Institute of Parallel and Distributed Systems

University of Stuttgart
Universitätsstraße 38
D–70569 Stuttgart

Bachelorarbeit

Fairness-Monitoring in Data Preparation Pipelines

Patrick Epple

Course of Study: Informatik

Examiner: Prof. Dr. Melanie Herschel

Supervisor: Prof. Dr. Melanie Herschel

Commenced: October 1st, 2023

Completed: April 1st, 2024

Kurzfassung

Maschinelles Lernen (ML) ist in den heutigen Produktionsumgebungen unentbehrlich geworden und zielt auf leistungsstarke Modelle ab [FS20]. Die Leistung dieser Modelle hängt jedoch in hohem Maße von der Datenqualität der zugrunde liegenden Daten ab [JPN+20]. Daher werden Data Preparation Pipelines eingesetzt, um hohe Datenqualitätsstandards zu gewährleisten. Bei der Sicherstellung der Datenqualität kann es zu unbeabsichtigten Ungerechtigkeiten kommen, die eine bestimmte Gruppe begünstigen. Dies kann schwerwiegende Auswirkungen auf Entscheidungsprozesse haben [VLA19].

In dieser Arbeit befassen wir uns mit der Herausforderung, Fairness in Data Preparation Pipelines zu gewährleisten, indem wir ein System entwickeln, das Benutzer warnt, wenn potenzielle Unfairness in einem oder mehreren Schritten der Pipeline eingeführt worden sein könnte. Anhand einer umfassenden Literaturrecherche identifizieren wir die üblicherweise verwendeten Fairness-Metriken und untersuchen die Beziehung zwischen Fairness und typischen Datencharakteristika. Auf der Grundlage dieses Verständnisses entwickeln wir ein System, das darauf abzielt, Schritte zu erkennen, in denen innerhalb der Data Preparation Pipeline Unfairness eingeführt worden sein könnte, wenn das resultierende ML-Modell bestimmte Fairness-Metriken nicht erfüllen kann.

Wir bewerten die Leistung unseres vorgeschlagenen Systems in Bezug auf Fairness über verschiedene Benchmark-Datensätze und unterschiedliche Data Preparation Pipelines hinweg. Wir stellen fest, dass unser System eine geeignete Möglichkeit darstellt, um Schritte innerhalb einer Data Preparation Pipeline zu erkennen, die die Fairness des resultierenden Modells potenziell negativ beeinflusst haben könnten, und somit den Nutzern wertvolles Feedback zu geben. Wir erkennen jedoch auch die Grenzen unserer Arbeit an, darunter die Fokussierug auf ausschließlich binäre Klassifikationsaufgaben und die Berücksichtigung nur einer Auswahl von Fairness-Metriken und Datencharakteristika.

Zusammenfassend lässt sich sagen, dass unsere Arbeit einen Beitrag zum Bereich der fairen Data Preparation Pipelines leistet, indem sie einen Rahmen bietet, der Fairness innerhalb von Data Preparation Pipelines gewährleisten soll. Durch die Auseinandersetzung mit dem Zusammenspiel von Fairness und Datenvorverarbeitung wollen wir gerechtere Entscheidungsprozesse für ML-Modelle in der realen Anwendung fördern.

Abstract

Machine Learning (ML) has become essential in today's production environments, which aim for high-performance models [FS20]. However, the performance of these models highly depends on the data quality of the underlying data [JPN+20]. Therefore, data preparation pipelines are used to ensure high data quality standards [FS20]. In the process of ensuring data quality, it is possible to unintentionally introduce unfairness that favors a specific group. This can lead to serious implications for decision-making processes [VLA19].

In this work, we address the challenge of ensuring fairness in data preparation pipelines by developing a system that alerts users if potential unfairness may have been introduced in one or multiple steps of the pipeline. Through a comprehensive literature review, we identify fairness metrics commonly used and investigate the relationship between fairness and typical data characteristics. Based on this understanding, we develop a system that aims to detect steps in which unfairness may have been introduced within the data preparation pipeline if the resulting ML model cannot satisfy certain fairness metrics.

In terms of fairness, we evaluate the performance of our proposed system across diverse benchmark datasets and various data preparation pipelines. We find that our system is a suitable method to detect steps within a data preparation pipeline that have potentially negatively influenced the fairness of the resulting model and thereby provide valuable feedback to the users. However, we also acknowledge the limitations of our work, including the focus on binary classification tasks and the consideration of only selected fairness metrics and data characteristics.

