UNIVERSITY OF SALERNO

Department of Computer Science

Master of Science in Computer Science

MASTER'S DEGREE THESIS

Investigating Data Smells and their Influence on Data Quality: an In-Depth Analysis

SUPERVISOR

Prof. **Fabio Palomba**

CO-SUPERVISORS

Dr. **Carmine Ferrara**

Dr. **Gilberto Recupito**

University of Salerno

CANDIDATE

Raimondo Rapacciuolo

0522501266

Academic Year 2022-2023

This thesis was carreied out at the

When you think of A.I., it's forward-looking, but A.I. is based on data, and data is a reflection of our history." -Joy Buolamwini

Abstract

In today's era of rapid technological advancement, machine learning-intensive systems have become integral across various domains, from e-commerce, where they enhance customer experiences through personalized recommendations and optimize supply chains, to healthcare, where they aid in disease diagnosis and treatment optimization, these systems rely on machine learning algorithms to extract insights, predict outcomes, and automate intricate tasks.

The development and deployment of such systems necessitate the consideration of non-functional requirements that extend beyond traditional software criteria. As ethical concerns surrounding AI and machine learning become increasingly apparent, the concept of machine learning fairness emerges as a vital quality aspect, highlighting its role in addressing biases and ensuring equitable outcomes for all user groups. Fairness, alongside safety and others, are integral components of the overall quality of a system.

Furthermore, the quality of input data serves as the bedrock of these systems. Ensuring data quality entails addressing data anomalies, irregularities, and the issue of data smell to maintain reliability and effectiveness.

This work's primary motivation is to comprehend the progress in the realm of data smells, their presence in publicly available datasets used for machine learning model training, and their impact on functional and non-functional requirements. A systematic literature review was conducted to understand which data smells have been introduced, their impact on the properties of machine learning systems, and tools proposed in literature.

Secondly, this work investigates the potential impact of data smells on data quality issues. The aim is to understand the prevalence of data smells in the datasets and their extent of influence on data quality aspects, like completeness, uniqueness, consistency, readability and fairness.

Contents

List of Tables

CHAPTER 1

Introduction

1.1 Application Context

Nowadays in the context of contemporary technological advancements, machine learning-intensive systems have gained significant prominence across a variety of domains, starting from e-commerce, where they enhance customer experiences through personalized recommendations, demand forecasting, and supply chain optimization up to healthcare, where they aid in diagnosing diseases, predicting patient outcomes, and optimizing treatment plans. These systems leverage the power of machine learning algorithms to extract insights, make predictions, and automate complex tasks.

The quality of input data holds immense significance, as data forms the foundation of these systems. Ensuring data quality addressing the problem of data smell, anomalies and irregularities in data, becomes crucial to uphold the reliability and effectiveness of the machine learning processes.

Data quality, which encompasses the accuracy, completeness, and reliability of the data, is essential not only for the proper functioning of the system but also for addressing biases and striving for equitable outcomes across all user groups. In an era where ethical concerns surrounding AI and machine learning are increasingly

evident, ensuring data quality takes center stage as a vital component. Fairness, alongside safety and other factors, contributes to the overall quality of the system, with data quality forming the bedrock upon which these critical aspects are built.

1.2 Motivation and Goals

The main motivation that led this work was to understand the progress in the literature on the topics of data smells, their presence in the public datasets that can be used to train a machine learning model for a variety of tasks and domains, and their impact on its functional and/or non-functional requirements from both the points of view of researchers and developers who are interested in understanding how continuously improve the quality of ML systems.

Starting from this motivation we carried on a Systematic Literature Review (SLR) with the main goals of understanding: how many and which are the data smells defined in the literature; what is the impact of data smells on functional and non-functional properties of ml-intensive systems; and finally which datasets have been analyzed and which tools have been defined in the literature in the scope of data smells.

Later in this work we investigated on the possible impact that the presence of certain types of data smell could have on data quality issues. With the main goal of understanding: what is the prevalence of data smells in the analyzed datasets and to what extent data smells impact data quality aspects, like completeness, uniqueness, consistency, readability and fairness.

1.3 Results

The results of the SLR showed the definition of seven main classes of data smell, namely, Believability Smells, Encoding Smells, Syntactic Smells, Consistency Smells, Redundant Value Smells, Categorical Smells and Miscellaneous Smells, with a final catalog of 46 data smells, with 16 more data anomalies that can be found in the literature, but since many of them can be mapped to one of the smells defined in the previous groups, while others are related to the definition of the database schema

instead of the dataset itself they are out of our scope.

Then we found different examples of how lack of data quality can influence functional and/or non-functional requirements, such as defect proneness, safety, and maintainability, or more critical requirements like safety and fairness in systems that interact with the real world. Since the data are defined in the initial steps of the pipeline, the presence of data smells could raise a degradation of the model in combination with other issues related to technical debt specific to AI-based systems (i.e., Pipeline Jungles and Hidden Feedback Loops).

The last research question of the SLR showed that since smelly data cannot always be mapped to data errors, they could not be suitable to be detected by validation tools, this led Harald Foidl et al. [SLR1] to the development of two tools for data smell detection based on rules and machine learning and that the main study about the presence of data smells in public datasets has been carried out by Arumoy Shome et al. [SLR2] finding out that redundant value smells and the categorical value smells are the most common categories of smell.

Finally, in the second part of this study, we analyzed the correlation between 3 selected data smells, namely, Extreme Value Smell, Missing Value Smell and Suspect Sign Smell, and the metrics of Data Quality using a multiple regression, the results showed a strong impact of the Missing Value Smell on the metric of Completeness, and a lower impact of the three data smells on the Consistency, while other metrics like Uniqueness and Readability, may be more influenced by different smells like Duplicated Value Smells or different types of Encoding Smells.

1.4 Thesis Structure

The next chapters are structured as follow:

- **Chapter 2 - Background:** In the second chapter we will provide a background knowledge about the main aspects of a **machine learning-intensive system**, which are the most important **non-functional requirements** in this kind of systems, and a particular focus on the aspect of data quality and the ways to assess it.

- **Chapter 3 - A Systematic Literature Review on Data Smells:** In the third chapter we will carry on a Systematic Literature review on the topics of data quality and in particular data smells in order to understand how many and which types of data smells have been defined and the impact that they can have on a machine learning-based system and its NFRs and, finally, which are the tools and datasets reported in the state of the art in this scope.

- **Chapter 4 - On The Impact of Data Smells on ML Fairness:** In this chapter we will report how we carried on our research starting from the definition of research questions and the development of the working hypothesis, continuing with the data collection, the design and development of our tool and the analysis of the results.

- **Chapter 5 - Threats To Validity:** This chapter describes and illustrates the threats to the validity of our study and how we mitigated them.

- **Chapter 6 - Conclusions:** This final chapter sums up the processes and the findings of the two phases of this study.

CHAPTER 2

Background

In this chapter we provide a background on Machine Learning-Intensive systems, its non-functional requirements, the importance of data quality aspects with a particular focus on data management and data quality metrics.

2.1 ML-Intensive Systems

Machine Learning is an important branch of the Artificial Intelligence, it includes all the theoretical and practical aspects that allow a computer to learn based on past experience in order to resolve tasks like pattern recognition, classification, regression and others. A Machine Learning system may be defined as an application able to improve itself in a task T, with respect to a performance indicator P, based on experience E [1]. The three main factors to take in account when designing a machine learning system are the data, the model and the pipeline.

2.2 Data and Data Engineering

Nowadays data have become the most valuable aspect in almost every company or organization, leading to a data-driven point of view in development. Since there cannot be artificial intelligence without data and there cannot be accurate artificial intelligence without good data, the data governance process has received increasing attention. **Data Governance** is the management process that curates the availability, usability, integrity and safety of the data so this process is strictly correlated and have a main focus on data quality since good data have an higher value [1]. **Data Quality** include all the processes that describe the accuracy, completeness and consistency of the collected data [1]. In order to enable data quality we need **Data Engineering** which is the set of techniques and algorithms that enable the extraction, analysis and preparation of the data [1]. It can be considered the backbone of artificial intelligence.

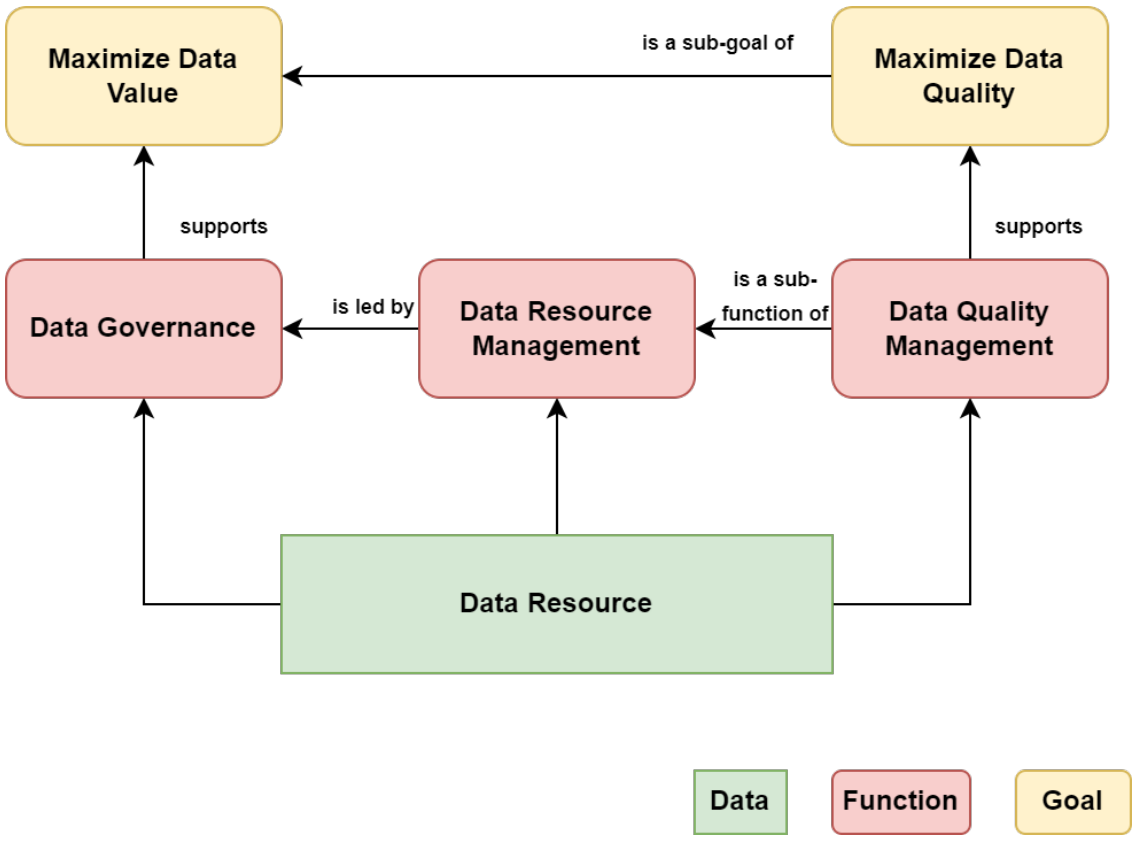

Figure 2.1: A closer look to the connection between Data Governance and Data Quality [1]

The main obstacles to data-driven organizations are the so called **data debts** [1], which are the application of sub-optimal data quality and data management processes, this lead data not to be meaningful enough, some of these are related to database skills while some are concerned with all the preprocessing steps in data extraction and data preparation. There are different types of this debt [1] including:

- **Structural Debts**: Issues with the design of the database like improperly named columns or insufficiently normalized data;
- **Data Quality Debts**: Issue with the consistency or usage of data values like duplicated business key values or corrupted data;
- **Integrity Debts**: Issues about the integrity of data between different tables;
- **Architectural Debts**: Issues about how external programs interact with the data source;
- **Documentation Debts**: Issues with any supporting documents, including models, some examples are inconsistent or outdated information;
- **Functional Debts**: Issues with execution aspects within the data source.

There are other more specific data debt in the literature that we will analyze afterwards. The work to deal with data debts id done by the figure of the **Data Engineer**, we can recognize three types of data engineers: The **Data Architect**, he is the responsible of the development and maintenance of the data engineering pipeline; The **Database Engineer** is the responsible of all the activities related to the database including designing, modeling, development and maintenance and the **Pipeline Engineer** who is involved in the steps of data cleaning and collection. There is another important role, namely the **Data Scientist** who is responsible for data analysis, he is not a data engineer, but the interaction between the two is fundamental because some data may need a transformation in order to become a feature before being stored, in fact **data** is any element available to solve a problem, while a feature is a characteristic of the problem that can be extracted from data [1].

Potentially, we can mine data from every source, but obviously we will have different types of data requiring different mining instruments. We can find **Structured**

Data like XML, JSON or CSV files, which are the easiest to mine and understand as they have a well-defined structure and they are often already organized and available. The other type is **Unstructured Data**, these are represented by any other form of file which doesn't follow a defined structured like natural language text, pictures and sounds which are hardest to mine as they require ad-hoc parsers. Data can give different type of information depending on the point of view and the problem of interest. For example, let's suppose we have this piece of code:

```
void method(...){
    noFirewall = new JRadioButton("no firewall");
    socksFirewall = new JRadioButton("no SOCKS 4/5
                        Firewall");
    webProxy = new JRadioButton("HTTP web proxy");
}
```

Listing 1: Example of code to mine

we can use the data in the code in **Listing 1** to answer these questions in order to gather different information:

Q1. How many objects are created in the method? **Structural information**

Q2. What does this method do? **Textual information**

Q3. When was the method introduced? **Historical information**

Q4. What is the execution time of the method? **Dynamic information**

2.3 Data Preparation

As we said there cannot be artificial intelligence without data and there cannot be accurate artificial intelligence without good data. While there exist a variety of instruments for data mining and their use depends on what we would like to mine, there are a number of steps that we need to conduct in order to make data quality high enough to build a reliable machine learning system.

The first step is **Data Cleaning**[1], data can be noisy or incomplete and this step aims at normalizing the extracted data using different methods based on the

problem and how much data are available, for example if we consider a dataset with missing values, the easiest way to deal with this problem is to remove the columns or the rows affected by missing data, but this can lead to a loss of information that can't be handled if the dataset isn't big enough, in this cases a solution can be **Data Imputation**[1] which are a set of methods used to deal with incomplete data, aiming at estimating them based on the data available or based on logical implications. Other tasks in data cleaning could be addressing formatting inconsistencies, removing redundant rows, if needed or removing redundant columns if needed, essentially, all the operations required to have a dataset which is the most reliable possible.

The second step is **Feature Construction**[1], namely,the process of building intermediate features from the original data, the aim of this phase is to build more efficient features to address the problem of interest. This is the step where a data scientist takes the lead, the first main thing to do is to *understand the domain*, in fact we cannot automate an operation if we don't know how to do it manually and with artificial intelligence we typically try to let machines do something that is too costly for humans. The second factor is the understanding of the data collected, a data scientist may explore them to measure how they can be reduced and manipulated, this would ease the job of feature construction enabling to work on a reduced set of possible features. A way to address this task is **Dimensionality Reduction**[1] which is an unsupervised transformation of data from a high-dimensional space into a low-dimensional space so that the low-dimensional representation retains some meaningful properties of the original data.

The third step is **Feature Scaling**[1], this step aims at reducing the problem when a set of values of a variable is too different from the set of values of another variable, this step tries to address the problem for which an algorithm might overestimate or underestimate the importance of one of the variables just because of the different distribution of values. There exist a set of techniques that enable the normalization of the distribution of values of the features like Min-Max normalization or Z-Score normalization.

The fourth step is **Feature Selection**[1], this is a process through which the most relevant features are identified, while the least relevant or redundant are filtered out. Redundant features are problematic because they risk to bias the inner-working

of artificial intelligence systems, because it cannot decide which one to use for its goals. A way to identify the redundant features is **Univariate feature removal**[1], this method relies on statistical methods to understand the correlation between pairs of variables, If two of them are highly correlated, one of them should be removed, typically the most difficult to understand.

The last step is **Data Balancing**[1], this is a crucial step for most of the problems, in fact in reality, most problems are unbalanced, so a class have much more instances than the other, and data balancing provides a set of techniques that convert an imbalanced dataset into a balanced one, an example is **SMOTE**[1] (Synthetic Minority Oversampling TEchnique) it generate synthetic instances based on the *k*-nearest neighbors of an element of the minority class.

2.4 Model

After the extraction and the preprocessing of data, they can be used to solve different problems. Based on the problem of interest there are different types of algorithms that can be used to address it. Starting with **Unsupervised Machine Learning**[1], it is used to find patterns in a set of data, it is unsupervised because it just relies on the input data, without any label or additional information on the characteristics of the problem. It is used to solve problems like *Clustering, Anomaly Detection, Pattern Recognition and Association*. Then we have **Supervised Machine Learning**[1] which is used to learn from data in order to predict future trends and/or recommend actions. It is supervised since it needs labeled data to learn from, different predictors can be used together to enable ensemble learning. Last we have **Reinforcement Learning**[1] which is based on learning by the rewards of the environment.

