2020	04/03/2020	SIM	SIM	Y	Y	N	Y	Y	CONFIRMED	DIED	27/03/2020
13	Cristiny Silver	Anna Bert		F	N	01/01/1969	Rua José de Alencar, 1234	Bouganville	Sete Lagoas	MG	07/04/2020	05/04/2020	Y	Y	N	Y	Y	N	Y	PROBABLE		
14	Cristiny Silver	Anna B.		F	N	01/01/1969	Rua José de Alencar, 1234	Bouganville	Sete Lagoas	Minas Gerais	07/04/2020	05/04/2020	Y	Y	N	Y	Y	N	Y	PROB		
data field errors																						
duplicate records																						
functional dependance issues																						

Figure A.1: Fictional Dataset of Disease Cases Investigation for Evaluating The Commercial Tools



# Appendix B

## Proof of Concept

### B.1 Methodology Overview

### B.2 Dataset

case-id	gender	date-birth	date-attendance	date-onset	symptoms-fever	symptoms-tiredness	symptoms-headache	lab-result	case-status	outcome
1	M	01-01-1960	02-03-2020	01/03/2020	YES	N	N	positive	CONF.	RECOVERED
2	F	02/31/1961	13/04/2020	04/03/2020	SIM	N	Y	positive	CONFIRMED	RECOVERED
3	M	02/29/1962	04/03/2020	01/03/2020	Y	Y	N	positive	DISCHARGED	
4	M	02/29/1962	04/03/2020	01/03/2020	Y	Y	N	P	Discharged	
5	Masculino	01/50/1963	04/04/2020	01/03/2020	YES	YES	YES	N	DISCH.	
6	Feminino	01/01/1964	04/03/2020	02/03/2020	Y	Y	Y	POS	DISCh	
7	F	1965-01-01	04/03/2020	02/03/2020	1	Y	N	NEG	DISCHARGED	
8	F	01-01-1965	04/03/2020	02/03/2020	0	Y	N	N	DISCHARGED	
9	Masc	01/01/1966	05/03/2020	03/03/2020	YES	Y	Y	P	CONFIRMED	RECOVERED
10	Fem	01/01/1967	05/03/2020	03/03/2020	Y	Y	N	NEG	DISCHARGED	
11	Mulher	01/01/2000	06/03/2020		N	Y	N	POS	CONFIM.	RECOVERED
12	Homem	01/01/1968	06/03/2020	04/03/2020	SIM	Y	N	POS	CONFIRMED	DIED
13	F	01/01/1969	07/04/2020	05/04/2020	Y	Y	Y		Probable	
14	F	01/01/1969	07/04/2020	05/04/2020	Y	Y	Y		PROB	
15	Man	20/03/1970	08/04/2020	05/04/2020	S	Si	NO		1	
16	Woman	30/09/2001	08/04/2020	05/04/2020	S	si	SIM		1	
17	Male	21/04/1985	08/04/2020	05/04/2020	S	si	YES		1	
18	Male	24/08/1994	08/04/2020	05/04/2020	Sim	Sim	YES	p	2	2
19	Male	17/09/1990	08/04/2020	05/04/2020	SIM	sim	SIM	P	2	2
20	Female	20/03/1970	08/04/2020	06/04/2020	NAO	sim	Sim	N	3	

Figure B.1: Sample of the Dataset of Disease Cases Investigation for the Solution Validation

## B.3 Questionnaire

### TruData - Teste de Usabilidade

O objetivo deste teste é realizar tarefas de limpeza de dados em um dataset fornecido para avaliar o esforço, tempo e satisfação no uso da solução TruData.

Por favor, acesse o endereço abaixo para obter mais informações sobre a realização do teste:

[https://drive.google.com/file/d/12vo\\_MtfTNXgoY4LN2sxpK5ReFmdofL5u/view?usp=sharing](https://drive.google.com/file/d/12vo_MtfTNXgoY4LN2sxpK5ReFmdofL5u/view?usp=sharing)

Nas seções seguintes, serão solicitadas informações sobre você e também sobre a execução das tarefas descritas no documento.

Desde já, muito obrigado pela sua participação !

\* Required

Informações  
sobre você

Por favor, informe alguns dados sobre você. Os dados serão utilizados para análise do perfil de utilizador, e serão tratados de forma anônima. Neste momento, também será definido o seu número de identificação no teste e qual grupo de utilizadores que você deseja participar.

1. Seu número de identificação para o teste \*

---

2. Qual é o grupo de utilizadores? \*

*Mark only one oval.*

- Grupo A - Python  
 Grupo B - Excel

3. Qual é a sua idade? \*

---

4. Qual o seu género ? \*

*Mark only one oval.*

- Masculino  
 Feminino  
 Prefiro não dizer

5. Como relação às suas habilidade de TI, você se considera: \*

*Mark only one oval.*

- Usuário Básico  
 Usuário Avançado  
 Desenvolvedor de Software

6. Quantos anos de experiência você tem em preparação de dados? \*

*Mark only one oval.*

- Menos de 1 ano.  
 Entre 1 e 2 anos, inclusive.  
 Acima de 2 anos.

7. Você costuma realizar tarefas de preparação de dados em quais contextos: \*

*Check all that apply.*

- Business Intelligence (BI)
- Big Data
- Data Science
- Teste de software
- Implantação de Sistemas de Informação

Other:  \_\_\_\_\_

#### Tarefa 1 - Reparar erros sem o aplicativo TruData

8. A tarefa 1 foi concluída ? \*

*Mark only one oval.*

- Totalmente
- Parcialmente
- Não foi possível

9. Por favor, descreva brevemente como você fez a tarefa 1. \*

---

---

---

---

---

10. Em termos de esforço, você acha que essa tarefa demandou: \*

*Mark only one oval.*

	1	2	3	4	5	
Pouco Esforço	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Muito Esforço

11. Você achou a tarefa agradável? \*

*Mark only one oval.*

	1	2	3	4	5	
Muito desagradável	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Muito agradável

#### Tarefa 2 - Reparar erros com o aplicativo TruData (com instruções de uso)

12. A tarefa 2 foi concluída ? \*

*Mark only one oval.*

- Totalmente
- Parcialmente
- Não foi possível

13. Em termos de esforço, você acha que essa tarefa demandou: \*

Mark only one oval.

	1	2	3	4	5	
Pouco esforço	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Muito esforço

14. Você achou essa tarefa agradável? \*

Mark only one oval.

	1	2	3	4	5	
Muito desagradável	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Muito agradável

Tarefa 3 - Reparar erros com o aplicativo TruData (sem instruções de uso)

15. A tarefa 3 foi concluída ? \*

Mark only one oval.