In conclusion, our work contributes to the field of data preparation pipelines by offering a framework designed to ensure fairness within data preparation pipelines. By addressing the intersection of fairness and data preprocessing, we aim to foster fairer decision-making processes for ML models in real-world applications.

Contents

List of Figures

List of Tables

List of Algorithms

1 Introduction

1.1 Motivation and Context

Machine Learning (ML) has become increasingly important in modern production environments, which aim for high-performance models [FS20]. However, the performance of these models is restricted by the data quality [JPN+20]. Hence, achieving high-performance models from ML strongly depends on the quality of the underlying data. In practice, many ML projects fail because of poor data quality [FS20]. Therefore, it is important to ensure that the data used for ML follows certain data quality criteria. The quality of data is judged based on data quality dimensions such as accuracy, completeness, consistency, and timeliness [BCFM09; PLW03; Why24].

To address this challenge and ensure good data quality, a data preparation pipeline is usually used in practice [FS20]. More concretely, a data preparation pipeline is used to increase the quality of a dataset and, ultimately, obtain high-quality data that can then be used for ML or other purposes [FS20]. Such a pipeline consists of consecutively executed steps that analyze, cleanse, reduce, and manipulate the dataset [FS20]. After passing all these steps, the dataset is ready for the training algorithm.

The described data preprocessing may have some side effects [VLA19]. Especially *fairness* can suffer under specific data preprocessing steps [VLA19]. Fairness has become a central issue in ML in recent years since many decision-making processes are currently in the hands of ML models. Therefore, decision-making processes must not be biased and must ensure fair outcomes.

In a dataset, unfairness is usually reflected in specific data characteristics such as an imbalanced dataset, limited size, or an excess of missing values, also called sparsity [Ben01; BR21; FMST20; LGA22; SG02]. However, implementing data preprocessing steps within pipelines may accidentally introduce unfairness.

The impact of biased ML models becomes particularly important in real-world scenarios, such as credit approval in banking institutions. Banks rely heavily on historical data to decide whether applicants get credit approval [GGR23]. However, the historical data may encode unfairness, particularly disadvantaging certain demographic groups [GGR23]. Customers' credit history often plays a vital role in these datasets, leading to discrimination of young individuals, students, or immigrants who often do not have such a credit history [GGR23]. In other words, if two people with identical data except age apply for credit, the older person is more likely to get approved by the bank. This concrete example underlines the importance of this topic in practice.

1.2 Contribution of this Work

Based on existing literature, our research aims to gain a comprehensive understanding of the various fairness metrics commonly used to evaluate the fairness of ML models. Additionally, we seek to understand the relationship between fairness and typical data characteristics such as size, balance, and sparsity.

The primary research gap we address in this work consists of a framework that alerts users when unfairness may arise during dataset modifications in the data preparation pipeline. More precisely, We aim to develop a system that enables preprocessing through a data preparation pipeline, with the additional feature of notifying users about potential increases in unfairness resulting from the implemented steps. To determine when fairness may have deteriorated significantly, we aim to use data characteristics based on our findings from the literature research.

In order to develop such a system, which is the overarching goal of this work, we will follow the subsequent questions:

- What dataset characteristics should be monitored to assess fairness-relevant data quality of intermediate results of a data preparation pipeline?
- How can these characteristics be efficiently computed or estimated?
- How can a system be built ensuring fairness in data preparation pipelines based upon these techniques?

By addressing these questions, we aim to ensure a fair dataset, leading to fair ML models and fair decisions in real-life applications.

1.3 Structure

Our work will unfold several steps to address the identified research gap and questions. Following this introduction, Chapter 2 will provide a comprehensive literature review. This review will give an overview of various fairness metrics, examine the relationship between data characteristics and fairness metrics, and analyze existing approaches for detecting unfairness in preprocessing and debugging data preparation pipelines.

In Chapter 3, we will present the baseline solution of our system. We will start by outlining the concept of the framework before showing the considered data characteristics and fairness metrics. Subsequently, we will present the algorithms required to determine whether a warning should be announced to users, as well as profile the selected data characteristics.

In Chapter 4, we will conduct an experiment to evaluate the performance of our developed system in terms of fairness based on various data preparation pipelines.