Whatever model is chosen, there are a set of steps to configure it. The first one is the **selection of the independent variables**, this aim at finding features that are valuable to solve the problem of interest, data and feature engineering enables a learner to rely on high-quality data correlated to the dependent variable. The second is one of the most crucial steps and it is the **selection of the dependent variables**, it concerns with the selection of the label that describe the problem. The third step is the **Configuration**, first of all , configuration implies the choice of a suitable

machine learning algorithm to use, there are several algorithms available , but certain consideration must be made. When we choose an algorithm we must consider all the assumptions made by it, for instance, a Naive-Bayes have an independence assumption: *all the variables must be independent from each other*, meaning that a feature must not influence another, this assumption implies that you cannot use Naive-Bayes without taking care of feature selection. As another example, Linear Regression requires data to be normally distributed, before using it, a normality test on your data should be run, if the test fails, we should find other solutions. Typically, more complex algorithms (e.g., Logistic Regression) relax the assumptions made by their simpler versions, enabling their adoption. The choice of the algorithm is not the only task of the configuration, **hyper-parameters** are parameters whose value can be used to control the learning process, but they are not parameters that can be learned from the training data.

2.5 Optimization

In order to optimize the learning process the hyper-parameters should be tuned in the best way possible, it means provide the learner with the optimal instruments to let it learn. A way to find a good configuration is using a search-based algorithm: *"algorithm used to retrieve information stored within a data structure or computed in a more complex search space of a problem domain, with either discrete o continuous values"*[1]. An optimization procedure involves defining a search space, this can be thought of geometrically as an n-dimensional volume, where each hyper-parameter represents a different dimension and the scale of the dimension are the values that the hyper-parameter may take on, such as real-valued, integer-valued, or categorical and a point of the search space represents a configuration, namely a vector with a specific value for each hyper-parameter value. The goal of the optimization procedure is to find a vector providing the best performance of the model after learning. The two simplest algorithms are known as Random Search and Grid Search. Random Search define the search space as a bounded domain of hyper-parameter values and randomly sample points in that domain, this algorithm is quite efficient as it is bounded to the maximum combinations to try, yet, it may return a sub-optimal

solution that only slightly improve the performance over the default configuration. Grid Search define a search space as a grid of hyper-parameter values and evaluate every position in the grid. A grid search is effective for spot-checking combinations that are known to perform well generally. Random search is effective to discover and get hyper-parameter combinations that you would not have guessed intuitively, although it often requires more time to execute. A way to improve the research are genetic algorithms, An individual might be set as a vector of hyper-parameter values, a fitness function as the resulting accuracy of the model set using a certain vector of hyper-parameters. Evolutionary optimization is more accurate than basic algorithms since it explores the search space in a smarter way, providing solutions that are closer to the optimum.

2.6 Life Cycle and Pipeline

The development of a Machine Learning intensive system, as any software project, can be engineered defining its own life cycle. The life cycle of machine learning models describes how models go from development to production, it can be seen as the following set of steps [2]:

1. Feature Engineering;
2. Training;
3. Tuning;
4. Model Store;
5. Serving;
6. Monitoring;
7. Update.

As shown in figure 4.2, development is just the first phase, then it's crucial to enable inference on the new model and update it if necessary. Based on these steps, new approaches try to automate this life cycle, there are several challenges to this, in

first place different roles with different abilities have to interact, and the process is non-linear and very iterative. Despite this, as the models in an organization increase, automation becomes vital for success. The automation of the life cycle create a, so called, **Machine Learning Pipeline** which is a sequence of tasks that starts from the initial dataset arriving to the model deployment and its evolution, so it's not only the model that is put in production, but the whole pipeline in order to meet the evolving needs of the system. In such a scenario maintenance of the pipeline become crucial and as said by Zhou et al. [3] ignoring this step may lead to considerable technical debt. For example, machine learning has been introduced into areas with high safety requirements, such as autonomous driving technology and paramedical diagnostics, the quality and privacy need to be assured before application serving, which requires testing and validation on both datasets and trained models. In response to these needs, new tools and platforms becomes more and more popular, providing embedded systems that can preprocess data, re-train models and deploy models. The birth of this automation systems led to specific agile development practices and cultures, in such a scenario the most important is, indeed, **MLOps** [3] which is a specialization of the most generic **DevOps**. DevOps is based on a set of practices that combine development and operations, and aims at reducing the overspecialization [1], fading the edges between roles. MLOps specialize the principles of DevOps using them in AI-based systems aiming at the continuous delivery of high performance models [3]. A standard MLOps-based pipeline [3] is showed in **Figure 4.2** and it's composed of different activities, we have to distinguish development processes including:

1. Data Extraction and Preprocessing;
2. Data Analysis and Preparation;
3. Training and Testing of the model;
4. Tuning of the model;
5. Model Validation.

From the operational processes, generally executed from specialized figures, including:

1. Model Deployment;
2. Monitoring;
3. Reporting;
4. Feedback retrieval.

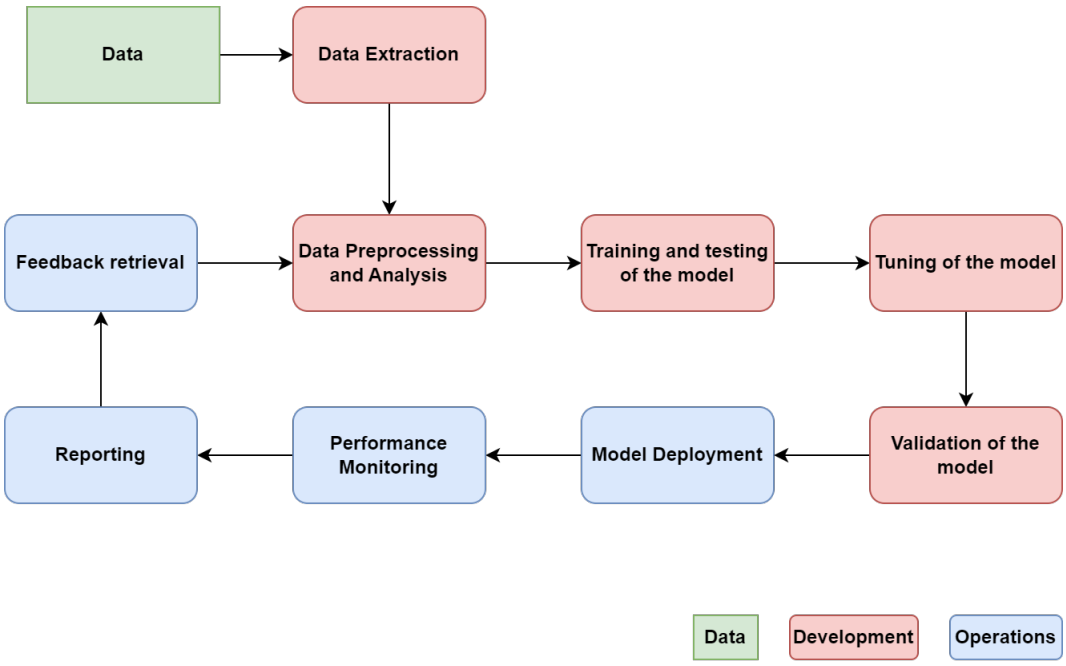

Figure 2.2: Generic MLOps-based Pipeline

2.7 Data Quality

Nowadays is well documented in the literature that the performance of a machine learning model is upper bounded by the quality of the data, despite this, while researchers and practitioners have focused on improving the quality of models there is not as much effort towards improving data quality management processes [4].

Data quality is studied in numerous domains, for instance in machine learning the assessment of data quality plays a key role in the evaluation of the usefulness of data collected, a variety of tools and metrics have been designed for the purpose,

Cases Inconsistency Level (CIL) is a metrics for analyzing conflicts in software engineering datasets [5], but there are many other domains in which data quality is highly studied including cyber-physical systems, assisted living systems, smart cities, big data management, IoT and many other [6]. In such a scenario a lack of data quality manifests in several forms, including missing, incomplete, inconsistent, inaccurate dated or duplicate data and this can lead to several problems related to different field like fairness or safety.

Organizations often underestimate the implications of a poor data quality management process. The consequences of not caring about the data quality could lead to catastrophic damage to companies. The Data Warehousing Institute (TDWI)[1] estimates that poor quality costs businesses in the US over $700 billion annually.

2.7.1 Data Quality Dimensions and Metrics

Otmane Azeroual et al. [7] define data quality as "multi-dimensional measure of the suitability of data to fulfill the purpose bound in its acquisition/generation. This suitability may change over time as needs change". Therefore, when we talk about data quality we are talking of the reliability of data in a certain point in time [7], this definition clearly shows that data quality is continuously influenced by the evolving changes over time. This aspect leverages the need to define standards and guidelines to measure and evaluate it. From this point of view data quality can be seen as the totality of different quality dimensions and data quality metrics are necessary to evaluate this dimensions.

In the grey literature different dimensions of the data have been analyzed [8], Data Quality Dimensions are big family of metrics that help measure different aspects of the same data in accordance with the project's goals and technical limitations. This are the dimensions reported:

- **Completeness**: A dimension that measures whether the data is present (non-blank values in data) or absent (null or blank values; the value is missing).
- **Conformity/Validity**: A measure of how well the data acts according to internal and external standards, guidelines, or standard definitions.

[1] https://tdwi.org/Home.aspx

- **Consistency**: A dimension that refers to whether the same data stored in different places does or does not match. A measure of the degree of uniformity of data as it moves across the network or between applications on a computer. Data values in one dataset must match the values in the other datasets.
- **Integrity**: Captures the relationships between data objects, the validity of data across these relationships, and the guarantee that the data can be traced and connected to other related data.
- **Accuracy**: A measure of how well the data reflects the real-world scenario and how correctly it can describe the reality; a measure of correctness of the content.
- **Correctness**: Similar to accuracy. The characteristic of whether the data is free of errors or mistakes.
- **Granularity**: A measure of the level of detail in a data structure.
- **Precision**: A degree of detail of the measurement and thoroughness of description for a data element.
- **Accessibility**: A measure of the extent to which data or metadata is provided and shared in open formats, as well as the suitability of the data.
- **Time-related dimensions**: Timeliness, volatility, and currency/freshness of data.

This metrics can impact several quality aspects of the AI model like transparency, Testability, Reliability, and Fairness.

Data quality dimensions work together, having multiple dimensions helps organizations make better data-driven choices about where the time and effort should be devoted to. Only by using several dimensions at a time, one can understand what can be learned from the various data quality evaluations [8].

2.7.2 Non-functional Requirements in ML Systems

Since, as already defined, a machine learning model is strongly influenced by the data we use to train it, a concept related to data quality is its functional and non-functional behaviour. To ensure the success of ML-enabled systems, it is essential to be aware of certain qualities of ML solutions, known from a Requirement Engineering perspective as non-functional requirements (NFRs) [9]. However, when systems involve ML, NFRs for traditional software may not apply in the same ways; some NFRs may become more or less important;

As showed by Habibullah et al.[9], most of the NFRs defined for traditional software systems are still relevant in an ML context, while only a few become less prominent. According to their interviews the most important include **fairness**, but also **flexibility**, **usability**, **accuracy**, **efficiency**, **correctness**, **reliability**, and **testability**.

2.7.3 Fairness

Nowadays machine learning systems work and influence the real environment, in such a scenario their application needs to meet ethical regulations to be actually used. There are two main point of view when talking about ethical machine learning. Firstly, the **legal** aspects consider fairness in accordance to societal laws. Then, **ethical** aspects which means following moral principles of tradition, group or individuals without any legal binding. Ethical machine learning tries to address the problem of discrimination, unequal treatment, unequal outcomes obtained by reinforcing patterns in predicting policing with feedback loops. Unfortunately, voluntarily or not, data can perpetuate society's existing race-, class- and gender-based inequities. Software Fairness is a hot topic nowadays, Horkoff describes the software fairness as a non-functional requirement for machine learning-based systems, it aims at making machine learning algorithms unbiased by dealing with sensitive features and defining systems that take into account the level of fairness required for their own application domain [9].

There are plenty of examples of how a bias in the dataset led to unfortunate inconveniences, the most famous are:

- *Amazon recruiting system*: Amazon developers made known that in 2015 their recruiting tool for technical positions didn't offer equal job opportunities based on the sex of the candidate, due to a bias in the training data, in fact the model was trained on a dataset of employees from the past 15 years, and the majority were men. This led Amazon to close the project as officially reported [10].

- *US healthcare*: US hospitals used , for years, a prediction system for the medical care that started to prefer white patients more than black patients, due to a bias in the data related to the cost of the healthcare, which considered white people more convenient. This translated in considering black people healthier [11].

- *Google translate*: Google translate, one of the most used tool for automatic translation, showed a sex-related bias. In particular when translating "She is an engineer, He is a nurse", from English to Turkish the subjects were reversed becoming "He is an engineer, She is a nurse", associating the technical job to a man [12].

- *COMPAS*: In 2014 a tool called Correctional Offender Management Profiling for Alternative Sanctions (COMPAS), used in the prediction of future crimes responsible based on facial analysis starting from a dataset of criminals, has been analyzed and it showed a tendency in judging black people more incline to violent crimes, due to a bias in the dataset which contained sensible features like sex and race [13].

Fairness Metrics and Definitions

Defining when a machine learning system act fair is not easy, in the state of the art Verma and Rubin [14] introduced different definition of software fairness based on different metrics and points of view. First of all we need to introduce some basic concepts:

1. **Protected or sensitive attribute**: We refer to a protected or sensitive attribute as an attribute of the dataset for which a machine learner may produce some form of discrimination. For instance, attributes referring to race or gender might be considered protected.

2. **Protected or sensitive group**: We refer to a protected or sensitive group as a group of individuals who might assume a value of a protected attribute that may lead to discrimination. For instance, in the known problem of recruitment, black people or women might be considered as protected groups.

3. **Real value of classification**: We refer to the real value of classification as the category to which an individual belongs to and which is computed on the basis of her/his features.

4. **Prediction probability**: We refer to the prediction probability as the conditioned probability that an individual is classified as belonging to a certain category.

5. **Prediction**: We refer to the prediction as the decision taken by an algorithm with respect to the category to assign to a certain individual.

Based on these, different definitions have been formulated.

Statistical Measures

This type of measures are based on some statistical metrics like true positive and negatives, false positive and negatives, precision and recall [14].

Following some of the most important definition from this group of measures are reported:

- **Group fairness, a.k.a. Statistical parity**: According to this definition a classifier is fair if elements in both protected and unprotected groups have equal probability of being assigned to the positive predicted class.

$$P(d = \text{positive}|g = g1) = P(d = \text{positive}|g = g2) \quad (2.7.1)$$

  Where d is the prediction, g1 is the protected group and g2 is the unprotected group.

- **Predictive parity, a.k.a. Outcome test**: According to this definition, a binary machine learner is fair if both protected and non-protected groups have the same probability of assuming the real value of classification, so the probability of individuals to be classified as true positive/negative is the same.

$$(Y = \text{positive}|d = \text{positive}, g = g1) = P(Y = \text{positive}|d = \text{positive}, g = g2) \quad (2.7.2)$$

  Where d is the prediction, Y is the actual label, g1 is the protected group and g2 is the unprotected group.

- **False positives Error rate balance, a.k.a. Predictive equality**: According to this definition, a binary machine learner is fair if the probability of individuals to be associated to the positive class even though the real value of classification is opposite is the same for both protected and non-protected groups, so the probability of being a false positive is the same.

$$P(d = \text{positive}|Y = \text{negative}, g = g1) = P(d = \text{positive}|Y = \text{negative}, g = g2) \quad (2.7.3)$$

  Where d is the prediction, Y is the actual label, g1 is the protected group and g2 is the unprotected group.

Since every definition consider a different point of view it may be a good idea to consider a combination of different definition in order to have a better idea of the fairness of the model.

Similarity-Based Measures

The statistical definitions suffer from a key issue: all are based on the concepts of protected groups, so they all define fairness only considering the protected attributes without taking into account other implementation constraints, this might potentially hide unfairness. If we consider the case of statistical parity in a case of recruitment: the same fraction of both genders applicants may be assigned a positive score, yet male applicants may still be chosen at random while female applicants may only be those that have the most savings, so without considering the whole set of attributes hiding a problem of "unfair fairness". In order to address this problem some measures based on the similarity of individuals [14], considering protected and non-protected attributes, have been introduced:

- **Casual discrimination**: This definition is based on the concept that a machine learner should return the same classification for every pair of subjects of different groups with the same unprotected attributes. For instance, this implies that a male and female applicants who otherwise have the same unprotected attributes X will either both be hired or not.

  $$\text{if } X_1 = X_2 \text{ and } G_1 \neq G_2 \text{ then } d_1 = d_2 \quad (2.7.4)$$

  Where X_1 and X_2 are the unprotected attributes, G_1 and G_2 are the protected and unprotected group and d_1 and d_2 are the classification of the system.