- Totalmente
- Parcialmente
- Não foi possível

16. Em termos de esforço, você acha que essa tarefa demandou: \*

Mark only one oval.

	1	2	3	4	5	
Pouco esforço	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Muito esforço

17. Você achou essa tarefa agradável? \*

Mark only one oval.

	1	2	3	4	5	
Muito desagradável	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Muito agradável

18. As avaliações e os comentários sobre os operadores influenciaram a minha escolha? \*

Mark only one oval.

	1	2	3	4	5	
Não, totalmente.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Sim, totalmente.

Questões Gerais

Por favor, responda às questões gerais sobre o uso do aplicativo.



19. Gostaria de usar a aplicação frequentemente \*

Mark only one oval.

	1	2	3	4	5	
Discordo totalmente	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Concordo totalmente

20. Achei o sistema fácil de usar. \*

Mark only one oval.

	1	2	3	4	5	
Discordo totalmente	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Concordo totalmente

21. Penso que a maioria das pessoas que trabalham com preparação de dados consegue aprender a usar esta aplicação muito rapidamente \*

Mark only one oval.

	1	2	3	4	5	
Discordo totalmente	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Concordo totalmente

22. Por favor, coloque aqui suas opiniões gerais e recomendações sobre a aplicação TruData e também sobre a realização do teste.

---

---

---

---

---

This content is neither created nor endorsed by Google.

Google Forms

## B.4 Usability Test Description

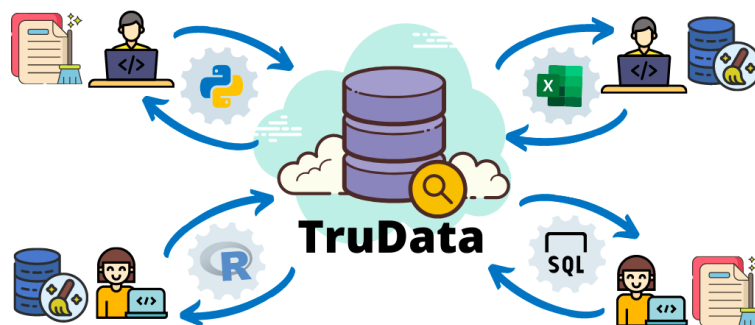
### TruData - Teste de Usabilidade

Dados de qualidade são necessários para gestão efetiva de negócios em diversas áreas (e.g., Saúde, Educação, Manufatura, Retalho e Finanças). Existem aplicações em que os dados precisam ser validados e reparados antes de serem utilizados para geração de informação (e.g., BI, Data Science, BI, Big Data).

Um cenário que tem chamado a atenção recentemente é o monitoramento de casos de COVID em cenários com falta de processos bem definidos e ferramentas adequadas para o tratamento de dados, impactando a informação para tomada de decisão em tempo oportuno. Nos contextos em que não existem sistemas de informação integrados, são utilizados arquivos de dados (e.g., folhas de cálculo, CSV) para a transmissão de dados e geração de informação, deixando o processo suscetível a falhas.

As tarefas de limpeza de dados têm um papel fundamental para melhorar a qualidade dos dados e gerar informação efetiva para tomada de decisão. No entanto, essas tarefas são consideradas por muitas pessoas como desgastantes, demoradas e desagradáveis de serem realizadas.

A solução TruData visa apoiar as tarefas de limpeza de dados através da partilha de operadores de diversas plataformas, permitindo a reutilização e também a colaboração entre pessoas no desenvolvimento desses operadores, como forma de tornar essas tarefas mais eficientes e eficazes, além de menos desgastantes.



## Sobre o Teste

O objetivo do teste é realizar tarefas de limpeza de dados em um dataset fornecido para avaliar o esforço, tempo e satisfação no uso da solução TruData.

O teste será executado com dois grupos, conforme a experiência prévia dos utilizadores: Grupo A - Plataforma Python e Grupo B - Plataforma Excel. Em algumas tarefas, você verá instruções específicas para cada plataforma, e nesse caso, somente execute as instruções da plataforma conforme o grupo que você pertence.

As tarefas devem ser realizadas na presença do facilitador para que ele possa orientar em caso de dúvidas e também fazer as medições necessárias durante o teste.

## Sobre a Plataforma TruData

TruData é uma plataforma para facilitar a reutilização de operadores de limpeza de dados desenvolvidos por outras pessoas em várias plataformas, como: Python, R, SQL, Excel, Frictionless Data, Trifacta, OpenRefine, entre outras. Um utilizador poderá encontrar um operador que levaria muito tempo e esforço de ser desenvolvido, como por exemplo, um operador para transformar valores do tipo texto conforme uma lista de valores possíveis (e.g., municípios de morada ou sintomas possíveis de um paciente), que inicialmente não foi considerada na coleta de dados

A plataforma cataloga os operadores, permitindo a classificação por tipo de operação (e.g. validação, correção ou ambas), tipo de dados (e.g., texto, número, data), domínio (e.g. Saúde, Finanças) e plataforma (e.g., Python, R, Excel, OpenRefine, Trifacta).

Para realização dos testes de usabilidade, foi desenvolvida uma versão inicial do software com o intuito de validar o conceito da solução e também obter feedbacks da audiência, com foco, inicialmente, na reutilização de operadores de limpeza de dados.

## Sobre o Dataset

O dataset fornecido é uma lista simplificada de casos clínicos (fictícios), simulando uma situação de investigação de surto de doenças. Os valores do dataset possuem vários erros, de forma proposital, para que seja possível exercitar tarefas de limpeza de dados.

Link do DataSet: [outbreak\\_dataset.csv](#)

## Informações gerais sobre o dataset

- O dataset está em formato CSV, com o separador vírgula ( "," );

- Pode conter valores textuais nos idiomas: Português, Inglês e Espanhol;
- Contém 200 linhas de registro.

### Informações sobre os campos de dados

A tabela abaixo descreve os campos do dataset, contendo o tipo de dados e outras informações relevantes para realização das tarefas. Os dados, originalmente, não estão seguindo as regras ou formato indicado. O intuito dos testes é realizar as tarefas de limpeza, deixando os dados compatíveis com as regras definidas.