Finally, Chapter 5 will provide an overview of our findings, discuss the limitations of our work, and offer insights into potential future directions for research in this area.

2 Related Work

The foundation of our work is mainly based on prior research in the fields of fairness metrics, the relationship between common data characteristics and fairness, and the identification of existing unfairness detection systems. This chapter provides a concise overview of the relevant research in these specific fields.

2.1 Fairness Metrics

Fairness is generally known to be "the quality of treating people equally or in a way that is right or reasonable" [Cam23]. In the context of ML and the resulting models on which decisions are based, the term fairness has to be extended. Specifically, a measurable metric definition is needed to evaluate whether a ML model is fair. This is highly important to the research gap we are trying to close. Specifically, we need to determine whether an ML model can be considered fair. Different definitions of fairness metrics are present in the literature, which we outline in the following.

Before establishing these fairness metrics, we must define the term *protected attribute*. A protected attribute, sometimes also called *sensitive attribute*, is an attribute that must not be used for discriminatory purposes, such as gender, age, or race [Gri22]. Furthermore, if the protected attribute is binary, we can divide instances into *privileged* and *unprivileged* depending on the value of the attribute and the context [PCM23].

Generally, fairness definitions can be divided into two different types, namely *Individual Fairness* and *Group Fairness*:

- *Individual Fairness:* Individuals should be treated equally. More precisely, this means that similar predictions should be given to similar Individuals. [DHP+12; KLRS17; MMS+21]
- *Group Fairness:* As the name already says, groups should be treated equally [MMS+21]. In other words, similar predictions should be given to similar groups.

Similar to Mehrabi et al. and according to their findings, we provide a table with the nine fairness metrics we present in this section to assign them either to Individual Fairness, Group Fairness, or both [MMS+21]. The categorization is depicted in Table 2.1.

The fairness metric *Equalized Odds* was introduced by Hardt et al. [HPS16] and states that protected and unprotected attributes should have the same rate for true positives as well as for false positives [HPS16; MMS+21; VR18]. More formally, equalized odds are satisfied by a predictor $\hat{Y}$ "with respect to protected attribute A and outcome Y, if $\hat{Y}$ and A are independent conditional on Y" [HPS16]. Therefore, it must hold: $\mathbb{P}(\hat{Y}|A = 0, Y = y) = \mathbb{P}(\hat{Y}|A = 1, Y = y)$ with $y \in \{0, 1\}$ [HPS16]

In addition to the metric Equalized Odds, Hardt et al. also suggested the fairness metric *Equal Opportunity* [HPS16]. This definition states that protected and unprotected attributes should have the same rate for true positives [HPS16; MMS+21; VR18]. Formally, "a binary predictor $\hat{Y}$ satisfies equal opportunity with respect to A and Y if $\mathbb{P}(\hat{Y}|A = 0, Y = 1) = \mathbb{P}(\hat{Y}|A = 1, Y = 1)$" [HPS16]. Therefore, the Equal Opportunity definition is a weaker metric than the Equalized Odds metric since the Equalized Odds definition also covers it. By doing that, everybody gets an equal chance from the beginning (e.g., to get a loan from the bank) [HPS16]. In other words, the Equal Opportunity metric relaxes the Equalized Odds metric.

A well-known and widespread fairness metric is *Demographic Parity*. This definition states that the probability of a positive result should be the same for protected attributes and unprotected attributes [MMS+21; VR18]. Formally, a predictor $\hat{Y}$ meets the requirements of the definition if the following holds: $\mathbb{P}(\hat{Y}|A = 0) = \mathbb{P}(\hat{Y}|A = 1)$ [KLRS17]

Another popular fairness metric is *Conditional Statistical Parity*. This definition extends the Demographic Parity metric by as set L of legitimate attributes that influence the result [VR18]. As before, this fairness definition states that the probability of a positive result should be the same for protected and unprotected attributes based on the known legitimate attributes [MMS+21; VR18]. More formally, a predictor $\hat{Y}$ meets the requirements of the definition with legitimate attributes L if the following holds: $\mathbb{P}(\hat{Y}|L = 1, A = 0) = \mathbb{P}(\hat{Y}|L = 1, A = 1)$, [CPF+17; MMS+21].