- **Fairness through awareness**: This definition is based on the principle that similar individuals should have similar classifications. The similarity is calculated via a distance metrics. For instance, a possible distance metric k could define the distance between two applicants i and j to be 0 if the unprotected attributes are the same and 1 if some attributes in X are different.

Causal Reasoning

In the paper a third group of definitions is described, based on the concept of causal reasoning [14]. Causal reasoning definitions assume a given causal graph, namely a directed, acyclic graph with nodes representing attributes of an applicant and edges representing relationships between the attributes. Given this graph the relations between attributes and their influence on outcome are captured by a set of structural equations which are further used to provide methods to estimate effects of sensitive attributes and build algorithms that ensure a tolerable level of discrimination due to these attributes. An example of this type of definition is:

- **Counterfactual fairness**: According to this definition, a binary machine learner is fair if its predictions do not causally depend on any protected attributes.

CHAPTER 3

A Systematic Literature Review on Data Smells

Since we previously introduced the main concept of data quality and non-functional requirements in ml-based systems, in this chapter we will carry on an analysis of the state of the art and the related works in the literature on data quality and, in particular, Data Smells in order to understand how many and which types of data smells have been defined and the impact that they can have on a machine learning-based system and its NFRs and, finally, which are the tools and datasets reported in the state of the art for this scope.

To address this objectives we carried on a systematic literature review on the topics of data quality and data smells. The systematic literature review has been used as a research methodology in this study as it is a defined and methodical way of identifying, assessing, and analyzing published literature in order to investigate a specific research question or a phenomenon of interest [15].

We used the Goal Question Metric approach [16] in order to elicit the research questions of our study and then we followed the guidelines proposed by Kitchenham and Charters [15], and then we used the snowballing methodology defined by Wohlin [17], in order to adopt a systematic inclusion of references.

Using GQM we identified our Purpose, Issue, Object, Viewpoint defining a main goal as:

◎ Our Goal.
Purpose: Understand
Issue: the progress on the topics of
Object: data smells, their presence in the public datasets and their impact on functional and/or non-functional requirements
Viewpoint: from both the points of view of researchers and developers.

From the goal we defined three main research questions. First of all, since there is a lack of a standard definition of the data smells defined in the literature we wanted to elicit a complete and unified catalog of the main data smells defined in the literature with a name and a description of its features. In particular, we asked:

🔍 RQ$_1$. *How many and which are the data smells defined in the literature?*

Then, we wanted to find out in which ways defects in the dataset used for the training of a machine learning model can impact on its functional behaviour and on its non-functional requirements, this led to the formulation of the following research question:

🔍 RQ$_2$. *What is the impact of data smells on functional and non-functional properties of ml-intensive systems?*

Finally, we wanted to analyze the most popular datasets that have been studied in the state of the art when assessing the presence of data smells, and the main tools that have been defined for data quality assessment and data smell identification, this was driven by the third research question:

🔍 RQ$_3$. *Which datasets and tools have been defined in literature?*

3.1 Research Method

3.1.1 Research Queries Definition

The first step in our systematic literature review is identifying the most appropriate search terms that may help retrieve the right set of sources and defining the right research queries. We have done this following three steps:

1. We identified the key terms for the queries extracting them from the research questions;

2. For all the terms we found alternative spellings and/or synonyms;

3. We used boolean operators for conjunction, in particular, the OR operator for the union of alternative spellings and synonyms and the AND operator for the concatenation of the key terms.

The main search terms extracted from the research questions are: Data Smells, Definition, Functional Requirements, Non-Functional Requirements, ML System, Dataset, Tool;

Leading to a set of four queries:

1. (("data smells" OR "data smell" OR "data defects" OR "data defect") AND ("definition" OR "definitions" OR "catalog"));

2. (("data smells" OR "data smell" OR "data defects" OR "data defect") AND ("functional requirements" OR "non-functional requirements" OR "requirements") AND ("machine learning system" OR "machine learning systems");

3. (("data quality") AND ("data smells" OR "data smell" OR "data defects" OR "data defect") AND ("tool", "tools"));

4. (("data smells" OR "data smell" OR "data defects" OR "data defect") AND ("dataset" OR "datasets"));

Then all the queries have been put together in order to avoid duplicated sources, leading to the final research query:

((("data smells" OR "data smell" OR "data defects" OR "data defect") AND ("definition" OR "definitions" OR "catalog")) OR (("data smells" OR "data smell" OR "data defects" OR "data defect") AND ("functional requirements" OR "non-functional requirements" OR "requirements") AND ("machine learning system" OR "machine learning systems" OR "machine learning")) OR (("data quality") AND ("data smells" OR "data smell" OR "data defects" OR "data defect") AND ("tool", "tools")) OR (("data smells" OR "data smell" OR "data defects" OR "data defect") AND ("dataset" OR "datasets")))

It can then be refactored in:

("data smell" OR "data defect") AND (("data quality" AND "tool") OR ("dataset") OR ("definition" OR "catalog") OR (("functional requirement" OR "non-functional requirement" OR "requirement") AND ("machine learning system" OR "machine learning")))

3.1.2 Search Database Selection

The second step of our systematic literature review was the selection of the database, we selected *Scopus*[1], we choose it because as said in the description of the platform, it has the comprehensive scientific data and literature, and analytical tools to keep the user up to date with the latest research trends and it can be useful in our case because data smells are a relatively new topic of research. Furthermore Scopus provides user-friendly analytical tools to filter documents and perform data analysis and visualization. Finally, since data smells could also affect non-functional properties of a system, the use of a interdisciplinary search database could be helpful in identifying articles coming from different research communities.

3.1.3 Inclusion & Exclusion Criteria

Study selection criteria are intended to identify those primary studies that provide direct evidence about the research questions. Inclusion and exclusion criteria should be based on the research questions and they should be piloted to ensure that they can be reliably interpreted and that they classify studies correctly [15]. Based on this, we defined some inclusion and exclusion criteria in order to have a more specific set of documents to work with. First we filtered out resources based on the following exclusion criteria:

- **EC1** - Articles that were not written in English.
- **EC2** - Restricted license papers, so papers whose full text read was not available for free.
- **EC4** - Duplicated documents, mainly used in the snowballing phase.

[1]Scopus: https://www.scopus.com/

While as inclusion criteria, we included the papers that met the following criteria:

- **IC1** - Articles defining data smells.
- **IC2** - Articles reporting the impact of data smells on the functional and non-functional requirements of a machine learning system.
- **IC3** - Articles assessing datasets or tools about data quality and data smells.

3.1.4 Snowballing

The snowballing technique refers to the use of the reference list of a paper or its citations to identify additional papers that might have been missed by the search process [18]. According to the guidelines formulated by Wohlin [15], after each iteration of backward and forward snowballing we applied the inclusion and exclusion criteria defined before. The snowballing process has been iterated until a until a state of saturation (i.e., the snowballing process continued until one iteration added no new results).

3.1.5 Quality Assessment

Before the data extraction we defined a list of questions that can help assess the quality of the selected paper in order to have high quality resources and to discard the papers that did not provide enough details to be used in our study. Following we report the set questions:

Q1. Does the paper provide a definition of one or more data smells?

Q2. Does the paper provide instances of the impact of the data smell on a machine learning system?

Q3. Does the paper clearly defined how the dataset is analyzed to assess the presence of data smells?

Q4. Does the paper define one or more tools to assess the presence of data smell in a dataset?

Q5. Are the main aspects of the paper clearly defined?

The first four questions can be considered mutually exclusive. All the questions can be answered with "Yes" (Score = 1), "Partially" (Score = 0,5) or "No" (Score = 0), the final quality score for each paper was computed by summing up the score of the answers to the two questions, to be accepted the article should have at least a score of 1,5.

3.1.6 Data Extraction

Once that the final set of papers has been selected, we started the phase of data extraction in order to gather information useful to answer our research questions. To address this we have performed two main steps on each paper:

1. **"Abstract Analysis"**: First, we read the abstract of the paper, in order to find useful information about its contents.
2. **"Keywords Localization"**: Then, we looked for specific keywords such as "data smell", "requirement", "dataset" in order to allow a quick identification of relevant parts in the paper.
3. **"Full Paper Inspection"**: Finally, we analyzed the introduction, the main steps of the methodological part and the results part to find relevant information that we did not detect in the previous step.

3.1.7 Search Process Execution

After the definition of the search process, we then proceeded with its execution. In particular, the execution followed the steps below. The whole process is reported in **Figure 3.1**.

1. First of all we defined the research query, as showed in the **Section 2.1.1**, and we we run it on the selected database, Scopus, filtering for Title, Abstract and Keywords, retrieving an amount of 23 documents.

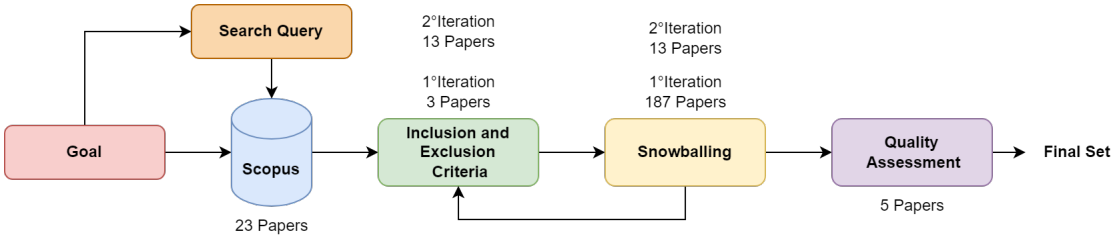

Figure 3.1: Search Process Execution Overview

2. Then, from the initial set we filtered out the irrelevant papers using the exclusion criteria removing all the papers not English and without an open access, leading to a total amount of 12 items. After this we manually selected the papers that passed the inclusion criteria checks, retrieving 3 final papers.

3. We performed a snowballing process to search for possible missing papers following what we described in **Section 2.1.4**, the backward snowballing added 179 papers to the list, from this, we performed an analysis of the title and, if needed, of the abstract in order to apply the exclusion and inclusion criteria leading to a set of 7 papers, the forward snowballing added 5 more papers, 3 after the exclusion and inclusion criteria . Finding a final set of paper of 13 items. The second iteration of the snowballing didn't add any relevant document to the list, reaching the saturation state.

4. Then we did a quality assessment of the resources following what said in **Section 2.1.5**, finding that only 5 papers passed the checklist, defining in a clear way one of the aspects interested by our study.

5. The last phase was the data extraction in order to answer the research questions, all the results are reported and discussed in **Section 2.2**.

3.2 Analysis of the Results

In the following section we will try to answer our research questions.

RQ$_1$. *Which are the data smells defined in literature?*

After the data extraction phase, we found out that there are two main taxonomies defining data smells [SLR1] [SLR2]. In order to answer the first research question we reported all the main data smells defined in the literature, under a unified catalog divided by classes of smells, containing a name and a description for each one.

The **Figure 3.2** below shows an overview of the catalog.

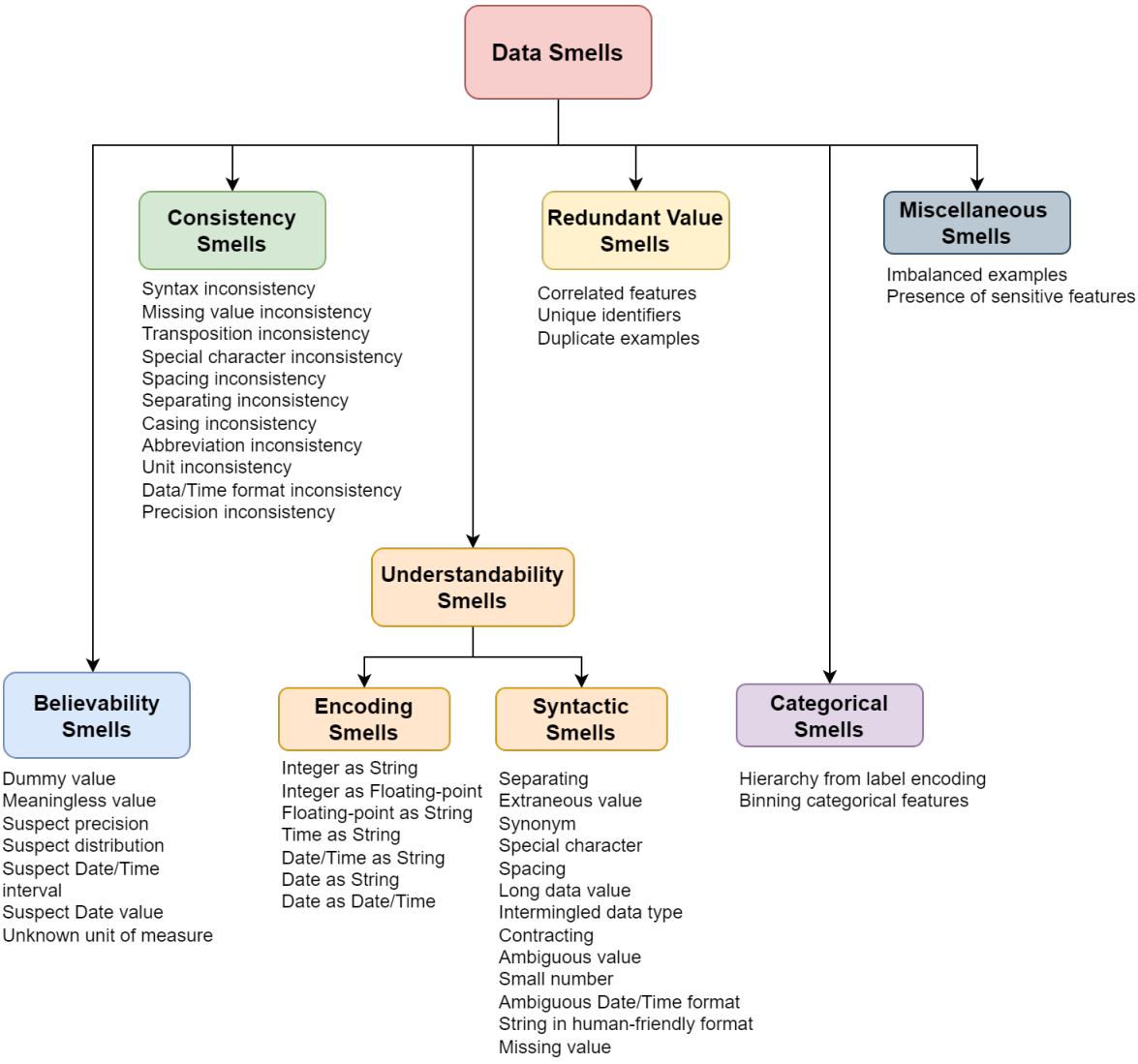

Figure 3.2: Data Smells [SLR1]

Believability Smells

Name	Description
Dummy value	Dummy Value occurs when a substitute value is used due to several reasons [SLR1].
Meaningless value	Meaningless Value is a smell that occurs when a data value has no common meaning or contains suspect repeating sequences of characters [SLR1].
Suspect precision	Suspect Precision occurs when a value has a large number of decimal places [SLR1].
Suspect distribution	Suspect Distribution arises in a situation where numerical values have a suspect distribution [SLR1].
Suspect Date/Time interval	Suspect Date/Time Interval occurs when there is a very long/short date/time interval between data instances [SLR1].
Suspect Date value	Suspect Date Value arises when a date value represents a date far in the past or the future [SLR1].
Unknown unit of measure	Unknown unit of measure occurs when due to the lack of standardised data collection procedures, long duration of data collection and use of undocumented data sources the same features can be measured with different units [SLR2].

Table 3.1: Believability Smells

Understandability Smells

Encoding Smells:

Name	Description
Integer as String	Integer as String occurs when an integer is encoded as a string [SLR1].

Integer as Floating-point Number	Integer as Floating-point Number occurs when an integer is encoded as a floating-point number [SLR1].
Floating-point Number as String	Floating-point Number as String occurs when a floating-point number is encoded as a string [SLR1].
Time as String	Time as String occurs when a time is encoded as a string [SLR1].
Date/Time as String	Date/Time as String occurs when a timestamp are encoded as a string [SLR1].
Date as String	Date as String occurs when a date is encoded as a string [SLR1].
Date as Date/Time	Date as Date/Time occurs when a date is encoded as a datetime data type [SLR1].

Table 3.2: Encoding Smells

Syntactic Smells:

Name	Description
Separating	Separating smell arises when data contains thousands separators for grouping digits, this can lead to ambiguity when decoding decimal separators [SLR1].
Extraneous value	Extraneous Value occurs when data values provide additional or unnecessary information [SLR1].
Synonym	Synonym smell occurs when different data values have the same semantic meaning, but a different syntax [SLR1].
Special character	Special Character smell is characterized by the presence in the data values of special characters like Commas, Dots, Hyphens, Apostrophes, Tab Char, Punctuation, Parentheses, Dashes, accented Letters, etc [SLR1].