Item	Campo	Descrição	Tipo de Dados	Outras informações
1	case-id	número de identificação do caso	Número Inteiro	Sequencial
2	gender	Género da pessoa	Texto	M (Masculino) ou F (Feminino)
3	date-birth	Data de Nascimento	Data	Formato: dd/mm/yyyy
4	date-attendance	Data de Atendimento médico	Data	Formato: dd/mm/yyyy
5	date-onset	Data dos primeiros sintomas	Data	Formato: dd/mm/yyyy
6	symptoms-fever	Sintoma de febre	Número	1: Com sintoma; 0: Sem sintoma.
7	symptoms-tiredness	Sintoma de cansaço	Número	1: Com sintoma; 0: Sem sintoma.
8	symptoms-headache	Sintoma de dor de cabeça	Número	1: Com sintoma; 0: Sem sintoma.
9	lab-result	Resultado do laboratório	Texto	Valores: P (Positivo); N (Negativo); I (Indeterminado).
10	case-status	Classificação do caso	Texto	Valores: Suspeito; Confirmado; Descartado.
11	outcome	Resultado final do caso	Texto	Valores: Recuperado, Óbito

## Tarefas

O participante irá realizar 3 tarefas de limpeza de dados em um dataset. A primeira será feita conforme as habilidades do participante, sem o uso da solução proposta; a segunda será feita com o uso da solução proposta, seguindo instruções de uso fornecidas; e a terceira será feita com o uso da solução, sem instruções de uso. Para cada tarefa, será verificado se foi possível concluir na totalidade, o tempo gasto, e a opinião do participante sobre o esforço e a satisfação na realização.

Solicitamos que o participante preencha o seguinte formulário para coletarmos sua percepção sobre a realização das tarefas: <https://forms.gle/waTpqYpPXipkSuFKA>

## Preparação

Antes de realizar as tarefas propostas, por favor, realize as seguintes ações para ter o ambiente preparado antes da execução das tarefas:

- a. Aceda ao aplicativo TruData no endereço: <http://157.245.79.51/>
- b. Familiarize com o aplicativo, pesquisando por operadores por: plataformas, tipos de dados, tipo de operadores, e também por uma descrição de operadores. Caso não seja fornecido nenhum argumento de pesquisa, a aplicação mostra todos os operadores registrados na plataforma.

Instruções para o uso no ambiente para plataforma Python (**Grupo A**):

- a. Crie uma pasta (sugestão: `usability_test`) para guardar o código fonte a ser desenvolvido juntamente com o dataset ;
- b. Faça o download e depois copie o dataset fornecido ([dataset.csv](#)) para a pasta criada;
- c. Abra o seu ambiente de desenvolvimento (IDE) Python (sugestão: VS Code) na pasta criada;
  - i. Se estiver optado pelo VS Code, abra também uma janela do terminal de comandos;
- d. No terminal (console), verifique se o interpretador do Python 3 está instalado e funcionando corretamente na pasta;
  - i. `pip3 --version`
- e. No seu IDE, crie um novo ficheiro, dentro da pasta, chamado **`trudata_operator_test.py`** ;
- f. Se a biblioteca Pandas não está instalada, faça a instalação através do terminal de comandos:
  - i. **`pip3 install pandas`**
- g. Importe a biblioteca Pandas;
  - i. 

```
import pandas as pd
```

- h. Realize a carga do dataset fornecido através de funções para abertura de arquivos (pandas.read\_csv):

i. `df = pd.read_csv("outbreak_dataset.csv")`

- i. Observe se possui 200 linhas e 11 variáveis no dataset, executando o comando:

i. `print(df.shape)`

- j. Execute um script simples para validar o ambiente:

i. `print("All set!")`

O código-fonte completo de preparação está disponível aqui: [setup.py](#)

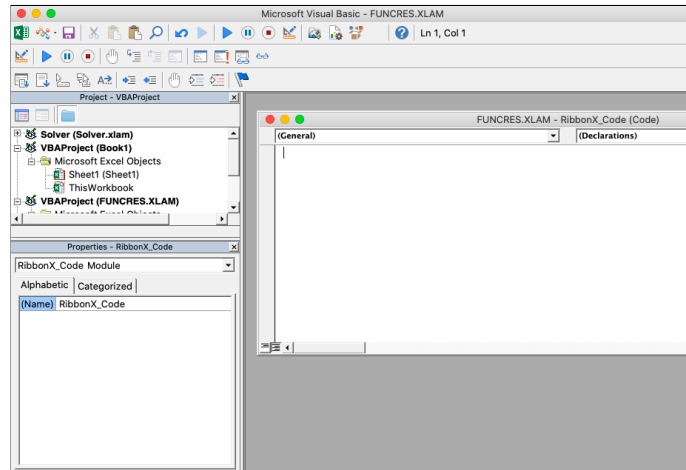
```
import pandas as pd
df = pd.read_csv("outbreak_dataset.csv")

# Check 200 lines and 11 columns
print(df.shape)

print("All set!")
```

#### Instruções para o uso no ambiente MS-Excel (**Grupo B**):

- a. Crie uma pasta para guardar os arquivos dos testes;
- b. Copie o dataset fornecido para a pasta criada;
- c. Execute o MS-Excel;
- d. Abra o dataset fornecido através da opção **Abrir** no menu Arquivo;
- e. Organize a exibição dos dados em colunas, caso ainda não esteja:
  - i. Selecione a primeira coluna;
  - ii. No menu Dados, selecione a opção "transformar texto em colunas";
  - iii. Escolha a opção referente a separado por delimitador";
  - iv. Selecione o delimitador vírgula;
  - v. Selecione a opção de concluir.
- f. Observe se as colunas estão visíveis conforme descrição fornecida;
- g. Observe se os valores estão visíveis até o total de linhas citado acima;
- h. Abra a janela de script, selecionando as opções na barra de menu:
  - i. Menu Bar -> Tools -> Macros -> Visual Basic Editor
- i. Verifique se a seguinte janela do editor de Visual Basic está aberta:



- j. Copie e cole o seguinte script, e depois execute:

```
Sub TestEditor()  
    ' teste simples do editor  
    MsgBox "Test of VB Script!"  
End Sub
```

- k. Deve aparecer uma janela de aviso, conforme a mensagem: "*Test of VB Script*".

## Execução

A partir de agora, serão realizadas as tarefas para avaliação da solução. Por favor, avise ao facilitador quando terminar a leitura do enunciado de cada tarefa para que se possa medir o tempo de realização.

Tarefa 1 - Reparar erros nos valores contidos na coluna symptoms-fever sem o uso do aplicativo TruData

- A coluna **symptoms-fever** deve conter uma lista de valores padrão: 1,0; sendo 1 para "Sim" e 0 para "Não". Sabe-se que os valores "X" e "OK" referem-se ao valor "1". Exemplos de valores corretos são : 0,1,0,1,1,1,0,0,1.
- Realize essa tarefa da forma que achar mais apropriada. Você pode: fazer manualmente, usar funções da ferramenta, pesquisar uma solução na Internet, ou implementar um script.