The metric *Fairness through Unawareness* is another basic definition to assess fairness. It states, "An algorithm is fair so long as any protected attributes A are not explicitly used in the decision-making process" [KLRS17].

The fairness metric *Fairness through Awareness* is strongly related to the previous metric, Fairness through Unawareness. In contrast, an algorithm is considered fair "if it gives similar predictions to similar individuals" [KLRS17]. More precisely, if the input of a fair algorithm is similar, the output should also be similar [MMS+21]. Similarity could be determined by using an appropriate metric.

The metric *Treatment Equality* defines the ratio of the false positive rate and the false negative rate of protected attributes should be the same [BHJ+21].

Another metric for fairness is *Test Fairness*, sometimes also called *Calibration*. This definition states that the probability of recidivism should be the same regardless of the group membership [Cho16]. Formally, for all values of a score S and groups $R \in \{b, w\}$, the following formula must hold: $\mathbb{P}(Y = 1|S = s, R = b) = \mathbb{P}(Y = 1|S = s, R = w)$ [Cho16]
In other words, given a Score S = S(x), protected and unprotected attributes must have the same probability of belonging to the positive class correctly [MMS+21; VR18].

Kusner et al. proposed another fairness definition 2017 called *Counterfactual Fairness* [KLRS17]. Formally, a "Predictor $\hat{Y}$ is counterfactually fair if under any context $X = x$ and $A = a$, $P(\hat{Y}_{A\leftarrow a}(U) = y|X = x, A = a) = P(\hat{Y}_{A\leftarrow a'}(U) = y|X = x, A = a)$, for all y and for any value a' attainable by A" [KLRS17]. In other words, a decision is considered fair if the outcome is the same in the real world and the counterfactual world in which individuals belong to different groups [KLRS17; MMS+21].

It is important to mention that according to Kleinberg et al., it is impossible to satisfy all fairness definitions simultaneously [Kle18].

Fairness Metric	Individual Fairness	Group Fairness
Equalized Odds	no	yes
Equal Opportunity	no	yes
Demographic Parity	no	yes
Conditional Statistical Parity	no	yes
Fairness through Unawareness	yes	no
Fairness through Awareness	yes	no
Treatment Equality	no	yes
Test Fairness	no	yes
Counterfactual Fairness	yes	no

Table 2.1: Categorization of Fairness Metrics according to Mehrabi et al. [MMS+21]

2.2 Relationships between Data Characteristics and Fairness Metrics

Based on the available literature in this section, we want to focus on the relationship between data characteristics such as data quality and the fairness metrics we have previously introduced. In other words, we want to investigate the influence a change in various data characteristics has on fairness metrics, i.e., the fairness of a dataset. We first review data characteristics that have been shown to affect fairness before we summarize further characteristics that are commonly considered influential for accuracy, where the relationship to fairness has not yet been studied.

First, we define and explain concrete data characteristics. Subsequently, we state which relationships to fairness metrics or fairness, in general, are provided by the currently available literature.

A vital data characteristic is *balance*. A dataset can be balanced or imbalanced and is considered imbalanced if it contains skewed class proportions [Goo23]. In other words, a dataset is balanced if it does not contain any skewed class proportions.

In general, balanced datasets seem to be fairer compared to imbalanced datasets [BR21; LGA22]. Farrand et al. showed that the fairness of a dataset deteriorates when a dataset becomes imbalanced [FMST20]. More precisely, this was shown for the metrics Demographic Parity and Equal Opportunity, whereby it appears that Equal Opportunity is more sensitive to the imbalance of a dataset [FMST20]. Li et al. stated that balancing a dataset may not change the fairness significantly or result in a greater degree of unfairness [LGA22]. This was observed for the metrics Demographic Parity and Equalized Odds [LGA22].

The *size* of a dataset is an obvious but also important characteristic. A dataset can either be small or large, even though the literature does not agree on a joint definition of when to consider a dataset to be small or large [AAA+21].

The size of a dataset has a significant impact on several fairness metrics [BR21]. Biswas et al. showed that the fairness metrics Statistical Parity, Equal Opportunity, Average Odds, and Error Rate consider a smaller dataset more unfair compared to a larger dataset [BR21]. The fairness definition Error Rate Difference was especially sensitive regarding a small dataset [BR21].