Spacing	Spacing smell occurs when data values contain an uncommon pattern of spaces [SLR1].
Long data value	Long Data Value occurs when a value is too long to understand [SLR1].
Intermingled data type	Intermingled Data Type characterizes a situation in which data values contain numeric as well as alphabetic characters [SLR1].
Contracting	Contracting smell arises when values represent a shortened version of a word or a phrase [SLR1]. **Casing:** Casing Smell occurs when data values represent an unusual use of upper and lower case (Mixed Case, Upper Only, Lower Only) [SLR1].
Ambiguous value	Ambiguous Value arises when a data value is an abbreviation, homonym or acronym [SLR1].
Small number	Small number occurs when a value represent a number below 1 [SLR1].
Ambiguous Date/Time format	Ambiguous Date/Time Format arises when a date is represented in short format or a time is represented in 12-hour clock format [SLR1].
Strings in human-friendly format	This smell occur when a features can be represented as a numerical value, but a representation comprehensible to humans is used [SLR2].
Missing value	Missing Value occurs when one or more features in a dataset present NULL or NaN values leading to a loss of information in the data [SLR2].

Table 3.3: Syntactic Smells

Consistency Smells

Name	Description
Syntax inconsistency	Syntax Inconsistency is characterized by an inconsistent use of data values [SLR1].
Missing value inconsistency	Missing Value Inconsistency is a smell that arises when missing values are not represented consistently by a constant [SLR1].
Transposition inconsistency	Transposition Inconsistency arises when the ordering of words or special characters is not used consistently [SLR1].
Special character inconsistency	Transposition Inconsistency occurs when special characters are not used consistently [SLR1].
Spacing inconsistency	Spacing Inconsistency occurs when spacing in not used consistently [SLR1].
Separating inconsistency	Separating Inconsistency occurs when the separator of thousands are not used consistently [SLR1].
Casing inconsistency	Casing Inconsistency characterizes a situation in which upper and lower case is not used consistently [SLR1].
Abbreviation inconsistency	Abbreviation Inconsistency occurs when abbreviations, acronyms or contractions are not used consistently [SLR1].
Unit inconsistency	Unit Inconsistency arises when units of measurement are not used consistently [SLR1].
Date/Time format inconsistency	Date/Time Format Inconsistency occurs when date or time formats are not used consistently [SLR1].
Precision inconsistency	Precision Inconsistency arises when the number of decimals is not used consistently [SLR1].

Table 3.4: Consistency Smells

Redundant Value Smells

Name	Description
Correlated features	Correlated Features occur when two features have a linear relationships between them, this can be a symptom for redundant information [SLR2].
Unique identifiers	Unique Identifiers represent a column of the dataset containing uid values, usually used for SQL operations, they can be redundant when used for the training of a machine learning model [SLR2].
Duplicate examples	Duplicate examples in a dataset are defined as two or more rows which refer to the same entity. They do not serve any purpose and can be removed from the dataset, making their presence a smell for redundancy [SLR2].

Table 3.5: Redundant Value Smells

Categorical Smells:

Name	Description
Hierarchy from label encoding	This smell can occur when using a label encoding for categorical features, encoding a sensitive feature may incorrectly associate a sex or race with a higher numerical value to be superior to other values with a lower number [SLR2].
Binning categorical features	This smell can occur when using one-hot encoding a feature with high cardinality, it can result in a very large feature space and incur higher memory and computational costs [SLR2].

Table 3.6: Categorical Smells

Miscellaneous Smells:

Name	Description
Imbalanced examples	This smell occur when a dataset has an unbalanced distribution of a certain class leading a machine learning model make unfair predictions [SLR2].
Presence of sensitive features	This smell arises when the presence of high-impact features may lead to a biased and unfair prediction of a machine learning model [SLR2].

Table 3.7: Miscellaneous Smells

There are also two more paper that add some other, so called, data anomalies to the previous ones [SLR3] [SLR4], the majority of the anomalies reported in these papers can be mapped to one of the smells defined above, while the others are related to the definition of the database schema instead of the dataset itself.

The anomalies reported by Sukhobok et al. [SLR4] are divided into four categories:

Cell Values:

Name	Description
Illegal values	Values outside of domain range.
Inconsistent values	Syntactically correct but contradicting with other attribute values.
Missing values	Column values are not present.

Table 3.8: Cell Values Anomalies

Column headers:

Name	Description
Column headers containing attribute values	Column headers are attribute values themselves, not attribute names.
Incorrect column headers	Column headers are inconsistent with the attribute they hold.

Table 3.9: Column headers Anomalies

Column headers and cell values:

Name	Description
Columns not related to data model	Column headers are inconsistent with the attribute they hold.
Multiple values stored in one column	Several attribute values of the data model are stored in one column.
Single value split across multiple columns	One attribute value of the data model stored across several columns.

Table 3.10: Column headers and cell values Anomalies

Rows:

Name	Description
Rows not related to data model	Records in the source dataset describe unrelated entities.
Duplicate rows	The same entity (having the same primary key according to the data model) is described more than once in the dataset.

Table 3.11: Rows Anomalies

While the anomalies reported by Ralph Foorthuis [SLR3] are based on two dimensions of the data, the type of the data - Continuous, Categorical and Mixed - and the cardinality of the relationship - Univariate, Multivariate - and six type of anomalies are defined **Extreme value anomaly**, **Rare class anomaly**, **Simple mixed data anomaly**, **Multidimensional numerical anomaly**, **Multidimensional rare class anomaly**, **Multidimensional mixed data anomaly**.

👉 **Answer to RQ_1.** To summarize the results of the first research question we defined seven classes of data smell, namely, Believability Smells, Encoding Smells, Syntactic Smells, Consistency Smells, Redundant Value Smells, Categorical Smells and Fairness Smells, with a final catalog of 46 data smells. 16 more data anomalies can be found in the literature, but they can be mapped to one of the smells defined previously, while others are related to the definition of the database schema instead of the dataset itself.

RQ$_2$. *What is the impact of data smells on functional and non-functional properties of ml-intensive systems?*

Nowadays is well known that a lack of quality in the data can lead to several problems in data-driven systems like machine learning-based systems. Unfortunately detecting data smells in not a trivial challenge and requires knowledge about their characteristics, the main problem is that poor quality will probably lead to unpredictable consequences such as financial loss, or even human loss [SLR5]. As showed by Foidl et al. [SLR5], real world systems add smells in the data at different points in the pipeline. They analyzed a real world scenario with the case of study of business travel data since travel data have a large potential for sustainable solutions testing and application regarding carbon footprint calculations, management, and reduction, which is highly relevant for any industry nowadays.

To address the second research question we analyzed different papers about causes and consequences of data smells. Felderer et al. [SLR1] identified as the most common causes of poor data quality the following elements:

- **Data management**. In this concept fall bad practices in data collection or preparation that can cause the emergence of data smells like inconsistent data collection or transformation processes. Then, a bad documentation or the poor communication between the different actors in the data life-cycle can introduce issues in the data. For instance, incomplete metadata can lead to incorrect assumptions about the data by software engineers, resulting in incorrect implemented data processing logic, which causes smelly data.

- **Data handling**. Poor data handling practices will likely cause data smells. For example, not being explicit when converting a date string (e.g., "2021-01-01") into a datetime object (e.g., pandas.to_datetime) causes the Date as Date/Time smell (i.e., "2021-01-01 00:00:00") to arise. As the conversion input represents just a date, developers may be unaware that without further declaration, a time suffix (i.e., "00:00:00") is added. By sequencing multiple data operations (i.e., method chaining), developers cannot see the intermediate processing results and thus identify problems introduced in the data.

- **Data source quality**. Obviously data smells can further arise through a poor intrinsic quality of the data sources from which the data originate. For example, a column name in a relational database is used in different tables but with different data types. Accordingly, this can cause data encoding smells when developers retrieve the data as they are not expecting different data types.

The main consequences are **defects and failures**, for instance data smells can lead to incorrect knowledge generation in an AI-based system. As modern AI-based systems often pursue a continuous learning strategy, they typically use serving data as training data. Thus, a well-trained ML model can degrade over time based on smelly serving data continuously used to update the model. Moreover, the output of one AI-based system is often directly consumed by other systems or even influences its own training data creating hidden or direct feedback loops [SLR1]. In such a scenario, hidden or latent data issues can cause severe problems over time, causing a gradual regression of the **performance** of the ML models involved [SLR1], for instance we can think of a financial system that can suggest financial shares to buy, influencing the market and then influencing its own behaviour.

More consequences can occur in development and **maintainability**, in fact data smells often require additional data cleaning or preprocessing pipelines in AI-based systems consequently, the occurrence of data smells and their corresponding treatment makes AI-based systems harder to understand and maintain and thus increases the risk of introducing further errors [SLR1].

Data smells can then have consequences on the **safety**, espcially when talking of cyber-phisical systems interacting with the real world [SLR1], for instance, self driving cars.

👉 **Answer to RQ_2.** To summarize the results of the second research question, in the literature, there are different examples of how a lack of data quality can influence non-functional requirements, such as defect proneness, safety, and maintainability. Since the data are defined in the initial steps of the pipeline, the presence of data smells could raise a degradation of the model in combination with other issues related to technical debt specific to AI-based systems (i.e., Pipeline Jungles and Hidden Feedback Loops).

RQ$_3$. *Which datasets and tools have been defined in literature?*

During the literature review we found different tools about data quality, and more in particular about data smell detection. Data smell detection is closely related to the field of data validation [SLR1], the main problem is that smelly data cannot always be mapped to data errors, so they could not be suitable to be detected by validation tools like Amazon's Deequ[2], or Google's TensorFlow Data Validation[3].

In order to address this problem Harald Foidl et al. [SLR1] developed two tools that are able to detect most of the data smell that they defined. First of all they introduced two main metrics:

- *Data Smell Strength*. With this metric they indicate the likelihood that a data value or pattern is treated as suspicious, and a smell is raised. Basically, this metric implies the concrete thresholds and/or hyper-parameters of the individual detection methods. For instance, the number of contiguous characters required to detect a Long Data Value smell [SLR1].

- *Data Smell Density*. This metric describes the relative number of detected smells of a data attribute. This metric can be used to focus on data attributes with a high density of smells. For example, a data attribute can only be considered smelly if at least 10 percent of its data instances represent a data smell [SLR1].

Then they presented two tools, the first one is **Data Smell Detection**[4], it is a rule-based detector, it have some rules defined by the developers and other rules based on the open source data validation tool Great Expectations[5]. By adjusting the expectations provided by the tool it is able to detect nine data smells, namely: **Long Data Value**, **Casing**, **Duplicated Value**, **Extreme Value, Missing Value**, **Suspect Sign**, **Integer as String**, **Floating Point Number as String** and **Integer as Floating Point Number**.

[2]https://www.amazon.science/publications/deequ-data-quality-validation-for-machine-learning-pipelines

[3]https://www.tensorflow.org/tfx/guide/tfdv

[4]Data Smell Detection: https://github.com/mkerschbaumer/rb-data-smell-detection

[5]Great Expectations: https://github.com/great-expectations/great_expectations

The second tool is **ML Data Smell Detection**[6], it is based on several machine learning algorithms trained for general applicability in order to detect different type of data smells, for instance it uses Word2Vec algorithms to synonym smell detection or neural networks to detect smells such as Ambiguous Date/Time Format or Date/Time Format Inconsistency.

While to address the part of the question about the datasets studied in the literature, there is one main article assessing the presence of data smells in 25 public datasets. Arumoy Shome et al. [SLR2] started picking 25 datasets from Kaggle[7] based on the Most Votes criteria and applying some exclusion criteria such as excluding Unstructured dataset like images, videos or text corpus. The complete list of datasets is reported in Table 3.12.

Name	Description
abalone	Predicting age of abalone from physical measurements.
adult	Predicting whether income exceeds $50K/yr based on census data.
airbnb	Airbnb listings and metrics in NYC, NY, USA.
avocado	Historical data on avocado prices and sales volumn in Multiple US markets.
bitcoin	Bitcoin data at 1-min intervals from select exchanges, Jan 2012 to March 2021.
breast-cancer	Predict whether the cancer is benign or malignant.
comic-dc	FiveThirtyEight DC comic characters.
comic-marvel	FiveThirtyEight Marvel comic characters.
covid-vaccine	Daily and total vaccination for COVID-19.
covid-vaccine manufacturer	Vaccinations for COVID-19 by manufacturer.
earthquake	Date, time and location of all earthquakes with magnitude of 5.5 or higher.

[6]https://github.com/georg-wenzel/ml-data-smell-detection
[7]Kaggle: https://www.kaggle.com/

fraud	Anonymised credit card transactions labeled as fraudulent or genuine.
happiness	Happiness scored according to economic production, social support, etc.
heart insurance	Insurance forecast by using linear regression.
iris	Classify iris plants into three species
netflix	Listings of movies and tv shows on Netflix
permit	San Francisco building permits
playstore	Google play store apps data
student	Marks secured by students in various subjects.
suicide	Suicide rates overview 1985 to 2016.
telco	Telco customer churn.
vgsales	Video game sales.
wine	Red wine quality.
youtube	Trending YouTube video statistics.

Table 3.12: Selected Datasets

From this study the Redundant Value Smells and the Categorical Value Smells are the most common categories with an occurrence of 33 instances and 17 instances respectively. While the Missing Value Smells and the String Value Smells are the least common categories with a total occurrence of 13 and 12 respectively [SLR2]. The *correlated features* smell is the most common smell being present in 19 dataset out of 25, the other of the top five are Hierarchy from label encoding smell, Missing value smell, Unique identifiers smell and and Unknown unit of measure smell.

While the least observed smells are Imbalanced examples smell, Strings in human-friendly formats smell and Presence of sensitive features smell which are observed in less than 5 datasets.

👉 **Answer to RQ$_3$.** To summarize the results of the third research question we found that smelly data cannot always be mapped to data errors, so they could not be suitable to be detected by validation tools. This led Harald Foidl et al. [SLR1] to the development of two tools for data smell detection, based on two research strategies, namely rule based detection and machine learning-based detection. Then, the study of 25 datasets carried out from the study of Arumoy Shome et al. [SLR2] showed redundant value smells and the categorical value smells are the most common categories of smell in the public available datasets.

3.2.1 Impact on the State of the Art

In the previous chapter we analyzed what a machine learning system is, which are the main non-functional requirements that distinguish this type of systems from classical software systems and we presented an overview on the data quality and its metrics and the problems that can be causes when this aspect is not taken into account during the development of an AI system. As shown in the systematic literature review, data quality aspects are a crucial factor for machine learning-based systems as they can impact its functional behaviour and some non-functional requirements like defect proneness, maintainability and safety. In such a scenario we want to understand to what extent the presence of data smells in a dataset could impact the metrics of data quality like Completeness, Uniqueness, Fairness and so on.

CHAPTER 4

On The Impact of Data Smells on Data Quality

In this chapter we will report all the process that led this work from the formulation of the research questions to the results. We will discuss the phases of data collection and data analysis.

4.1 Research Method

After the main goal defined in the previous chapte, we formulated two more research questions to led the research process. First of all we wanted to conduct a quantitative analysis in order to know how many and how much famous public datasets are affected by data smells, this led to the formulation of the fourth research question:

RQ$_4$. *What is the prevalence of data smells?*

After the quantitative assessment, we wanted to know if the presence of certain data smells could impact the metrics of data quality , and if applying one or more refactoring strategies on the detected data smells can mitigate the potential issues, so we formulated the last research question:

RQ$_5$. *To what extent data smells impact data quality?*

Once we defined the RQs that will led this work we set up our research methodology, following this main methodological steps:

1. First of all we collected our datasets, based on the studies of Tai Le Quy et al. [19] and Hirzel et al.[20], keeping all the datasets used for classification tasks;
2. Then we extracted all the main information from the datasets that we will use to compute our metrics, like sensitive features or number of non-empty values;
3. As third step we built our tool implementing three modules, the data smell detection one, the one implementing the refactoring strategies, and the one which compute the data quality metrics;
4. Then, we performed a statistical analysis to understand the impact that the presence of three types of data smells have on the quality metrics;
5. Finally we applied different refactoring strategies to the data smells in order to see if there is an actual improvement in the metrics.