Tarefa 2 - Reparar erros nos valores contidos na coluna **symptoms-tiredness** com uso do aplicativo TruData.

- A coluna **symptoms-tiredness** deve conter uma lista de valores padrão: 1 e 0; sendo 1 para SIM, e 0 para NÃO. Sabe-se que os valores "X" e "OK" referem-se ao valor "Sim". Alguns exemplos de valores corretos são: 0,1,0,1,1,1,0,0,1.

### Instruções

1. No aplicativo TruData:
  - a. Acesse ao aplicativo TruData;
  - b. No campo de procura, digite: "**Symptoms**" e escolha a plataforma conforme seu grupo no filtro de plataformas (Python ou Excel), e carregue no botão de pesquisa;
  - c. Na lista de resultado do aplicativo, clique no item:
    - i. Para **Python**: "**Fix symptoms data in an outbreak dataset.**"
    - ii. Para **Excel**: "**Repair symptoms data in an outbreak dataset using VBA Excel.**";
  - d. Na página de detalhes, são mostradas mais informações sobre o operador, como o tipo, descrição e exemplos de uso. Agora, carregue no botão para fazer o download do operador e depois copie para a pasta do teste, onde estão os demais arquivos. Verifique se o nome do arquivo não mudou após o download.

2. Instruções nas ferramentas:

- a. **No ambiente do Python:**

- a. No arquivo **trudata\_operator\_test.py**, importe o operador **symptoms\_data\_operator.py** logo abaixo da biblioteca pandas:

```
import pandas as pd
import symptoms_data_operator as sdo
```

- b. Após a leitura do arquivo CSV, execute o operador, através da função para reparar os valores no dataset, e depois guarde o dataset limpo no **outbreak\_dataset\_fixed.csv**:

```
#Open the dirty dataset
df = pd.read_csv("outbreak_dataset.csv")
#Execute the data cleaning operator
sdo.symptoms_transform(df, "symptoms-tiredness")
```



```
#Save the clean dataset
df.to_csv("outbreak_dataset_fixed.csv")
```

- c. Execute o script, clicando no botão *Run*;
  - d. Abra o dataset **outbreak\_dataset\_fixed.csv**, e verifique se os valores de **symptoms-tiredness** foram corrigidos;
- b. **No aplicativo Excel:**
- i. Na janela "Visual Basic Editor", cole o código fonte obtido da plataforma;
  - ii. Verifique se é preciso mudar algo no script conforme o seu caso;
  - iii. Execute o script;
  - iv. Na planilha, se o script foi executado com sucesso, observe que as células da respectiva coluna foram reparadas.

**Tarefa 3** - Reparar erros nos valores contidos na coluna **gender** com uso do aplicativo TruData, sem instruções de uso.

- A coluna **gender** deve conter uma lista de valores padrão: M,F; sendo M para Masculino, e F para Feminino. Alguns exemplos de valores corretos são: M, F, M, M, M, M, F, F, M, F.
- Procure na aplicação TruData um operador compatível para reparar erros na coluna **gender** conforme a plataforma definida (Python ou Excel), e use no seu script que já possui a reparação de dados na coluna **symptoms-fever**;
- Se o script foi executado com sucesso, observe se os valores na coluna **gender** foram reparados.

Muito obrigado por sua participação neste teste!  
Sua opinião é muito importante para a melhoria da solução.

## B.5 Consent Form

### Termo de Consentimento

A solução TruData foi desenvolvida no âmbito do trabalho de dissertação de mestrado do estudante Rogério Luiz Araújo Carminé da *Faculdade de Engenharia da Universidade do Porto* em cooperação com a *Associação Fraunhofer Portugal Research*.

A solução atua no contexto de tarefas de preparação de dados e possui o objetivo de fornecer um catálogo de operadores de limpeza de dados desenvolvidos em diversas plataformas (e.g., Python<sup>®</sup>, MS-Excel<sup>®</sup>, OpenRefine), mantido pelos utilizadores da solução. O utilizador poderá obter facilmente operadores desenvolvidos por outras pessoas, e também encontrar pessoas interessadas na colaboração para o aperfeiçoamento ou desenvolvimento de novos operadores. Neste teste, será disponibilizado um protótipo funcional com a finalidade de validar o conceito da solução.

Os dados recolhidos durante o teste estão relacionados com a usabilidade da aplicação e protótipo apresentados assim como alguns dados sociodemográficos, que serão recolhidos através da observação da interação, gravação de som e vídeo, assim como em questionário e entrevista. Os dados serão usados para identificar oportunidades de melhorias e novos rumos para o aperfeiçoamento da solução.

Gostaríamos de contar com a sua participação, que não envolve qualquer prejuízo ou dano material e não haverá lugar a qualquer pagamento. Os dados recolhidos são confidenciais. O responsável pelo projeto tomará todas as medidas necessárias à salvaguarda e protecção dos dados recolhidos por forma a evitar que venham a ser acedidos por terceiros não autorizados.

A sua participação é voluntária, podendo em qualquer altura cessá-la sem qualquer tipo de consequência.

Agradecemos muito o seu contributo, fundamental para a realização deste trabalho.

#### Responsável pelo projecto “TruData”

Nome: Rogério Luiz Araújo Carminé



#### O Participante:

*Declaro ter lido e compreendido este documento, bem como as informações verbais fornecidas e aceito participar nesta atividade. Permito a utilização dos dados que forneço de forma voluntária, confiando em que apenas serão utilizados para o trabalho de investigação e com as garantias de confidencialidade e anonimato que me são dadas pelo responsável do projecto. Autorizo a comunicação de dados de forma anónima a outras entidades que estabeleçam parceria com a Universidade do Porto e/ou Associação Fraunhofer Portugal Research para fins académicos e de investigação científica.*

Nome: \_\_\_\_\_

Assinatura: \_\_\_\_\_

## B.6 Results - Details

### B.6.0.1 Details about the Participants

Part. ID	Group	Age	Gender	IT Skills	Years of experience in data preparation	Data Preparation Contexts (Original Descriptions)
1	A	29	Male	Software Developer	Between 1 and 2 years (inclusive)	Business Intelligence (BI), Data Science, Implantação de Sistemas de Informação
2	B	36	Male	Software Developer	Above 2 years.	Teste de software, Implantação de Sistemas de Informação, Conferência de folhas de pagamento (extração dos dados do sistema em txt para conferência)
3	A	25	Male	Advanced User	Above 2 years.	Business Intelligence (BI), Big Data, Data Science
4	B	32	Female	Advanced User	Above 2 years.	Teste de software, Implantação de Sistemas de Informação
5	A	26	Male	Software Developer	Between 1 and 2 years (inclusive)	Data Science
6	A	29	Male	Advanced User	Above 2 years.	Big Data, Data Science
7	B	45	Female	Advanced User	Above 2 years.	Teste de software, Implantação de Sistemas de Informação, Relatórios gerenciais