Sparsity is a data characteristic that is worth considering. If the sparsity is high, a dataset contains many fields with the value zero or null [Par22]. High sparsity can lead to severe problems in the context of ML, such as over-fitting [Par22]. Therefore, this data characteristic may play an important role in decision-making algorithms.

In the context of ML, the concrete influence of the data characteristic sparsity on the fairness of datasets has not been studied yet, according to the literature. Instead, Schafer et al. and Bennett state that statistical analysis is likely significantly biased if at least 5 % to 10 % of the values are missing [Ben01; SG02]. It is a common approach to use algorithms that predict missing values based on other values [Wan21]. Wang et al. suggest algorithms that can be used to predict missing values in order to improve fairness [Wan21]. Removing data entries containing missing values may change the data distribution significantly and lead to a greater degree of unfairness [BR21]. According to Biswas et al., this is because it is not uncommon that the values are primarily missing from one attribute [BR21]. If one deletes all entries that contain missing values, the data distribution would change significantly. Therefore, it may be better not to correct or remove these entries. The fairness metric error rate difference seems sensitive to this [BR21]. In addition, Biswas et al. state that also the fairness metrics Statistical Parity and Equal Opportunity detect a fairness deterioration after removing entries with missing values [BR21].

The *dimensionality* is a common data characteristic. It "corresponds to the number of attributes/features that exist in a dataset" [My 22]. In other words, the dimensionality refers to the number of columns in a dataset. Every dataset has a dimension n and is called an n-dimensional dataset.

No literature states the influence the data characteristic dimensionality has on fairness. The fairness of a dataset may increase as its dimensionality decreases, as dimensionality greatly complicates the dataset. Therefore, the influence of dimensionality on the fairness of datasets should be investigated in future work.

An important data characteristic is *granularity*. Granularity describes the level of detail in a dataset [C3a23]. The representation of a location could serve as an example of granularity. The location could be represented as one field or split into several fields (e.g., street, house number, postal code, city, country). In the first case, the level of granularity would be high; in the latter case, the level of granularity would be low. It is essential to mention that the higher the level of granularity, the more information can be used by a decision-making algorithm [C3a23]. A high level of granularity is detrimental to computational resources such as memory [C3a23].

Similar to the data characteristic dimensionality, there is also no information available on the influence of a change in granularity on the fairness of a dataset. This should be investigated in future work.

2.3 Existing Unfairness Detecting Systems

Some systems that aim to debug fairness preparation pipelines have already been proposed in the literature. This section provides an overview of the literature's most significant approaches and systems.

The most significant system that can be used to find operations that may have increased unfairness in a data preparation pipeline is *MLINSPECT*, which was proposed by Grafenberger et al. in 2021. MLINSPECT is a framework that allows lineage-based inspection of data preparation pipelines. It can also be used to identify data distribution bugs that may influence the fairness of the resulting model. The core idea is to extract a directed acyclic graph (DAG) from the inspected pipeline, representing the data flow with logical operations such as joins or encoders. Then, the MLINSPECT system automatically instruments the code based on the DAG to detect data distribution bugs and returns the findings to the users [GGSS21; GGSS22]. As already mentioned, the proposed system only focuses on data distribution bugs. In other words, it can detect steps in a data preparation pipeline that have led to changes in the distribution. This encompasses the distribution of certain features, especially the protected attribute, and the balance of the classes itself. When we recall our findings of Chapter 2.2, it stands out that the MLINSPECT system only tracks one aspect because other data characteristic changes indicate a decrease in fairness and are not covered by this system, such as the data characteristic size. Furthermore, it does not check whether a particular fairness metric is satisfied. This could lead to the problem of a dataset being fair, but the system detects data distribution bugs. The users may then adjust the pipeline, which may even worsen it. In addition, and in accordance with the author's opinion, the quality of feedback to the users is sometimes challenging to understand and interpret correctly [GGSS22].

Another approach was presented by Bia Huang et al. in 2020. Their *fair-DAGs* library is provided for the programming language Python and supports debugging data preparation pipelines regarding changes in the distribution with respect to protected attributes. The core idea is similar to the approach of Grafenberger et al. First, the fair-DAGs framework extracts a directed acyclic graph (DAG) by inspecting the Abstract Syntax Tree (AST) derived from the code. Then, the generated DAG gets instrumented using the inspect module. By doing so, the framework can give the users feedback and inform them whether certain modifications are present in the inspected data preparation pipeline. It is important to mention that fair-DAGs can only detect suspicious distribution changes in terms of the protected attribute and classification outcome [YHSS20]. This system only considers the mentioned changes in the distribution of the protected attribute and the classification outcome. Other changes that may lead to unfairness in the data characteristics are not monitored by fair-DAGs. Hence, this is a significant limitation of this system. In addition and identical to MLINSPECT, fair-DAGs does not check the fairness metrics to determine whether the dataset is already fair after the data preparation pipeline.