The process is summarized in the figure 4.1:

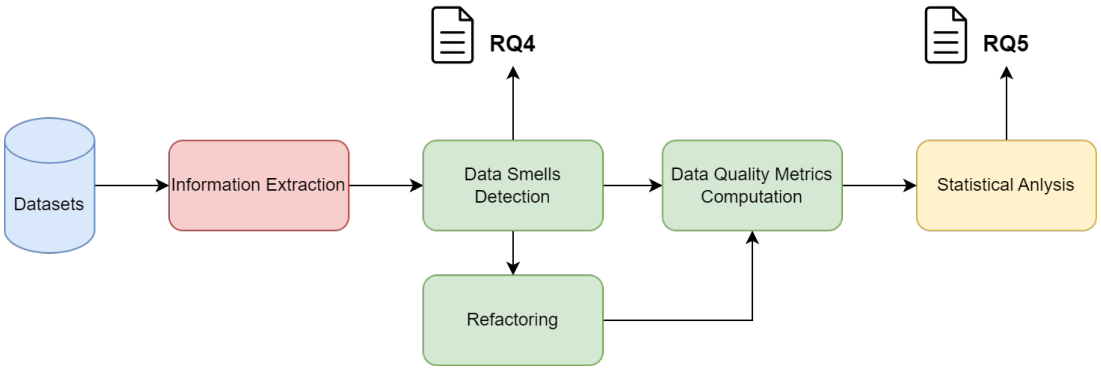

Figure 4.1: Research Method

4.1.1 Data Collection

The first step of this phase was the data collection, we wanted to collect a set of datasets studied in the literature in the scope of classification tasks and machine learning fairness to have a set of information useful to the computation of some quality metrics. In order to address this problem we based our research on the data studied by Tai Le Quy et al. [19], in their paper they overview a bunch of real world, tabular, datasets used for fairness-aware machine learning and they analyzed the correlation between the different attributes, particularly protected attributes, and class attribute, using a Bayesian network [19]. And to enhance this initial set we added the datasets reported by Hirzel et al. [20].

The datasets are divided in terms of application domain and we have five different categories, namely: financial datasets, criminological datasets, healthcare and social datasets, educational datasets and miscellaneous datasets, from all the datasets reported in the papers we extracted those related to classification tasks, for a total amount of 19 datasets.

For each dataset we provided a description and a set of metadata including name, path, protected attribute, privileged classes and favorable labels, in order to use them in the data analysis phase.

The selected dataset are reported below:

Financial Datasets

Name	Description
Adult	The adult dataset is one of the most popular dataset for fairness classification studies, the task associated with this dataset is to decide whether the annual income of a person exceeds 50,000 US dollars based on demographic characteristics. The **protected attributes** for this dataset are sex = {male, female} with the dataset dominated by male instances; race = {white, black, asian-pac-islander, amer-indian-eskimo, other} with the dataset dominated by white people instances; age = [17-90] with the dataset dominated by the [25-60] years old group. Finally the **class attribute** is income {<= 50K, > 50K} where the positive class is "> 50K"[19].
KDD Census-Income	The KDD Census-Income dataset was collected from Current Population Surveys implemented by the U.S. Census Bureau from 1994 to 1995, it is an enhancement of the adult dataset. The prediction task is to decide if a person receives more than 50,000 US dollars annually or not. The **protected attributes** for this dataset are sex = {male, female} with the dataset slightly imbalanced towards female instances; race = {white, non-white} with the dataset dominated by white people instances. The **class attribute** is income {<= 50K, > 50K} where the positive class is "> 50K"[19].

German Credit	The German credit dataset consists of samples of bank account holders. The dataset is used for risk assessment prediction to determine whether it is risky to grant credit to a person or not. The **protected attributes** for this dataset are sex = {male, female} with the dataset dominated by male instances; age = {<=25, >25} with the dataset dominated by people older than 25 years. The **class attribute** is class-label = {good, bad} revealing the customer's level of risk. The positive class is "good"[19].
Dutch Census	The Dutch census dataset represent aggregated groups of people in the Netherlands for the year 2001. Researchers have used Dutch dataset to formulate a binary classification task to predict a person's occupation which can be categorized as high-level or low-level profession. The **protected attribute** for this dataset is sex = {male, female} with a balanced distribution of instances. The **class attribute** is occupation = {0, 1} demonstrating if an individual has a prestigious profession or not. The positive class is 1 (high-level)[19].

Bank Marketing	The bank marketing dataset is related to the direct marketing campaigns of a Portuguese banking institution from 2008 to 2013. The classification goal is to predict whether a client will make a deposit subscription or not. The **protected attributes** for this dataset are age = {25-60, <25 or >60}, the dataset is dominated by people from 25 to 60 years old; marital = {married, non-married} married group is the majority class. The class attribute is y = {Yes, No} presenting whether a customer will subscribe a term deposit or not. The positive class is "Yes"[19].
Credit Card Clients	The credit card clients dataset investigated the customers' default payments in Taiwan in October 2005. The goal is to predict whether a customer will face the default situation in the next month or not. The **protected attributes** for this dataset are sex = {male, female} with the dataset dominated by female instances; marriage = {married, single, others} with single group as the majority class; education = {graduate school, university, high school, others} with university as majority class. The class attribute is default payment = {0, 1} indicating whether a customer will suffer the default payment situation in the next month (1) or not (0). The positive class is 1[19].
Credit G	This is an alternative version of the German Credit Dataset and classifies people described by a set of attributes as good or bad credit risks. The **protected attributes** for this dataset are the age and the personal status [20].

Table 4.1: Financial Datasets

Criminological datasets

Name	Description
COMPAS	The COMPAS dataset was released in 2016 based on the Broward County data. Risk of recidivism (denoted as COMPAS recid.) and Risk of violent recidivism (denoted as COMPAS viol. recid) subsets are the most widely used in the literature. The former has a classification task to predict if an individual is rearrested within two years after the first arrest, while the latter predicts if an individual is rearrested for a violent crime within two years. The **protected attributes** for these two datasets is race, in both subsets, black and white are the main races with black as majority class. The **class attribute** is two year recid = {0, 1} indicating whether an individual will be rearrested within two years (1) or not (0). The positive class is 1[19].

Table 4.2: Criminological Datasets

Healthcare and Social datasets

Name	Description
Diabetes	The diabetes dataset describes the clinical care at 130 US hospitals and integrated delivery networks from 1999 to 2008. The classification task is to predict whether a patient will readmit within 30 days. Typically Gender = {male, female} is chosen as the **protected attribute** with female as majority class. The **class attribute** is readmitted = {< 30, > 30} indicating whether a patient will readmit within 30 days. The positive class is "< 30"[19].

Ricci	The Ricci dataset was generated by the Ricci v.DeStefano case, in which they investigated the results of a promotion exam within a fire department in Nov 2003 and Dec 2003. Although it is a relatively small dataset, it has been employed for fairness-aware classification tasks in many studies. The classification task is to predict whether an individual obtains a promotion based on the exam results. The **protected attribute** for this dataset is race which contains three values (black, white, and hispanic) with white as majority class. The **class attribute** is promoted = {True, False} revealing whether an individual achieves a promotion or not. The positive class is "True"[19].
Heart disease	This dataset is a subset of feature of a medical dataset of patients with heart disease. The **class attribute** is the prediction of a disease, while the **protected attribute** is the age of the patient [20].
Nursery	Nursery Database was derived from a hierarchical decision model originally developed to rank applications for nursery schools. It was used during several years in 1980's when there was excessive enrollment to these schools in Ljubljana, Slovenia, and the rejected applications frequently needed an objective explanation. The **class attribute** is the final evaluation in terms of priority. The **protected attribute** is the Parents feature [20].

Table 4.3: Healthcare and Social Datasets

Educational datasets

Name	Description
Student Performance	The student performance dataset described students' achievement in the secondary education of two Portuguese schools in 2005 - 2006 with two distinct subjects: Mathematics and Portuguese. The **protected attribute** for this dataset are sex = {male, female} with the dataset is dominated by female students; age = {<18, >= 18} with young students (less than 18 years old) as the majority. The **class attribute** is G3 (final score) = {low, high}, The positive class is "high"[19].
Law School	The Law school dataset was conducted by a Law School Admission Council (LSAC) survey across 163 law schools in the United States in 1991. The dataset contains the law school admission records. The prediction task is to predict whether a candidate would pass the bar exam. The **protected attribute** for this dataset are male = 1, 0 whit 1 as majority group; race = white, black, Hispanic, Asian, other with white dominating the distribution. The **class attribute** is pass bar = {0, 1}, the positive class is 1 (pass)[19].

Teaching Assistant Evaluation	The data consist of evaluations of teaching performance over three regular semesters and two summer semesters of 151 teaching assistant (TA) assignments at the Statistics Department of the University of Wisconsin-Madison. The **class attribute** is the score and it is divided into 3 roughly equal-sized categories ("low", "medium", and "high") to form the class variable. The **protected attribute** is the feature telling if the assistant is or not an english speaker [20].

Table 4.4: Educational Datasets

Miscellaneous datasets

Name	Description
Speed Dating	This data was gathered from participants in experimental speed dating events from 2002-2004. During the events, the attendees would have a four-minute "first date" with every other participant of the opposite sex. At the end of their four minutes, participants were asked if they would like to see their date again. The **class attribute** is if there's a match or not, and the **protected attributes** are "same race" and "importance same race" [20].
Titanic	This dataset is a report of the survivors of the titanic, its **protected attribute** is the sex of the passenger and the **class attribute** is if the passenger survived or not [20].

Table 4.5: Miscellaneous Datasets

4.1.2 Data Analysis

The second step of this phase was the analysis of the collected datasets, to carry out this phase we built a tool wrapping the data quality tool DSD[1] and implementing a module for the computation of metrics about data quality.

Data Smell Detection

The first tool on which we worked is Data Smell Detection (DSD) the rule-based data smell detector validated in the literature. The tool given a dataset in input allows the user to run a static analysis on it to detect the presence of data smells, based on the selected configuration of the hyper-parameters.

In order to add the tool in our pipeline, first of all, we forked the repository, in order to make some changes in the configuration of the project aiming at automating the process of detection, the repository that we used for this work can be found on GitHub[2]. We made three main changes to address our goal:

1. First, we forced the configuration of the hyper-parameters on the strict mode, in order to have more accurate results in the detection;

2. In second place, we deleted the system of cross site request forgery tokens, that didn't allow the automation of the web service from the script;

3. Finally, we modified the front-end page adding tags to the main HTML elements in order to ease the parsing of the results using Python.

After the changes we used the application as Docker container in order to ease the setup as suggested by the developers.

With the final configuration the tool is able to receive a dataset, to configure the detection on the strict mode and to detect nine different data smells, namely, **Long Data Value, Casing, Duplicated Value, Extreme Value, Missing Value, Suspect Sign, Integer as String, Floating Point Number as String and Integer as Floating Point Number**.

[1] https://github.com/mkerschbaumer/rb-data-smell-detection
[2] https://github.com/DinoDx/rb-data-smell-detection

Data Quality Metrics

This module implements the classes to compute the quality metrics of our datasets. We decided to rely on the metrics described by W.Elouaraoui et al.[21], since they introduced a set of metrics to assess the quality in the context of big data processes, we decided to take into account only a subset of metrics, excluding all the time-related metrics and the process-related metrics.

The final set includes the following metrics:

- **Completeness**: In big data environments, the collected raw data are usually incomplete and lack contextual information. Thus, data completeness is one of the most crucial criteria when assessing data quality [21]. It can be defined as:
  $\frac{Number\_of\_non\_empty\_values}{Total\_values} \times 100$
- **Uniqueness**: Large-scale datasets are usually redundant since the data are gathered from multiple sources; therefore, the same information can be recorded more than once in a different format [21]. It can be defined as:
  $\frac{Number\_of\_unique\_rows}{Total\_rows} \times 100$
- **Consistency**: Consistent data could be defined as data presented in the same structure, types and coherent with data schemas and standards [21]. It can be defined as:
  $\frac{Number\_of\_values\_with\_consistent\_types}{Total\_values} \times 100$
- **Readability**: Data validity is not limited to data format but refers to data semantics as well. Indeed, raw data may contain misspelled words or even nonsense words, especially when the database is overwhelmed by human data entries [21]. It can be defined as:
  $\frac{Number\_of\_non\_misspelled\_values}{Total\_values} \times 100$

After the selection of the metrics, we used AI Fairness 360, the main objectives of this toolkit are to help facilitate the transition of fairness research algorithms to use in an industrial setting and to provide a common framework for fairness researchers to share and evaluate algorithms[3]. The package includes a comprehensive set of fairness

[3]https://github.com/Trusted-AI/AIF360

metrics for datasets and models, explanations for these metrics, and algorithms to mitigate bias in datasets and models. The tool has three main classes, two out of this three will be useful for this study:

- The first one is the **Dataset Class**: this class and its sub-classes are a key abstraction that handle all forms of data. To maintain a common format, independent of what algorithm or metric is being applied, the structure of the Dataset class is implemented so that all of these relevant attributes (features, labels, protected attributes, and their respective identifiers) are present and accessible from each instance of the class. Sub-classes add further attributes that differentiate the dataset and dictate with which algorithms and metrics it is able to interact.

- The second one is the **Metrics Class**: The Metric class and its sub-classes compute various individual and group fairness metrics to check for bias in datasets and models. In particular The *DatasetMetric* class and its subclass *BinaryLabelDatasetMetric* can assess a single dataset as input in order to find metrics like the group fairness measures of disparate (DI), the statistical parity difference (SPD) and the consistency. The first one can be used to calculate metrics about the *predictive parity* of the dataset [14]., the second one can be mapped to the *statistical parity* definition of fairness [14] and the third one can be seen as a measure for *fairness through awareness* based only on the dataset [14].

The tool can compute fairness metrics based on a dataset given as input. In order to calculate the metrics the tool takes in input a list of meta-data:

1. The path of the dataset,
2. The label of the independent variable,
3. The favorable class,
4. The name of the protected attribute,
5. The privileged class,
6. The features to drop.

In order to enable the tool to work we filled the missing values if present using one of the refactoring strategies that we will discuss later in the chapter and finally we factorize the data values in order to have only numerical attribute on which the tool can work.

Once the dataset is ready the tool compute three fairness metrics, **Disparate Impact, Statistical Parity Difference and Fairness Consistency.**

The metrics are defined as follow:

- **Disparate Impact**: $\frac{P(Y=1|D=unprivileged)}{Pr(Y=1|D=privileged)}$
- **Statistical Parity Difference**: $P(Y = 1|D = unprivileged) - P(Y = 1|D = privileged)$
- **Fairness Consistency**: $1-\frac{1}{n}\sum y_i - \frac{1}{n\_neighbors}\sum y_j$

Refactoring Strategies

The last step to set up the experiment was the definition of a refactoring strategy for each one of the data smell selected for our research.

Since the extreme value smell is characterized by a value which is an outlier for the distribution of values we used, as first method, the **feature clipping normalization**, which caps all feature values above (or below) a certain value to a fixed value.

For the second smell, since in the literature there are plenty of strategies to deal with missing values, we used two of the most used methods, namely, **mean value imputation** which set a missing value to the mean value of the distribution.

Finally, for the suspect sign smell we used the **feature clipping normalization** as strategy, we selected the same strategy for the first and the last data smells since their definition is similar.

The following Table 4.6 summarizes the choices made to deal with each data smell:

Data Smell	Refactoring Strategy
Extreme Value Smell	Feature Clipping
Missing Value Smell	Mean Value
Suspect Sign Smell	Feature Clipping

Table 4.6: Refactoring Strategies

Tool

Once we defined all the main component for our tool, we defined its architecture. First of all starting from the datasets that we selected, we created two files, the first paths.csv is the input file for the data smell detection module, it contains, for each dataset, the name and the relative path, while the second one meta_dataset.csv is the input file for the quality metrics module, it contains, for each dataset, the main information that will be used by AI Fairness 360, including sensitive features, privileged classes and favorable labels.

Then we have the 3 implemented modules, the first one is DSD, it is the wrapper to call our version of the detection tool it works using HTTP requests to the tool and then analyze the resultant HTML page collecting all the information about the smell, giving in output the complete list of data smell, afterwards the list is given in input to the refactoring module that will apply the selected strategies to the data smell. The last module is the quality metrics module that will compute the quality and fairness metrics before and after the application of the refactoring strategies.

The tool architecture is shown in figure 4.2:

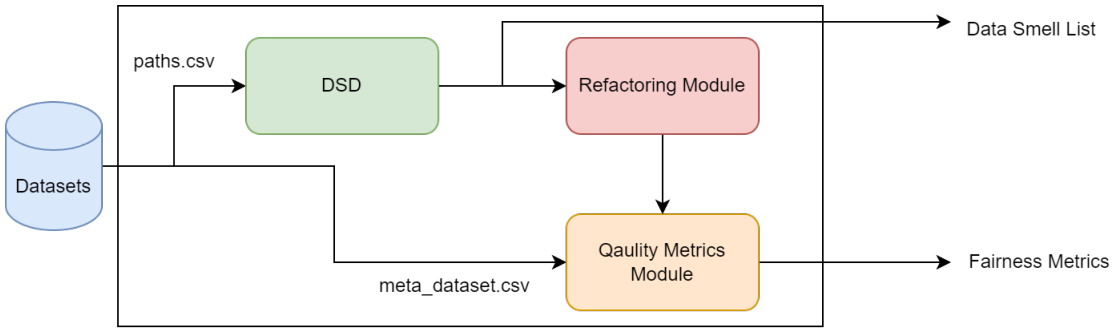

Figure 4.2: Tool Architecture

4.1.3 Working Hypothesis

To address the RQ_5, we will lead a statistical analysis based on the hypothesis testing. First of all null-hypotheses were formulated in order to test the effect certain data smells may have on the data quality metrics:

- $H0_1$: There is no difference between the presence and the absence of the Extreme Value, Missing Value and Suspect Sign smells on the Completeness metric in a dataset.
- $H0_2$: There is no difference between the presence and the absence of the Extreme Value, Missing Value and Suspect Sign smells on the Uniqueness metric in a dataset.
- $H0_3$: There is no difference between the presence and the absence of the Extreme Value, Missing Value and Suspect Sign smells on the Consistency metric in a dataset.
- $H0_4$: There is no difference between the presence and the absence of the Extreme Value, Missing Value and Suspect Sign smells on the Readability metric in a dataset.
- $H0_5$: There is no difference between the presence and the absence of the Extreme Value, Missing Value and Suspect Sign smells on the Disparate Impact metric in a dataset.
- $H0_6$: There is no difference between the presence and the absence of the Extreme Value, Missing Value and Suspect Sign smells on the Statistical Parity Difference metric in a dataset.
- $H0_7$: There is no difference between the presence and the absence of the Extreme Value, Missing Value and Suspect Sign smells on the Fairness Consistency metric in a dataset.