Table B.1: Information about the Participants

### B.6.0.2 Answers about Time, Effort, Satisfaction and Other Aspects

Item	Group	Part. ID	Task	Duration (Minutes)	Complete?	Assists	Errors
1	A	1	1	18.5500	Yes	2	2
2	A	1	2	3.4667	Yes	1	1
3	A	1	3	4.3833	Yes	1	1
4	B	2	1	1.7833	Yes		
5	B	2	2	3.7667	Yes		1
6	B	2	3	1.1833	Yes		
7	A	3	1	4.3333	Yes	2	2
8	A	3	2	1.4333	Yes	2	1
9	A	3	3	0.9167	Yes	2	1
16	B	4	1	4.3167	Yes	1	
17	B	4	2	1.4833	Yes		
18	B	4	3	1.5833	Yes		
10	A	5	1	3.8500	Yes		1
11	A	5	2	3.1833	Yes	1	2
12	A	5	3	0.8500	Yes		
13	A	6	1	13.1167	Yes	5	3
14	A	6	2	3.3333	Yes	2	1
15	A	6	3	2.0167	Yes	1	
19	B	7	1	19.3333	Yes	3	3
20	B	7	2	8.0167	Yes	3	1
21	B	7	3	5.1500	Yes	1	0

Table B.2: Completion Time, Errors and Assists per Participant and Task

Part. ID	Group	Task 1 Effort	Task 1 Satisfaction	Task 2 Effort	Task 2 Satisfaction	Task 3 Effort	Task 3 Satisfaction	Ratings and Comments Influence	Frequent Use	Easy to Use	Learn Rapidly
1	A	4	2	1	5	1	5	5	5	5	5
2	B	2	2	1	5	1	5	5	5	5	4
3	A	1	3	1	5	1	5	3	5	5	4
4	B	5	3	1	5	1	5	4	5	5	5
5	A	2	1	1	1	1	1	1	4	5	5
6	A	2	2	1	5	1	3	4	4	4	5
7	B	2	3	2	4	1	5	1	5	5	5

Table B.3: Results - Effort, Satisfaction, and Other Aspects per Participant and Task

### B.6.0.3 Task 1 - Approach - Original Descriptions

The original descriptions of the approaches used to complete the first task are presented below:

- Participant 1: *Foi implementada uma abordagem com utilização de dicionários para mapeamento dos valores, utilizando a biblioteca Pandas para carregamento do dataset e o método apply para transformar o atributo.*
- Participant 2: *1. Extraí valores únicos da coluna. 2. Classifiquei os valores únicos como 0 ou 1. 3. Utilizei a função de procura para associar o valor original ao valor mapeado no passo anterior.*
- Participant 3: *Selecionando as possibilidades de respostas positivas (SIM e demais variações). Atribuindo o valor 1 caso a comparação fosse verdadeira (se o valor comparado estivesse na lista de possibilidades positivas), caso contrário atribuindo o valor 0.*
- Participant 4: *Apliquei o filtro na coluna symptoms-fever e digitei manualmente os valores corretos, considerando a legenda descrita na tarefa e ainda a minha interpretação.*
- Participant 5: *substituição dos valores da coluna especificada por valores definidos. Processo iterativo em que fui verificando os valores diferentes do valor pretendido, e adicionando o valor diferente à lista de valores a substituir.*
- Participant 6: *Criacao de um dicionario/map entre values e keys. Aplicar esse map na coluna correspondent do dataframe com a funcao .map*
- Participant 7: *Inicialmente com fórmula SE, porém como a fórmula somente verifica um valor por vez foi mudada para fazer manualmente.*

## B.7 Tasks - Statistical Analysis Results

Metric	Task 1	Task 2	Task 3
Average Completion Time (minutes)	9.33	3.53	2.30
Average Effort (1 to 5)	2.57	1.14	1.00
Average Satisfaction (1 to 5)	2.29	4.29	4.14

Table B.4: Comparison based on time, effort and satisfaction metrics

Task	Average	Standard Deviation	Minimum	Maximum
1	9.33	7.49	1.78	19.33
2	3.53	2.20	1.43	8.02
3	2.30	1.75	0.85	5.15

Table B.5: Comparison based on completion time metric - Statistical Analysis

Task	Average	Standard Deviation	Minimum	Maximum
1	2.57	1.40	1	5
2	1.14	0.38	1	2
3	1.00	0.00	1	1

Table B.6: Comparison based on Effort Level - Statistical Analysis

Task	Average	Standard Deviation	Minimum	Maximum
1	2.29	0.76	1	3
2	4.29	1.50	1	5
3	4.14	1.57	1	5

Table B.7: Comparison based on Satisfaction Level - Statistical Analysis

Metric	Task 1	Task 2	Task 3
Errors	11	7	2
Assists	13	9	5

Table B.8: Comparison based on Numbers of Errors and Assistance

Task	Group	Average	Standard Deviation	Minimum	Maximum
1	A	9.96	7.14	3.85	18.55
1	B	8.48	9.49	1.78	19.33
2	A	2.85	0.95	1.43	3.47
2	B	4.42	3.32	1.48	8.02
3	A	2.04	1.65	0.85	4.38
3	B	2.64	2.18	1.18	5.15

Table B.9: Group Comparison per Completion Time - Statistical Analysis

Task	Group	Average	Standard Deviation	Minimum	Maximum
1	A	2.25	1.26	1	4
1	B	3.00	1.73	2	5
2	A	1.00	0.00	1	1
2	B	1.33	0.58	1	2
3	A	1.00	0.00	1	5
3	B	1.00	0.00	1	5

Table B.10: Group Comparison per Effort Level - Statistical Analysis

Task	Group	Average	Standard Deviation	Minimum	Maximum
1	A	2.00	0.82	1.00	3
1	B	2.67	0.58	2.00	3
2	A	4.00	2.00	1.00	5
2	B	4.67	0.58	4.00	5
3	A	3.50	1.91	1.00	5
3	B	5.00	0.00	5.00	5

Table B.11: Group Comparison per Satisfaction Level - Statistical Analysis

## B.8 Original Comments from the Participants

The usability test was carried out with a Portuguese-speaking audience. The original comments from the participants are available in this section for further analysis:

- *O teste está adequado a realização das tarefas, descrevendo detalhadamente os passos a serem cumpridos. Sobre o TruData, é fácil de utilizar e os itens de navegação estão claros e objetivos. Gostei da combinação de cores e tipografia. A avaliação (estrelinhas) por outros usuários demonstram a confiabilidade do operador, porém, na minha perspectiva não chamou a atenção no momento da utilização do operador;*
- *Ferramenta facilitadora para preparação de dados de fácil utilização, demandando pouco tempo para a execução e finalização da atividade; Recomendações: no documento de teste de usabilidade na parte de excel é importante detalhar na tarefa 2 item b, subitem ii, sobre*

*a necessidade de verificação das colunas no script com o próprio excel, pois devido ser apenas um usuário de excel e não ter conhecimento de código-fonte, essa verificação pode passar despercebida, levando o usuário a sentir-se desconfortável e/ou ter certeza de que o código não é algo malicioso; Em geral, usaria mais vezes, pois facilita o trabalho reduzindo o tempo em atividade manual;*

- *Geração de um package para evitar o acto de copiar código ou fazer download do source code;*
- *Sobre a proposta: acredito que a ferramenta possui bastante potencial caso seja corretamente difundida. Sobre a utilização: bastante intuitiva. Sugestões: Dependendo do público alvo (caso sejam usuários com domínio básico de programação) o upload de dados e correção automática pode ser considerado;*
- *A utilização da ferramenta será mais vantajosa quanto mais complexo ser o problema, podendo ser desnecessária a sua utilização em problemas bastante triviais, mas bastante útil em p.e. alteração de valores em colunas relacionais (dados tabulares), processamento de texto ou operações em imagens. Será interessante ver a ferramenta aplicada a outros domínios: imagem e também com outras linguagem de programação: c++;*
- *Status de segurança do código compartilhado. Botão search adicionar "Operator". Talvez arquivo exemplo;*
- *Possibilidade de oferecer sugestões para os termos utilizados na pesquisa por operados; na descrição dos operadores, diferenciar o código de exemplo do código-fonte (botões, cores, etc); na descrição dos operadores, possibilidade de oferecer uma amostra de dataset para o exemplo apresentado.*





# References

- [1] Ziawasch Abedjan, Lukasz Golab, and Felix Naumann. Data Profiling. In *Proceedings of the 2017 ACM International Conference on Management of Data*, volume Part F1277, pages 1747–1751, New York, NY, USA, may 2017. ACM. ISBN 9781450341974. doi: 10.1145/3035918.3054772. URL <http://dx.doi.org/10.1145/3035918.3054772><https://dl.acm.org/doi/10.1145/3035918.3054772>.
- [2] Ricardo Almeida, Paulo Maio, Paulo Oliveira, and João Barroso. An ontology-based methodology for reusing data cleaning knowledge. *IC3K 2015 - Proceedings of the 7th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management*, 2(Ic3k):202–211, 2015. doi: 10.5220/0005596402020211. URL <https://doi.org/10.5220/0005596402020211>.
- [3] Angular. Angular - What is Angular?, 2019. URL <https://angular.io/guide/what-is-angular>.
- [4] N Askham, D Cook, M Doyle, H Fereday, M Gibson, U Landbeck, R Lee, C Maynard, G Palmer, and J Schwarzenbach. The Six Primary Dimensions for Data Quality Assessment. Technical report, 2013. URL <https://www.dqglobal.com/wp-content/uploads/2013/11/DAMA-UK-DQ-Dimensions-White-Paper-R37.pdf>.
- [5] C. Batini, M. Lenzerini, and S. B. Navathe. A Comparative Analysis of Methodologies for Database Schema Integration. *ACM Computing Surveys (CSUR)*, 18(4):323–364, 1986. ISSN 15577341. doi: 10.1145/27633.27634.
- [6] Katrin Braunschweig, Julian Eberius, Maik Thiele, and Wolfgang Lehner. The State of Open Data: Limits of Current Open Data Platforms Categories and Subject Descriptors. Technical report, 2012. URL <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.309.8903>.
- [7] Xu Chu and Ihab F. Ilyas. Qualitative data cleaning. In *Proceedings of the VLDB Endowment*, volume 9, pages 1605–1608, 2016. doi: 10.14778/3007263.3007320. URL <https://doi.org/10.14778/3007263.3007320>.
- [8] Xu Chu, Ihab F. Ilyas, Sanjay Krishnan, and Jiannan Wang. Data cleaning: Overview and emerging challenges. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, volume 26-June-20, pages 2201–2206, 2016. ISBN 9781450335317. doi: 10.1145/2882903.2912574.
- [9] Thomas H Davenport, A Blanton Godfrey, and Thomas C Redman. To Fight Pandemics, We Need Better Data. *MIT Sloan Management Review*, 62(1):1–4, 2020. URL <https://pesquisa.bvsalud.org/portal/resource/en/mdl-2444683468>.