It is notable that both independently developed systems can only be used to detect changes in the distribution. As pointed out previously in Chapter 2.2, the distribution of data has an impact on the fairness of a dataset and can be highly sensitive to changes in the distribution. Besides this characteristic distribution, many other data characteristics contribute to and impact the fairness of a dataset. Those are not considered in both approaches. Therefore, a substantial limitation of both systems is that they only rely on the data characteristic distribution to identify steps in a data preparation pipeline that may have introduced additional unfairness. Hence, we aim to develop our system while keeping that limitation in mind.

3 Fairness-Debugging based on multiple Data Characteristics

This chapter presents a baseline solution to assess the fairness-related data quality of intermediate results in a data preparation pipeline. First, we provide a preliminary sketch of the system concept. Secondly, we outline the data characteristics that we consider for our solution. We overcome the limitations of the systems reviewed in Section 2.3 by considering multiple data characteristics. Lastly, we present different algorithms that are needed to realize our system.

3.1 Solution Overview

This section introduces a solution to allow data scientists to debug their data preparation pipelines with respect to fairness by monitoring multiple data characteristics.

Our solution is sketched in Figure 3.1. Data is prepared for ML through the use of a preparation pipeline. The data preparation pipeline consists of multiple procession steps. In each step, we monitor the status of the observed data characteristics. This information is then stored in a separate database. Once the data has successfully undergone all the various processing stages in the pipeline, it is ready for utilization as training data to acquire an ML model. Subsequently, relevant fairness metrics are computed using test data. If the metrics indicate the existence of unfairness, the question that our solution should tackle is: At which point in the preparation pipeline is it likely that the unintentional introduction of unfairness occurred and should be reevaluated by the users? We use the information from the monitoring metadata database to compute this. Subsequently, we return feedback to the users so they can reevaluate the steps. If users have improved the aforementioned procedural step, the overall process may be replicated to evaluate whether the modification resulted in any improvement regarding the relevant fairness metrics.

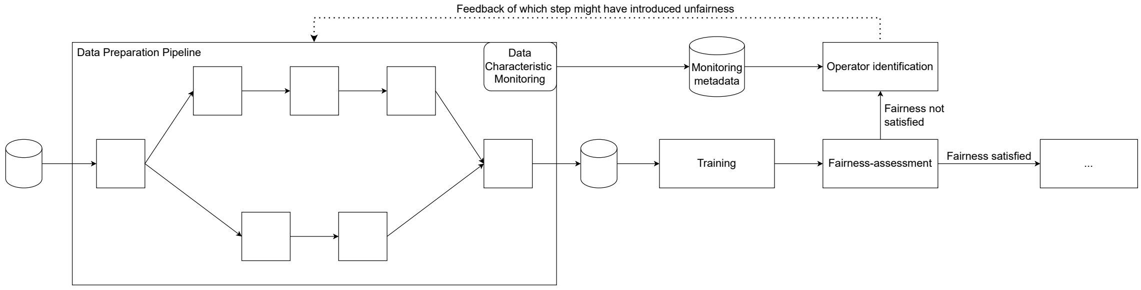

Figure 3.1: Sketch of the Solution

The basic idea of the baseline solution to tackle this question builds upon the findings of Chapter 2. The solution takes advantage of the fact that fairness correlates with specific data characteristics. We monitor the selected data characteristics during each processing step and then store the outcome of each monitoring analysis. In the event that the required fairness metric is not met, the algorithm will calculate one or more processing steps in the preparation pipeline. These steps will then be suggested to users for modification in order to improve the fairness of the dataset.

This approach differentiates significantly from the approaches of already existing fairness debugging systems, namely MLINSPECT and fair-DAGs (cf. Chapter 2.3). More accurately, we do not rely on a directed acyclic graph (DAG) but on a monitoring system that logs the status of each observed data characteristic in a database. Accordingly, this allows us to monitor multiple data characteristics not set on the distribution.