When a null-hypothesis can be rejected with an high confidence is it possible to accept the alternative hypothesis, which admits the negative effect that the data smells have on the fairness metrics:

- Ha_1: The Extreme Value, Missing Value and Suspect Sign smells negatively influence the Completeness metric in a dataset.
- Ha_2: The Extreme Value, Missing Value and Suspect Sign smells negatively influence the Uniqueness metric in a dataset.
- Ha_3: The Extreme Value, Missing Value and Suspect Sign smells negatively influence the Consistency metric in a dataset.
- Ha_4: The Extreme Value, Missing Value and Suspect Sign smells negatively influence the Readability metric in a dataset.
- Ha_5: The Extreme Value, Missing Value and Suspect Sign smells negatively influence the Disparate Impact metric in a dataset.
- Ha_6: The Extreme Value, Missing Value and Suspect Sign smells negatively influence the Statistical Parity Difference metric in a dataset.
- Ha_7: The Extreme Value, Missing Value and Suspect Sign smells negatively influence the Fairness Consistency metric in a dataset.

4.1.4 Statistical Testing

After the definition of our tool and our working hypothesis, we defined the statistical tests to understand if there are one or more dependency relationships between the number of data smell instances in a specific feature and the quality metrics computed on that feature. Starting from our research question we can derive our *predictor variable* and our *outcome variable*, with the count of instances of the data smells as predictor variables and one of the selected quality metrics as outcome variable, since we have 3 continuous predictor variables and 1 continuous outcome we will use multiple regression as statistical test, the test combinations are reported in the following Table 4.7:

Predictor Variable	Outcome Variable	Research Question
Extreme Value Smell count, Missing Value Smell count, Suspect Sign Smell count	Completeness	What is the effect of the selected data smells on the Completeness?
Extreme Value Smell count, Missing Value Smell count, Suspect Sign Smell count	Uniqueness	What is the effect of the selected data smells on the Uniqueness?
Extreme Value Smell count, Missing Value Smell count, Suspect Sign Smell count	Consistency	What is the effect of the selected data smells on the Consistency?
Extreme Value Smell count, Missing Value Smell count, Suspect Sign Smell count	Readability	What is the effect of the selected data smells on the Readability?
Extreme Value Smell count, Missing Value Smell count, Suspect Sign Smell count	Disparate Impact	What is the effect of the selected data smells on the Disparate Impact?
Extreme Value Smell count, Missing Value Smell count, Suspect Sign Smell count	Statistical Parity Difference	What is the effect of the selected data smells on the Statistical Parity Difference?
Extreme Value Smell count, Missing Value Smell count, Suspect Sign Smell count	Fairness Consistency	What is the effect of the selected data smells on the Fairness Consistency?

Table 4.7: Multiple Regression Tests

Based on the defined test we computed:

- **The Coefficient of Determination R-squared**: Indicates how much of the variability in the dependent variable can be explained by the independent variables in the model, and **Adjusted R-squared** which is a corrected version of R-squared that accounts for the number of predictors in the model.

- The **Coefficients associated with the independent variables** in order to understand their impact on the dependent variable.
- The **P-values** as an indicator that helps us assess the strength of evidence against the null hypothesis. A low p-value suggests strong evidence against the null hypothesis, while a high p-value suggests that we do not have sufficient evidence to reject it. It's important to note that a low p-value doesn't prove the "truth" of the alternative hypothesis but only that there is evidence against the null hypothesis.

4.2 Analysis of the Results

In this section we will answer our research questions based on the results of the tool and we will decide if accept or reject our working hypothesis based on the results of the statistical analysis.

RQ$_4$. *What is the prevalence of data smells?*

To address this question we analyzed the results of our tool, in particular the results of the module of data smell detection based on DSD, on the nineteen datasets that we selected. The results are summarized in the Table 4.8:

Data Smell	Total Count
Duplicated Value Smell	318
Extreme Value Smell	143
Missing Value Smell	119
Integer As Floating Point Number Smell	92
Casing Smell	26
Suspect Sign Smell	20
Floating Point Number As String Smell	2

Table 4.8: Distribution of Data Smells

Based on this results there is a consideration that is worth to take into account on the Duplicated Value Smell, this smell counts 318 faulty elements, but it has an high rate of false positive, since categorical attributes, like race {white, hispanic, african american, other} and many others are reported as duplicated value also if the values to assign are fixed.

So considering the smells on which we are working the most common is the Extreme Value Smell with a total count of 143 smelly instances, the second one is the Missing Value Smell with a total amount of 119 instances and the third is Suspect Sign Smell with a count of 20 faulty elements.

👉 **Answer to RQ_4.** To summarize the results, the most common data smell detected is the Duplicated Value Smell, with a total count of 318 faulty elements, but since it has a very high rate of false positive we ignored it. Then we have the Extreme Value Smell with a total amount of 143 smelly instances, the second one is the Missing Value Smell with a total amount of 119 instances and the third is Integer As Floating Point Number Smell with a count of 92 faulty elements followed by the Casing Smell and the Suspect Sign Smell with 26 a 20 faulty elements respectively. The last one is the Floating Point Number As String Smell with only 2 occurrences

RQ$_5$. *To what extent data smells impact data quality?*

To address this question, first of all, we collected all the quality metrics, namely, completeness, uniqueness, consistency, readability, disparate impact, statistical parity difference and fairness consistency, and all the features affected by data smells. Based on this data we run the multiple regression tests as described in the previous section.

The first result is about the impact of the data smells on the completeness metric:

Independent Variables	x1 = Extreme Value Smell Count x2 = Missing Value Smell Count x3 = Suspect Sign Smell Count
Dependent Variable	y = Completeness
R-squared	0.917
Adjusted R-squared	0.917
Coefficients	x1: 0.0003, x2: -0.0161, x3: 0.0001
P-values	x1: 0.6880, x2: 0.0000, x3: 0.8964

Table 4.9: Regression Results on Completeness

As we can see the r-squared and the adjusted r-squared are high, meaning that the model has good predictive power.

For x1, the coefficient is 0.0003, this means that a one-unit increase in Extreme Value Smell Count is associated with a 0.0003 increase in the effect on the dependent variable Completeness. However, the p-value associated with x1 is 0.6880, indicating that this coefficient is not statistically significant.

For x2, the coefficient is -0.0161. This suggests that a one-unit increase in Missing Value Smell Count is associated with a decrease of 0.0161 in the effect on the dependent variable Completeness. The p-value associated with x2 is very low 0.0000, indicating that the coefficient is statistically significant.

For x3, the coefficient is 0.0001. This suggests that a one-unit increase in Suspect Sign Smell Count is associated with a 0.0001 increase in the effect on the dependent variable Completeness. However, the p-value associated with x3 0.8964 also indicates that this coefficient is not statistically significant.

In summary, it appears that Missing Value Smell Count (x2) is the only one of the three independent variables that has a significant effect on the dependent variable Completeness.

Independent Variables	x1 = Extreme Value Smell Count x2 = Missing Value Smell Count x3 = Suspect Sign Smell Count
Dependent Variable	y = Uniqueness
R-squared	0.001
Adjusted R-squared	-0.005
Coefficients	x1: 0.0016, x2: -0.0003, x3: 0.0002
P-values	x1: 0.5021, x2: 0.6880, x3: 0.8635

Table 4.10: Regression Results on Uniqueness

The r-squared value is 0.001, which is very low. It suggests that the independent variables x1, x2, and x3 explain only a very small portion (approximately 0.1%) of the variation in the dependent variable y.

For x1, the coefficient is 0.0016. This means that a one-unit increase in Extreme Value Smell Count is associated with a 0.0016 increase in Uniqueness. However, the p-value associated with x1 is 0.5021, indicating that this coefficient is not statistically significant.

For x2, the coefficient is -0.0003. This suggests that a one-unit increase in Missing Value Smell Count is associated with a decrease of 0.0003 in Uniqueness. The p-value associated with x2 is 0.6880, indicating that this coefficient is not statistically significant.

For x3, the coefficient is 0.0002. This means that a one-unit increase in Suspect Sign Smell Count is associated with a 0.0002 increase in Uniqueness. However, the p-value associated with x3 is 0.8635, indicating that this coefficient is not statistically significant.

In summary, none of the independent variables (x1, x2, x3) appear to be statistically significant predictors of Uniqueness, as their p-values are all greater than the

typical significance threshold of 0.05. Suggesting that this metric may be influenced from different type of data smells, like the Duplicated Value Smell.

Independent Variables	x1 = Extreme Value Smell Count x2 = Missing Value Smell Count x3 = Suspect Sign Smell Count
Dependent Variable	y = Consistency
R-squared	0.138
Adjusted R-squared	0.132
Coefficients	x1: 0.0540, x2: 0.0225, x3: 0.0144
P-values	x1: 0.0000, x2: 0.0000, x3: 0.0324

Table 4.11: Regression Results on Consistency

The R-squared value is 0.138, which indicates that the independent variables x1, x2, and x3 collectively explain about 13.8% of the variation in the dependent variable y. This suggests that the model has some explanatory power, although there is still a significant amount of unexplained variation. The Adjusted R-squared value is 0.132. This value takes into account the number of independent variables in the model and adjusts for their inclusion. An Adjusted R-squared of 0.132 suggests that the model is reasonably good at explaining the variation in Consistency.

For x1, the coefficient is 0.0540. This means that a one-unit increase in Extreme Value Smell Count is associated with a 0.0540 increase in Consistency. The p-value associated with x1 is very low (0.0000), indicating that this coefficient is highly statistically significant.

For x2, the coefficient is 0.0225. This suggests that a one-unit increase in Missing Value Smell Count is associated with a 0.0225 increase in Consistency. The p-value associated with x2 is also very low (< 0.0001), indicating that this coefficient is highly statistically significant.

For x3, the coefficient is 0.0144. This means that a one-unit increase in Suspect Sign Smell Count is associated with a 0.0144 increase in Consistency. The p-value associated with x3 is 0.0324, which is less than the typical significance threshold of

0.05, indicating that this coefficient is statistically significant but to a lesser degree compared to x1 and x2.

In summary, all three independent variables (x1, x2, and x3) are statistically significant predictors of Consistency, as indicated by their low p-values. Among these variables, x1 (Extreme Value Smell Count) and x2 (Missing Value Smell Count) have stronger associations with Consistency, as their coefficients are larger and highly significant.

Independent Variables	x1 = Extreme Value Smell Count x2 = Missing Value Smell Count x3 = Suspect Sign Smell Count
Dependent Variable	y = Readability
R-squared	0.035
Adjusted R-squared	0.029
Coefficients	x1: 0.0159, x2: 0.0028, x3: 0.0042
P-values	x1: 0.0015, x2: 0.0937, x3: 0.1535

Table 4.12: Regression Results on Readability

The R-squared value is 0.035, which indicates that the independent variables x1, x2, and x3 collectively explain about 3.5% of the variation in the dependent variable y. This suggests that the model has limited explanatory power, and the majority of the variation in Readability remains unexplained. The Adjusted R-squared value is 0.029. This value takes into account the number of independent variables in the model and adjusts for their inclusion. An Adjusted R-squared of 0.029 suggests that the model's explanatory power is weak.

For x1 , the coefficient is 0.0159. This means that a one-unit increase in Extreme Value Smell Count is associated with a 0.0159 increase in Readability. The p-value associated with x1 is 0.0015, which is less than the typical significance threshold of 0.05, indicating that this coefficient is statistically significant.

For x2, the coefficient is 0.0028. This suggests that a one-unit increase in Missing Value Smell Count is associated with a 0.0028 increase in Readability. The p-value

associated with x2 is 0.0937, which is greater than the typical significance threshold of 0.05, indicating that this coefficient is not statistically significant.

For x3, the coefficient is 0.0042. This means that a one-unit increase in Suspect Sign Smell Count is associated with a 0.0042 increase in Readability. The p-value associated with x3 is 0.1535, which is also greater than 0.05, indicating that this coefficient is not statistically significant.

In summary, among the independent variables, only x1 (Extreme Value Smell Count) appears to be a statistically significant predictor of Readability, as its p-value is less than 0.05. However, the coefficient of x1 is relatively small, suggesting that it may not be a strong predictor in a practical sense. The other two independent variables, x2 and x3, do not appear to significantly influence Readability, as their p-values are greater than 0.05.

Independent Variables	x1 = Extreme Value Smell Count x2 = Missing Value Smell Count x3 = Suspect Sign Smell Count
Dependent Variable	y = Disparate Impact
R-squared	0.001
Adjusted R-squared	-0.079
Coefficients	x1: -0.0001, x2: -0.0014, x3: 0.0000
P-values	x1: 0.9314, x2: 0.8902, x3: nan

Table 4.13: Regression Results on Disparate Impact

Independent Variables	x1 = Extreme Value Smell Count x2 = Missing Value Smell Count x3 = Suspect Sign Smell Count
Dependent Variable	y = Statistical Parity Difference
R-squared	0.050
Adjusted R-squared	-0.026
Coefficients	x1: -0.0002, x2: 0.0002, x3: 0.0000
P-values	x1: 0.2661, x2: 0.9228, x3: nan

Table 4.14: Regression Results on Statistical Parity Difference

Independent Variables	x1 = Extreme Value Smell Count x2 = Missing Value Smell Count x3 = Suspect Sign Smell Count
Dependent Variable	y = Fairness Consistency
R-squared	0.065
Adjusted R-squared	-0.010
Coefficients	x1: 0.0001, x2: 0.0004, x3: 0.0000
P-values	x1: 0.2043, x2: 0.7823, x3: nan

Table 4.15: Regression Results on Fairness Consistency

For all the models dealing with the metrics of fairness we can summarize as suggested by the low r-squared and the negative adjusted r-squared that the models have a weak explanatory power, due to a lack of observations leading to an underfitting of the model.

In summary none of the independent variables x1, x2 appear to be statistically significant predictors of Disparate Impact, Statistical Parity Difference and Fairness Consistency, as their p-values are all greater than the typical significance threshold. Additionally there are no observations for the x3 (Suspect Sign Smell Count) leading to the impossibility of calculation of coefficients and p-values.

Finally, we conducted an analysis to assess the impact of the refactoring strategies applied to the datasets on the quality metrics. Initially, we ran the detection tool DSD to evaluate the effectiveness of the refactoring, identifying the absence of all previously detected Extreme Values, Missing Values, and Suspect Signs smells.

Then we computed the metrics on the new datasets in order to analyze the distribution of the values, in order to do it we used box plots showing on one column the distribution pre-rafactoring and on the other the distribution post-refactoring. Box plots, also known as box-and-whisker plots, are graphical representations that provide a visual summary of the distribution, central tendency, and variability of a dataset. They are particularly useful for identifying potential outliers and gaining insights into the data's overall spread.