- [10] Caroline Duvier, Daniel Neagu, Crina Oltean-Dumbrava, and Dave Dickens. Data quality challenges in the UK social housing sector. *International Journal of Information Management*, 38(1):196–200, feb 2018. ISSN 02684012. doi: 10.1016/j.ijinfomgt.2017.09.008. URL <https://www.sciencedirect.com/science/article/abs/pii/S0268401216308222>.
- [11] SB Elmasri, R and Navathe. *Fundamentals of Database Systems*. Addison-Wesley, 6th edition, 2010.
- [12] Nurul A. Emran. Data completeness measures. In *Advances in Intelligent Systems and Computing*, volume 355, pages 117–130. Springer Verlag, 2015. ISBN 9783319173979. doi: 10.1007/978-3-319-17398-6\_11. URL [https://link.springer.com/chapter/10.1007/978-3-319-17398-6\\_11](https://link.springer.com/chapter/10.1007/978-3-319-17398-6_11).
- [13] Theodore Forbath, Peter Brooks, and Anand Dass. Beyond Cost Reduction: Using Collaboration to Increase Innovation in Global Software Development Projects. In *2008 IEEE International Conference on Global Software Engineering*, pages 205–209. IEEE, aug 2008. ISBN 978-0-7695-3280-6. doi: 10.1109/ICGSE.2008.32. URL <http://ieeexplore.ieee.org/document/4638668/>.
- [14] Dan Fowler, Jo Barratt, and Paul Walsh. Frictionless Data: Making Research Data Quality Visible. *International Journal of Digital Curation*, 12(2):274–285, may 2018. ISSN 1746-8256. doi: 10.2218/ijdc.v12i2.577. URL <http://www.ijdc.net/article/view/577>.
- [15] Mehrdad Ghayoumi. Review of Security and Privacy Issues in e-Commerce. Technical report, 2016. URL [https://search.proquest.com/conference-papers-proceedings/review-security-privacy-issues-e-commerce/docview/1806558682/se-2?accountid=14511%0Ahttps://ucl-new-primo.hosted.exlibrisgroup.com/openurl/UCL/UCL{}\\_VU2?url{}\\_ver=Z39.88-2004{&rft{}\\_val{}\\_fmt=info:ofi/fmt:k](https://search.proquest.com/conference-papers-proceedings/review-security-privacy-issues-e-commerce/docview/1806558682/se-2?accountid=14511%0Ahttps://ucl-new-primo.hosted.exlibrisgroup.com/openurl/UCL/UCL{}_VU2?url{}_ver=Z39.88-2004{&rft{}_val{}_fmt=info:ofi/fmt:k).
- [16] Jiawei Han, Micheline Kamber, and Jian Pei. *Data Mining : Concepts and Techniques : Concepts and Techniques (3rd Edition)*. Elsevier, 2012. ISBN 978-0-12-381479-1. URL <http://linkinghub.elsevier.com/retrieve/pii/B9780123814791000010>.
- [17] Yeye He, Kris Ganjam, Kukjin Lee, Yue Wang, Vivek Narasayya, Surajit Chaudhuri, Xu Chu, and Yudian Zheng. Transform-Data-by-Example (TDE). In *Proceedings of the 2018 International Conference on Management of Data*, pages 1785–1788, New York, NY, USA, may 2018. ACM. ISBN 9781450347037. doi: 10.1145/3183713.3193539. URL <https://dl.acm.org/doi/10.1145/3183713.3193539>.
- [18] Bernd Heinrich, Marcus Hopf, Daniel Lohninger, Alexander Schiller, and Michael Szubartowicz. Data quality in recommender systems: the impact of completeness of item content data on prediction accuracy of recommender systems. *Electronic Markets*, pages 1–21, aug 2019. ISSN 14228890. doi: 10.1007/s12525-019-00366-7. URL <https://doi.org/10.1007/s12525-019-00366-7>.
- [19] Emily Henriette, Mondher Feki, and Imed Boughzala. Digital transformation challenges. In *Mediterranean Conference on Information Systems*, 2016. ISBN 9789612861704. URL <http://aisel.aisnet.org/mcis2016/33>.

- [20] Carlos a Heuser. *Projeto de Banco de Dados*, volume 14. Porto Alegre Sagra Luzzatto, 1998. ISBN 9788577803828.
- [21] Stefanie Hossmann, Alan G. Haynes, Adrian Spoerri, Ibrahima Dina Diatta, Barry Aboubacar, Matthias Egger, Felix Rintelen, and Sven Trelle. Data management of clinical trials during an outbreak of Ebola virus disease. *Vaccine*, 37(48):7183–7189, nov 2019. ISSN 18732518. doi: 10.1016/j.vaccine.2017.09.094.
- [22] Ihab F. Ilyas and Xu Chu. *Data Cleaning*. Association for Computing Machinery, jul 2019. ISBN 9781450371520. doi: 10.1145/3310205. URL <https://dl.acm.org/citation.cfm?id=3310205{%&}picked=prox>.
- [23] Zhongjun Jin, Yeye He, and Surajit Chauduri. Auto-transform. *Proceedings of the VLDB Endowment*, 13(12):2368–2381, aug 2020. ISSN 2150-8097. doi: 10.14778/3407790.3407831. URL <https://dl.acm.org/doi/10.14778/3407790.3407831>.
- [24] Niklas Johansson, Anton Löfgren, and Carl Magnus Olsson. *Designing for Extensibility: An action research study of maximizing extensibility by means of design principles*. PhD thesis, Goteborgs, 2009. URL <http://hdl.handle.net/2077/20561>.
- [25] Sean Kandel, Jeffrey Heer, Catherine Plaisant, Jessie Kennedy, Frank Van Ham, Nathalie Henry Riche, Chris Weaver, Bongshin Lee, Dominique Brodbeck, and Paolo Buono. Research directions in data wrangling: Visualizations and transformations for usable and credible data. *Information Visualization*, 10(4):271–288, 2011. ISSN 14738716. doi: 10.1177/1473871611415994.
- [26] Jeonghyun Kim. Who is Teaching Data: Meeting the Demand for Data Professionals. *Journal of Education for Library and Information Science*, 57(2):161–173, mar 2016. ISSN 0748-5786. doi: 10.3138/jelis.57.2.161. URL <https://www.utpjournals.press/doi/abs/10.3138/jelis.57.2.161>.
- [27] Ferenc Kiss. Credit scoring processes from a knowledge management perspective. *Periodica Polytechnica Social and Management Sciences*, 11(1):95–110, 2003. ISSN 15873803.
- [28] Saranath Lawpoolsri, Jaranit Kaewkungwal, Amnat Khamsiriwatchara, Ly Sovann, Bun Sreng, Bounlay Phommasack, Viengsavanh Kitthiphong, Soe Lwin Nyein, Nyan Win Myint, Nguyen Dang Vung, Pham Hung, Mark S. Smolinski, Adam W. Crawley, and Moe Ko Oo. Data quality and timeliness of outbreak reporting system among countries in Greater Mekong subregion: Challenges for international data sharing. *PLoS Neglected Tropical Diseases*, 12(4), 2018. ISSN 19352735. doi: 10.1371/journal.pntd.0006425.
- [29] Lichuan Ma, Qingqi Pei, Lu Zhou, Haojin Zhu, Licheng Wang, and Yusheng Ji. Federated Data Cleaning: Collaborative and Privacy-Preserving Data Cleaning for Edge Intelligence. *IEEE Internet of Things Journal*, 8(8):6757–6770, apr 2021. ISSN 2327-4662. doi: 10.1109/JIOT.2020.3027980. URL <https://ieeexplore.ieee.org/document/9210000/>.
- [30] Mashaal Musleh, Mourad Ouzzani, Nan Tang, and AnHai Doan. CoClean: Collaborative Data Cleaning. In *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data*, pages 2757–2760, New York, NY, USA, jun 2020. ACM. ISBN 9781450367356. doi: 10.1145/3318464.3384698. URL <https://dl.acm.org/doi/10.1145/3318464.3384698>.