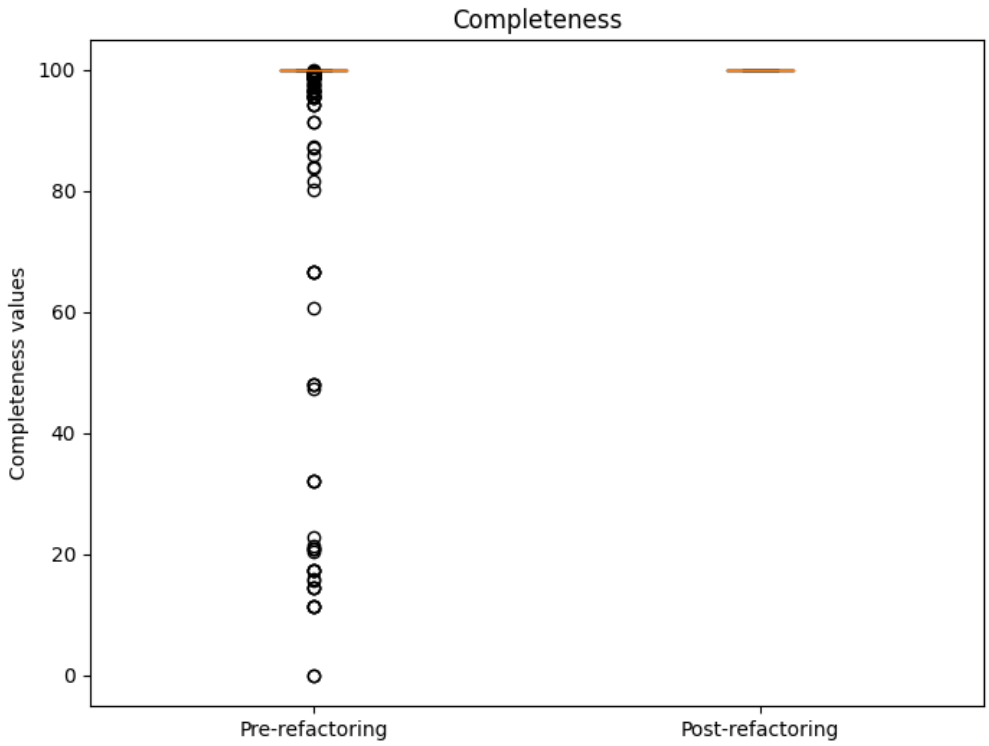

Figure 4.3: Completeness Box Plot

With regards to completeness Figure 4.3 shows that the median and the quartiles for both the distribution are similar indicating a similar distribution of values in the central part of the data, furthermore the pre-refactoring box plot presents some

outliers representing the features having missing values. As we can see the refactoring applied on the Missing Value Smell, which as shown by the multiple regression, is the only one of the three independent variables that has a significant effect on the dependent variable completeness, removed all the outliers of the distribution, leading all the features to have the 100% of completeness.

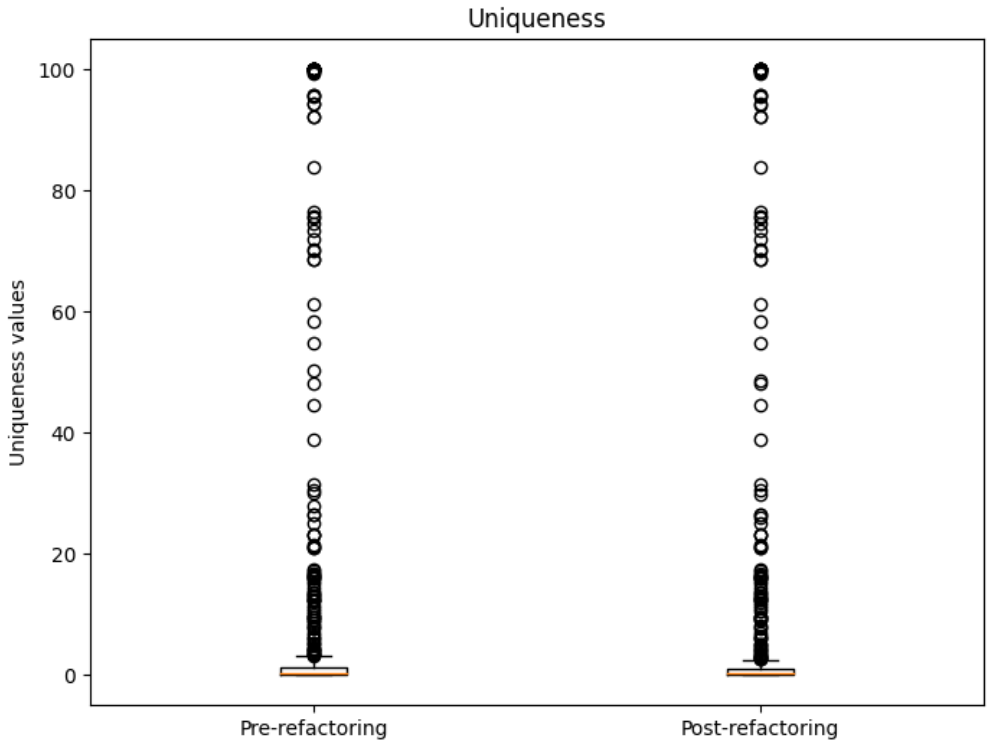

Figure 4.4: Uniqueness Box Plot

As also shown by the regression, none of the considered data smells influenced the values of uniqueness, this led to a non-significant variation in the distribution of the values, as we can see in Figure 4.4 the median and the quartiles as well as the outliers are similar for both the distributions.

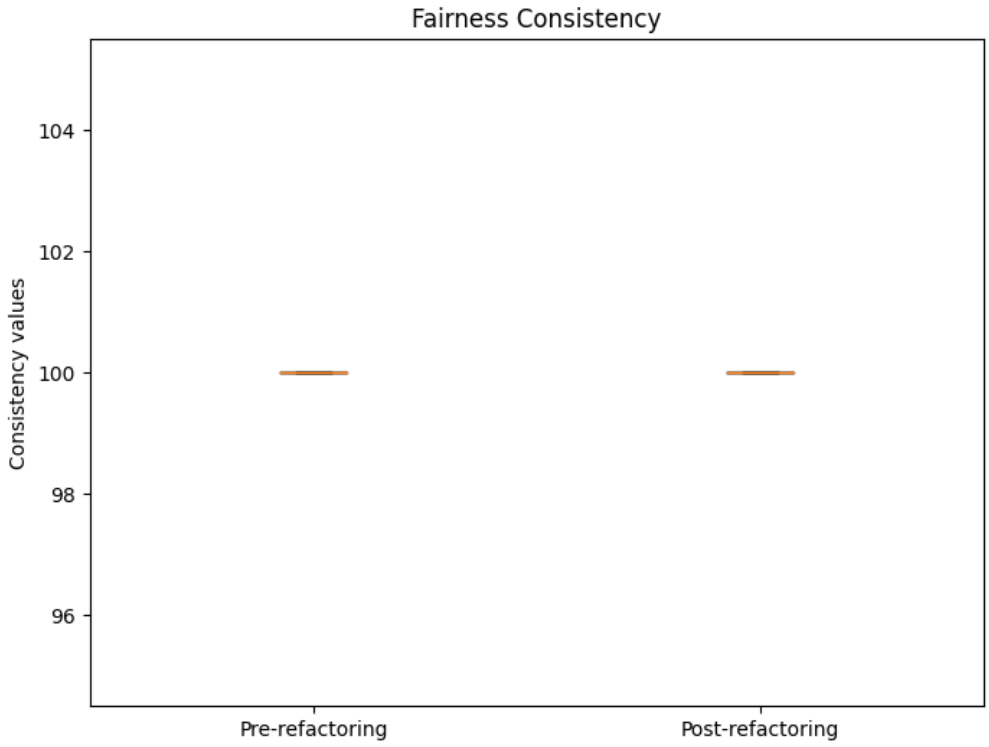

Figure 4.5: Consistency Box Plot

As showed in the box plot Figure 4.5, all the values of the features are completely consistent, both, before and after the application of the refactoring strategies, with a 100% of consistency for all the features. So in our study we cannot confirm an actual worsening or improvement due to the refactoring of the selected data smells.

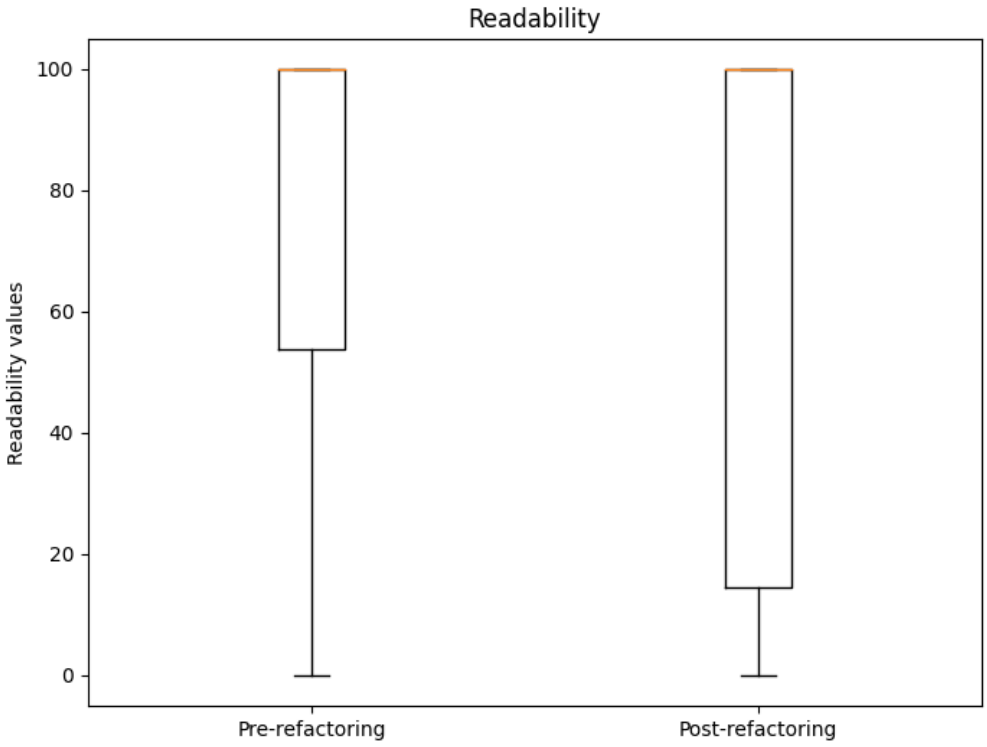

Figure 4.6: Readability Box Plot

As shown in Figure 4.6, regarding readability the median is the same for both the distribution, while we can see that first quartile is further from the median after the refactoring, showing that the application of the refactoring strategies led to a higher dispersion in the lower range of the distribution reflecting a slightly worsening in the metrics of readability of the features.

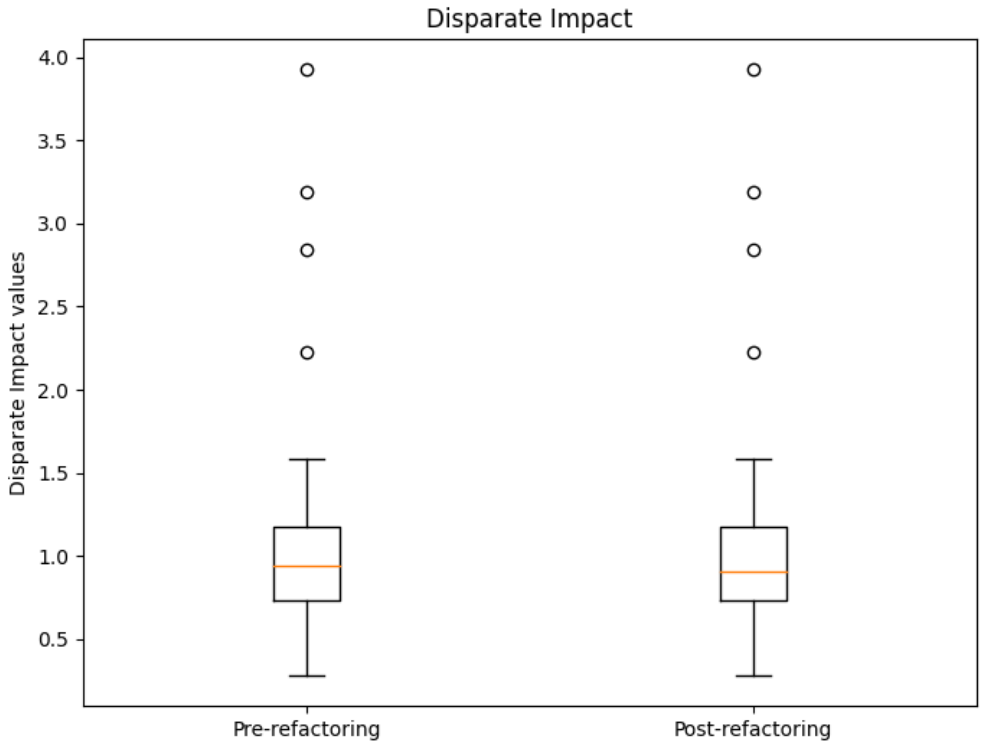

Figure 4.7: Disparate Impact Box Plot

Figure 4.7 shows that both box plots for the disparate impact share several key characteristics the medians for both variables are quite close, the positions of the first quartile and third quartile are identical for both variables, indicating a similar distribution in the central part of the data, the minimum and maximum values as well as the outliers are the same for both variables, reflecting a non significant impact of the refactoring on the quality metric.

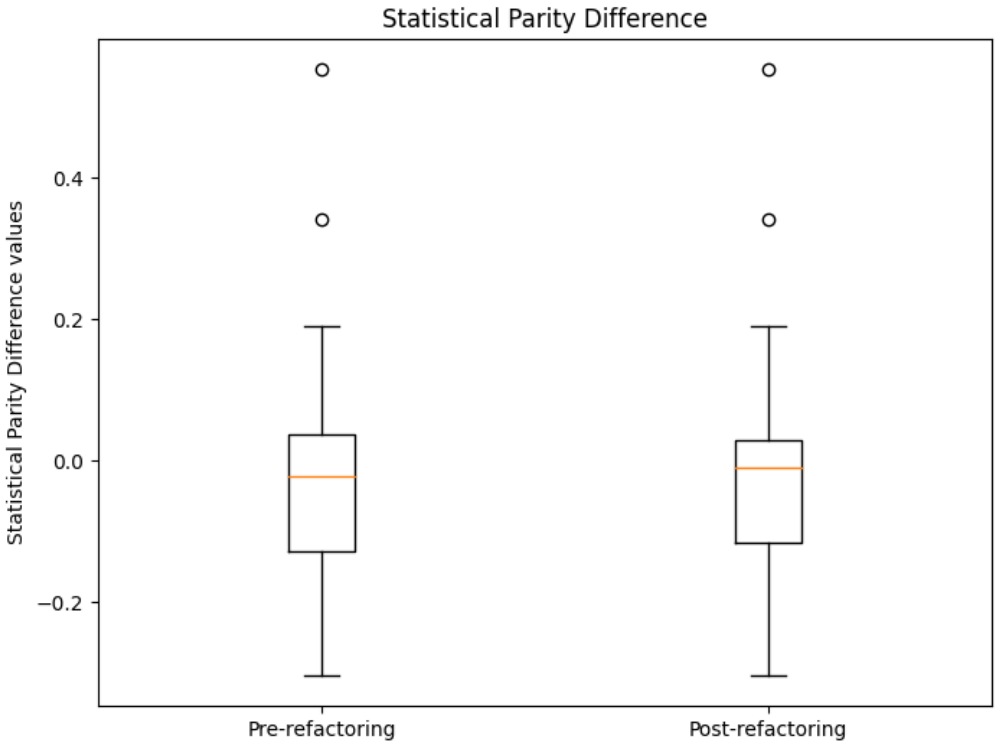

Figure 4.8: Statistical Parity Difference Box Plot

Regarding Statistical Parity Difference Figure 4.8 shows the medians for both variables are quite close with a low improvement after the refactoring, while the interquartile range become slightly shorter reflecting a lower dispersion of the values, the minimum and maximum values as well as the outliers are the same for both variables, reflecting a non significant impact of the refactoring on the quality metric.

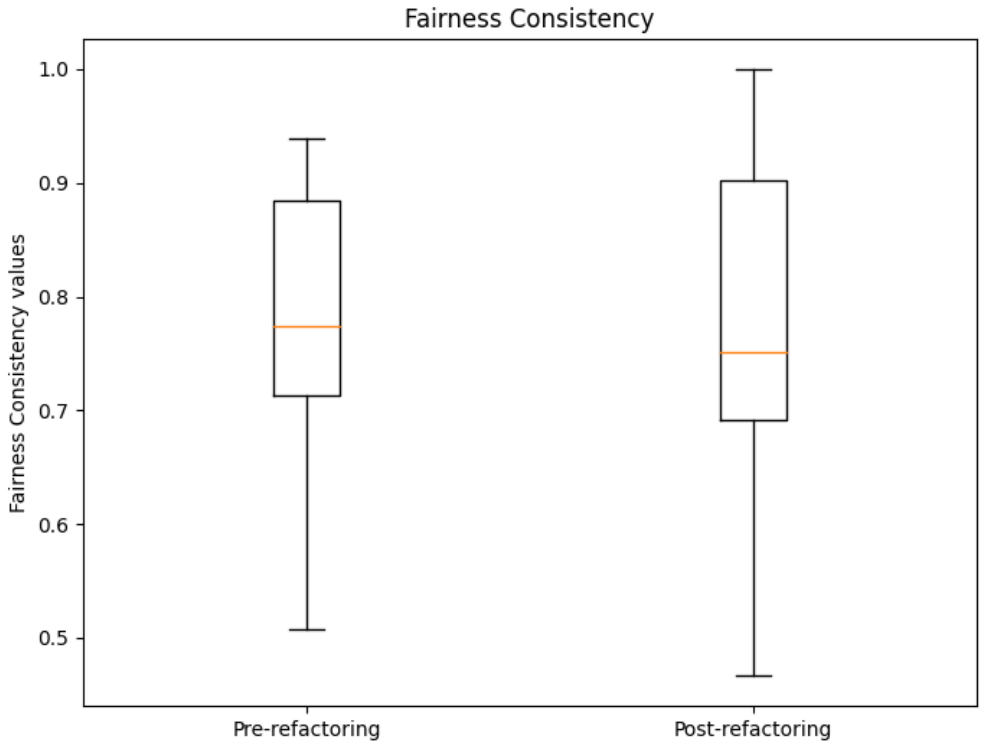

Figure 4.9: Fairness Consistency Box Plot

Last we analyze the impact of the refactoring on the fairness consistency. As showed in Figure 4.9 the medians are quite close for the two distribution with a slightly worsening after the refactoring, the positions of the first quartile and third quartile in the second box plot are notably different from those in the first box plot spanning a larger range, indicating a broader distribution of data. The interquartile range in the second box plot is wider compared to the first box plot, suggesting greater variability. The minimum and maximum values in the second box plot also extend to a greater range than those in the first box plot, indicating a more extensive spread of data points. The higher dispersion of the values is not an indicator of worsening or enanching of the metric so the data smells have not a significant impact on it.

In conclusion we can reject the null-hypothesis $H0_1$ since we found that the Missing Value Smell negatively affect the Completeness metric, but we failed to reject the other null-hypothesis since there is a lack of evidences of the negative effects of the data smells on them, this could be caused by lack of a sufficient quantity of datasets and observations, and by the restricted types of data smell detectable by the tool.

👉 **Answer to RQ$_5$.** To summarize the results for the RQ5, we analyzed the correlation between the selected data smells, namely, Extreme Value Smell, Missing Value Smell and Suspect Sign Smell, and the selected metrics of Data Quality using a multiple regression, the results showed a strong impact, in particular of the Missing Value Smell on the metric of Completeness, and a lower impact of the three data smells on the Consistency, while other metrics like Uniqueness and Readability doesn't seems to be impacted, but may be more influenced by different data smells like Duplicated Value Smells or different types of Encoding Smells or Syntactic Smells.

CHAPTER 5

Threats To Validity

5.1 Threats To Internal Validity

Threats to internal validity concern factors that might have influenced the causal relationship between treatment and outcome.

5.1.1 Use of Datasets

Throughout this research, the need of a large set of datasets was essential for conducting a comprehensive analysis of data smells. However, this approach introduces certain threats to data validity:

Dataset Selection: A threat to validity arises from the selection of datasets included in the study. The datasets analyzed in this research have been studied in the scope of machine learning fairness and may have specific characteristics or issues that do not represent the entire population of data, potentially influencing the results.

To mitigate these threat, we included the datasets of a second study, considering a wider context of application, increasing the number of dataset to 19.

5.1.2 Use of Scopus as a Search Engine

The exclusive use of Scopus as the search engine for collecting research papers introduces a threat to research validity:

Coverage Limitations: Scopus may not index all relevant papers about the topic covered in this research or have access to sources not present in its database. This may result in the exclusion of pertinent papers.

To mitigate this threat, a snowballing approach was adopted. This method allowed the research to be extended iteratively, identifying and including papers that might not have been initially accessible through Scopus. This strategy helped ensure a broad spectrum of research relevant to the topic.

5.2 Threats To External Validity

Threats to external validity are conditions that limit the ability to generalize the results to a broader environment.

5.2.1 Lack of observations

The major threat to the external validity of this study is the small number of datasets analyzed and the restricted number of types of data smells selected for the research. Although we tried to extend the number of datasets including data from different application context, there are some metrics that suffered from a lack of observation, like the fairness-related metrics. This makes it difficult to generalize this results out of the scope of this research.

CHAPTER 6

Conclusion

In this chapter we will carry out a summary for the work conducted.

6.1 Systematic Literature Review on Data Smells

In this thesis we faced an aspect of data quality that should be taken more and more into account while data become the most important asset for any machine learning-based system, but also for any data-centric business, the so called data smells. First of all, we conducted a systematic literature review in order to understand the current consideration of this aspect in the state of the art, in particular we wanted to know which type of data smells were defined, the impact that they can have on a machine learning system and its non-functional requirements, and finally which were the tools presented in the literature to address this problem.

We used the Goal Question Metric approach [16] in order to elicit the research questions of our study and then we followed the guidelines proposed by Kitchenham and Charters [15] to carry out the systematic literature review and finally we used the snowballing methodology defined by Wohlin [17], in order to adopt a systematic inclusion of references.

To summarize the results of the first research question we defined seven classes of data smell, namely, Believability Smells, Encoding Smells, Syntactic Smells, Consistency Smells, Redundant Value Smells, Categorical Smells and Fairness Smells, with a final catalog of 46 data smells. 16 more data anomalies can be found in the literature, but they can be mapped to one of the smells defined previously, while others are related to the definition of the database schema instead of the dataset itself.

The second research question showed that, in the literature, there are different examples of how a lack of data quality can influence non-functional requirements, such as defect proneness, safety, and maintainability. Since the data are defined in the initial steps of the pipeline, the presence of data smells could raise a degradation of the model in combination with other issues related to technical debt specific to AI-based systems (i.e., Pipeline Jungles and Hidden Feedback Loops)

Finally to summarize the results of the third research question we found that smelly data cannot always be mapped to data errors, so they could not be suitable to be detected by validation tools, this led Harald Foidl et al. [SLR1] to the development of two tools for data smell detection, based on two research strategies, namely rule based detection and machine learning-based detection. Then we found out that the main study about the presence of data smells in public datasets has been carried out by Arumoy Shome et al. [SLR2] finding out that redundant value smells and the categorical value smells are the most common categories of smell.

6.2 The impact of Data Smells on Data Quality

In the second half of this work we wanted to understand if the presence of data smells in a dataset influencing its distribution of values, could impact the data quality metrics leading to potential issues. Our main goals for this phase were understanding the prevalence of data smells and to what extent data smells impact data quality.

In order to address this problem we analyzed 19 famous public datasets studied in the scope of classification tasks and machine learning fairness and we built a tool able to detect the data smells, refactor 3 of them and compute different data quality metrics.

During the analysis of the data we found that the most common data smell detected in the studied datasets is the Duplicated Value Smell, with a total count of 318 faulty elements, but since it has a very high rate of false positive we ignored it. Then we have the Extreme Value Smell with a total amount of 143 smelly instances, the second one is the Missing Value Smell with a total amount of 119 instances and the third is Integer As Floating Point Number Smell with a count of 92 faulty elements followed by the Casing Smell and the Suspect Sign Smell with 26 a 20 faulty elements respectively. The last one is the Floating Point Number As String Smell with only 2 occurrences.

Finally to answer the RQ5, we analyzed the correlation between the selected data smells, namely, Extreme Value Smell, Missing Value Smell and Suspect Sign Smell, and the selected metrics of Data Quality using a multiple regression, the results showed a strong impact, in particular of the Missing Value Smell on the metric of Completeness, and a lower impact of the three data smells on the Consistency.

6.3 Future Works

In this section, we outline potential avenues for further research in the domain of data quality and its implications for AI-specific quality issues.

Tool enhancement: First of all, one of the future works may be the enhancement of the tool we developed, the detection module could be extended with more data smells defined in the literature and with more detection strategies, like the machine learning based ones. Furthermore, the set of metrics computed by the tool could extended including an important aspect of the data quality, namely, the time-related metrics, like Timeliness, Volatility and Integrity over the data life cycle.

Exploring Additional Data Smells Correlations: A promising direction for future research involves expanding our understanding of how various other data smells correlate with the metrics we have considered thus far. While we have already examined the correlation between data smells such as Extreme Value Smell, Missing Value Smell, and Suspect Sign Smell with specific data quality metrics, there are numerous other data smells within the broader category. For instance, metrics like Uniqueness and Readability don't seems to be impacted by the selected smells, but

may be more influenced by different ones like Duplicated Value Smells or types of Encoding or Syntactic Smells. Investigating the impact of these additional data smells on data quality metrics, including but not limited to Completeness, Consistency, Uniqueness, and Readability, would provide a more comprehensive view of their influence.

Studying Data Smells in Relation to AI-Specific Quality Issues: In the landscape of AI and machine learning, it is crucial to assess the connection between data smells and AI-specific quality issues. Future work should focus on how data quality concerns, represented by data smells, can propagate into AI-specific quality issues. For instance, research could delve into the extent to which data smells influence prediction fairness, safety, and other AI-specific quality criteria.

By addressing these areas of future research, we can advance our knowledge of data quality and its far-reaching implications within the realm of AI, paving the way for more robust, reliable, and ethical machine learning systems.

Bibliography

[1] A. Burkov, *Machine Learning Engineering*. True Positive Incorporated, 2020. [Online]. Available: https://books.google.it/books?id=DSj9zQEACAAJ (Citato alle pagine 5, 6, 7, 8, 9, 10, 11 e 13)

[2] G. Recupito, F. Pecorelli, G. Catolino, S. Moreschini, D. D. Nucci, F. Palomba, and D. A. Tamburri, "A multivocal literature review of mlops tools and features," in *2022 48th Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*, 2022, pp. 84–91. (Citato a pagina 12)

[3] Y. Zhou, Y. Yu, and B. Ding, "Towards mlops: A case study of ml pipeline platform," in *2020 International conference on artificial intelligence and computer engineering (ICAICE)*. IEEE, 2020, pp. 494–500. (Citato a pagina 13)

[4] A. Jain, H. Patel, L. Nagalapatti, N. Gupta, S. Mehta, S. Guttula, S. Mujumdar, S. Afzal, R. Sharma Mittal, and V. Munigala, "Overview and importance of data quality for machine learning tasks," in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery amp; Data Mining*, ser. KDD '20. New York, NY, USA: Association for Computing Machinery, 2020, p. 3561–3562. [Online]. Available: https://doi.org/10.1145/3394486.3406477 (Citato a pagina 14)

[5] M. Gan, Z. Yucel, and A. Monden, "Improvement and evaluation of data consistency metric cil for software engineering data sets," *IEEE Access*, p. 1, 2022, publisher Copyright: Author. (Citato a pagina 15)

[6] V. Gudivada, A. Apon, and J. Ding, "Data quality considerations for big data and machine learning: Going beyond data cleaning and transformations," *International Journal on Advances in Software*, vol. 10, no. 1, pp. 1–20, 2017. (Citato a pagina 15)

[7] L. L. Pipino, Y. W. Lee, and R. Y. Wang, "Data quality assessment," *Communications of the ACM*, vol. 45, no. 4, pp. 211–218, 2002. (Citato a pagina 15)

[8] u/Worried_Conference84, "What is data quality, and why should we care." [Online]. Available: https://www.reddit.com/user/Worried_Conference84/comments/vg3v3c/what_is_data_quality_and_why_should_we_care/ (Citato alle pagine 15 e 16)

[9] K. M. Habibullah and J. Horkoff, "Non-functional requirements for machine learning: Understanding current use and challenges in industry," 2021. (Citato a pagina 17)

[10] J. Dastin, "Amazon scraps secret ai recruiting tool that showed bias against women." [Online]. Available: https://www.reuters.com/article/us-amazon-com-jobs-automation-insight-idUSKCN1MK08G (Citato a pagina 18)

[11] Z. Obermeyer, B. Powers, C. Vogeli, and S. Mullainathan, "Dissecting racial bias in an algorithm used to manage the health of populations," *Science*, vol. 366, no. 6464, pp. 447–453, 2019. [Online]. Available: https://www.science.org/doi/abs/10.1126/science.aax2342 (Citato a pagina 18)

[12] T. N. Fitria, "Gender bias in translation using google translate: Problems and solution." (Citato a pagina 18)

[13] J. Angwin and J. Larson, "Machine bias - there's software used across the country to predict future criminals. and it's biased against blacks." [Online]. Available: https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing#disqus_thread (Citato a pagina 18)

[14] S. Verma and J. Rubin, "Fairness definitions explained," in *Proceedings of the International Workshop on Software Fairness*, ser. FairWare '18. New York, NY, USA: Association for Computing Machinery, 2018, p. 1–7. [Online]. Available: https://doi.org/10.1145/3194770.3194776 (Citato alle pagine 18, 19, 20, 21 e 56)

[15] B. Kitchenham and S. Charters, "Guidelines for performing systematic literature reviews in software engineering," vol. 2, 01 2007. (Citato alle pagine 22, 25, 26 e 80)

[16] V. R. B. G. Caldiera and H. D. Rombach, "The goal question metric approach," *Encyclopedia of software engineering*, pp. 528–532, 1994. (Citato alle pagine 22 e 80)

[17] C. Wohlin, "Guidelines for snowballing in systematic literature studies and a replication in software engineering," ser. EASE '14. New York, NY, USA: Association for Computing Machinery, 2014. [Online]. Available: https://doi.org/10.1145/2601248.2601268 (Citato alle pagine 22 e 80)

[18] ——, "Second-generation systematic literature studies using snowballing," in *Proceedings of the 20th International Conference on Evaluation and Assessment in Software Engineering*, ser. EASE '16. New York, NY, USA: Association for Computing Machinery, 2016. [Online]. Available: https://doi.org/10.1145/2915970.2916006 (Citato a pagina 26)

[19] T. L. Quy, A. Roy, V. Iosifidis, W. Zhang, and E. Ntoutsi, "A survey on datasets for fairness-aware machine learning," *WIREs Data Mining and Knowledge Discovery*, vol. 12, no. 3, mar 2022. [Online]. Available: https://doi.org/10.1002%2Fwidm.1452 (Citato alle pagine 45, 46, 47, 48, 49, 50, 51 e 52)

[20] M. Hirzel and M. Feffer, "A suite of fairness datasets for tabular classification," 2023. (Citato alle pagine 45, 46, 49, 51 e 53)

[21] W. Elouataoui, I. El Alaoui, S. El Mendili, and Y. Gahi, "An advanced big data quality framework based on weighted metrics," *Big Data and Cognitive Computing*, vol. 6, no. 4, 2022. [Online]. Available: https://www.mdpi.com/2504-2289/6/4/153 (Citato a pagina 55)

SLR References

[SLR1] H. Foidl, M. Felderer, and R. Ramler, "Data smells: Categories, causes and consequences, and detection of suspicious data in ai-based systems," in *Proceedings of the 1st International Conference on AI Engineering: Software Engineering for AI*, 2022, pp. 229–239.

[SLR2] A. Shome, L. Cruz, and A. van Deursen, "Data smells in public datasets," in *Proceedings of the 1st International Conference on AI Engineering: Software Engineering for AI*, ser. CAIN '22. New York, NY, USA: Association for Computing Machinery, 2022, p. 205–216. [Online]. Available: https://doi.org/10.1145/3522664.3528621

[SLR3] R. Foorthuis, "A typology of data anomalies," *Communications in Computer and Information Science*, vol. 854, pp. 26–38, 2018, cited By 7. [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85063899472&doi=10.1007%2f978-3-319-91476-3_3&partnerID=40&md5=6675efdd20e4f01d7ad3fb8e7afa86da

[SLR4] D. Sukhobok, N. Nikolov, and D. Roman, "Tabular data anomaly patterns," vol. 2018-January, 2018, pp. 25–34, cited By 10. [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85030672050&doi=10.1109%2fInnovate-Data.2017.10&partnerID=40&md5=33657ef291fd323d8d74edf13f5f4256

[SLR5] V. Golendukhina, H. Foidl, M. Felderer, and R. Ramler, "Preliminary findings on the occurrence and causes of data smells in a real-world business travel data processing pipeline," 2022, pp. 18–21, cited By 0. [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85143197103&doi=10.1145%2f3549037.3561275&partnerID=40&md5=a50135f57b790372a91ec909e7f4aae9

Acknowledgements

This is the end a journey of five years and I have to reserve this section to address my gratitude to the people who have contributed to the achievement of this goal.

I would like to express my deepest gratitude to Prof. Fabio Palomba, my supervisor during this work. Thanks to be a point of reference since the end of the Bachelor's degree and throughout the Master's degree and thanks for all the precious knowledge shared during the lectures and the meetings.

Thanks to Carmine Ferrara and Gilberto Recupito, my co-supervisors during this work. Thanks for the patience you showed during this year and for all the help and knowledge you shared with me.

I'm extremely grateful to all the members of the C15 team Leopoldo, Paolo, Carlo, Alessandro, Mario, Vincenzo and Marco. Thank you for making my last academic year the most fun and full of memories of my entire career.

Special thanks to all the friends that walked with me during all this years Vincenzo, Bruno, Felice, Stefania, Rosario and all the others that helped me during my path making everything lighter with their presence.

Words cannot express my gratitude to Simone, Matilde, Lucrezia and Giovanni. Thanks to all the moments we shared in the last 3 years you changed my life in the best way possible.

Last, but not least special thanks to all my family who supported me during all my life and whom i know will continue to do so.

This thesis helped to plant a tree in Kenya through the Treedom project.

https://www.treedom.net/it/user/sesalab/event/sesa-random-forest