- [31] MySQL.com. MySQL :: MySQL 8.0 Reference Manual :: 1.2.1 What is MySQL?, 2021. URL <https://dev.mysql.com/doc/refman/8.0/en/what-is-mysql.html>.
- [32] Charith Perera, Rajiv Ranjan, Lizhe Wang, Samee U. Khan, and Albert Y. Zomaya. Big data privacy in the internet of things era. *IT Professional*, 17(3):32–39, 2015. ISSN 15209202. doi: 10.1109/MITP.2015.34.
- [33] Dessislava Petrova-Antonova and Romyana Tancheva. Data Cleaning: A Case Study with OpenRefine and Trifacta Wrangler. In *Communications in Computer and Information Science*, volume 1266 CCIS, pages 32–40. Springer, 2020. ISBN 9783030587925. doi: 10.1007/978-3-030-58793-2\_3. URL [https://link.springer.com/10.1007/978-3-030-58793-2\\_3](https://link.springer.com/10.1007/978-3-030-58793-2_3).
- [34] Gil Press. Cleaning Big Data: Most Time-Consuming, Least Enjoyable Data Science Task, Survey Says, 2016. URL <https://www.forbes.com/sites/gilpress/2016/03/23/data-preparation-most-time-consuming-least-enjoyable-data-science-task-survey-says/#65a725686f63>
- [35] Vijayshankar Raman and Joseph M. Hellerstein. Potter’s wheel: An interactive data cleaning system. Technical report, 2001. URL [www.fedstats.gov](http://www.fedstats.gov).
- [36] Thomas Schäffer and Christian Leyh. Master data quality in the era of digitization - toward inter-organizational master data quality in value networks: A problem identification. In *Lecture Notes in Business Information Processing*, volume 285, pages 99–113. Springer Verlag, nov 2017. ISBN 9783319588001. doi: 10.1007/978-3-319-58801-8\_9. URL [https://link.springer.com/chapter/10.1007/978-3-319-58801-8\\_9](https://link.springer.com/chapter/10.1007/978-3-319-58801-8_9).
- [37] Matthew Smith, Christian Szongott, Benjamin Henne, and Gabriele Von Voigt. Big data privacy issues in public social media. *IEEE International Conference on Digital Ecosystems and Technologies*, pages 10–15, 2012. ISSN 21504938. doi: 10.1109/DEST.2012.6227909.
- [38] Ian Sommerville. *Software Engineering*. Pearson, 9th edition, 2011. ISBN 978-0-13-703515-1.
- [39] Diomidis Spinellis. Git. *IEEE Software*, 29(3):100–101, may 2012. ISSN 0740-7459. doi: 10.1109/MS.2012.61. URL <http://ieeexplore.ieee.org/document/6188603/>.
- [40] Spring. Spring Boot, 2021. URL <https://spring.io/projects/spring-boot>.
- [41] Spring. Spring Data, 2021. URL <https://spring.io/projects/spring-data>.
- [42] Spring. Spring Framework, 2021. URL <https://spring.io/projects/spring-framework>.
- [43] Vicenç Torra and Guillermo Navarro-Arribas. Data privacy. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 4(4):269–280, 2014. ISSN 19424795. doi: 10.1002/widm.1129.
- [44] Vladimir L. Uskov, Jeffrey P. Bakken, Keerthi Sree Ganapathi, Kaustubh Gayke, Brandon Galloway, and Juveriya Fatima. Data Cleaning and Data Visualization Systems for Learning Analytics. In *Smart Innovation, Systems and Technologies*, volume 188, pages 183–197.

- Springer, 2020. ISBN 9789811555831. doi: 10.1007/978-981-15-5584-8\_16. URL [https://doi.org/10.1007/978-981-15-5584-8\\_{\\_}16](https://doi.org/10.1007/978-981-15-5584-8_{_}16).
- [45] Kush R. Varshney, Dennis Wei, Karthikeyan Natesan Ramamurthy, and Aleksandra Mojsilović. Data challenges in disease response: The 2014 ebola outbreak and beyond. *Journal of Data and Information Quality*, 6(2), 2015. ISSN 19361963. doi: 10.1145/2742550.
- [46] Varun Vasudevan, Abeynaya Gnanasekaran, Varsha Sankar, Siddarth A Vasudevan, and James Zou. Disparity in the quality of COVID-19 data reporting across India. *medRxiv*, 2020. doi: 10.1101/2020.07.19.20157248. URL <https://doi.org/10.1101/2020.07.19.20157248>.
- [47] Rakesh M. Verma, Victor Zeng, and Houtan Faridi. Poster: Data quality for security challenges: Case studies of phishing, malware and intrusion detection datasets. In *Proceedings of the ACM Conference on Computer and Communications Security*, pages 2605–2607, New York, NY, USA, nov 2019. Association for Computing Machinery. ISBN 9781450367479. doi: 10.1145/3319535.3363267. URL <https://dl.acm.org/doi/10.1145/3319535.3363267>.
- [48] Jinlin Wang, Xing Wang, Yuchen Yang, Hongli Zhang, and Binxing Fang. A review of data cleaning methods for web information system. *Computers, Materials and Continua*, 62(3): 1053–1075, 2020. ISSN 15462226. doi: 10.32604/cmc.2020.08675.
- [49] Jim Whitehead, Ivan Mistrík, John Grundy, and André van der Hoek. Collaborative Software Engineering: Concepts and Techniques. In *Collaborative Software Engineering*, pages 1–30. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010. ISBN 9783642102936. doi: 10.1007/978-3-642-10294-3\_1. URL [http://link.springer.com/10.1007/978-3-642-10294-3\\_{\\_}1](http://link.springer.com/10.1007/978-3-642-10294-3_{_}1).
- [50] Anna M. Wichansky. Usability testing in 2000 and beyond. *Ergonomics*, 43(7): 998–1006, jul 2000. ISSN 0014-0139. doi: 10.1080/001401300409170. URL <https://doi.org/10.1080/001401300409170https://www.tandfonline.com/doi/full/10.1080/001401300409170>.
- [51] Rüdiger Wirth. CRISP-DM : Towards a Standard Process Model for Data Mining. In *Proceedings of the Fourth International Conference on the Practical Application of Knowledge Discovery and Data Mining*, number 24959, pages 29–39. Practical Application Co, 2000. doi: 1902426088.
- [52] Munir Kolapo Yahya-Imam and Felix O. Aranuwa. An Empirical Study on Big Data Analytics: Challenges and Directions. In *Lecture Notes in Networks and Systems*, volume 118, pages 669–677. Springer, 2020. doi: 10.1007/978-981-15-3284-9\_76. URL [https://doi.org/10.1007/978-981-15-3284-9\\_{\\_}76](https://doi.org/10.1007/978-981-15-3284-9_{_}76).