3.2 Data Caracteristic Monitoring

In this section, we present how monitoring certain data characteristics concretely works. First, we present the data characteristics we consider for our solution. Secondly, we introduce the algorithms that are used to profile these characteristics. Lastly, we discuss possible optimizations and extensions for the previously presented profiling algorithms.

3.2.1 Considered Data Characteristics

In Chapter 2.2, we provided an overview of the relationship between typical data characteristics and specific fairness metrics based on the literature. The subsequent paragraph outlines the data characteristics we monitor for this objective.

An easily measurable data characteristic that is also intuitive is the size of the used dataset. As stated in Chapter 2.2, the smaller a dataset is, the more likely it is to be biased and, therefore, unfair. In other words, if the processing step affects the dataset size, we document this and potentially use it to determine the steps responsible for introducing unfairness.

The second data characteristic we monitor is the balance of the data. From a literary perspective, it is evident that imbalanced datasets tend to exhibit unfairness. This implies that, in the current context, any changes to the balance of the dataset during a processing step are logged and may be used to determine which step has introduced unfairness.

The third and last data characteristic we take into account is sparsity. In the present context, sparsity refers to the quantity of absent values. As discussed in Chapter 2.2, it is essential to acknowledge that a model can be subject to bias and unfairness because of a high level of sparsity. To be more precise, in the event that the sparsity of the dataset is considerably augmented through a processing step, this is logged and potentially utilized for calculating which of the processes has contributed to unfairness.

3.2.2 Profiling Algorithms

This section provides profiling algorithms for each of the chosen data characteristics. Each algorithm s accompanied by a short description and an analysis regarding time and space complexity. Each algorithm is accompanied by a brief description and an analysis of its time and space complexity.

Size

Algorithm 3.1 Size Profiling

```
Input: Dataset D
size ← 0
for each row r ∈ D do
    size ← size + 1
end for
Write size into database
```

Algorithm 3.1 shown above computes the size of a dataset simply and straightforwardly. The size is determined by iterating over the entire dataset and incrementing a counting variable in each loop pass. If n describes the size of the dataset D, the time efficiency is given by $O(n)$. Since only the variable $size$ is stored and manipulated, the space complexity is determined to be $O(1)$.

Balance

Algorithm 3.2 Balance Profiling

```
Input: Dataset D, label attribute c, class1, class2
balanceClass1 ← 0
balanceClass2 ← 0
size ← 0
for each row r ∈ D do
    size ← size + 1
    if r[c] = class1 then
        balanceClass1 ← balanceClass1 + 1
    else
        balanceClass2 ← balanceClass2 + 1
    end if
end for
write balanceClass1/size and balanceClass2/size into database
```

Algorithm 3.2 calculates the balance of a dataset. The input is the dataset and the label attribute, which refers to the target variable, i.e., the classifier. For every class, the algorithm counts how often every class occurs in the dataset and divides it by the size. By doing so, the result is the balance between binary classifiers. This algorithm is only suitable for binary classifiers, i.e., the balance of datasets with multiple classes cannot be computed using this algorithm. If n describes the size of the dataset D, the time efficiency is $O(n)$. In addition, this algorithm has a space complexity of $O(1)$.

Sparsity

Algorithm 3.3 Sparsity Profiling

Input: Dataset D

$missingValues \leftarrow 0$
for each row $r \in D$ **do**
    for each value $v \in r$ **do**
        if v is missing **then**
            $missingValues \leftarrow missingValues + 1$
        end if
    end for
end for
Write $missingValues$ into database

Algorithm 3.4 determines the sparsity of a given dataset, i.e., the number of missing values that occur in a given dataset. The algorithm iterates over the entire dataset and a counter increments when a value is missing. Finally, the number of missing values gets divided by the number of total entries to get a factor between 0 and 1 that describes the degree of sparsity. If m describes the size and n the number of columns, i.e., the number of features, of the dataset D, the time efficiency is $O(m \cdot n)$. The space complexity is $O(1)$ since only two variables are used.

3.2.3 Extensions and Optimizations for Profiling Algorithms

In this section, we suggest extensions and optimizations that could be used to improve the profiling algorithms for the data characteristics size, balance, and sparsity that we introduced previously.

Contextual Optimizations

Initially, we assumed that missing values occur randomly. This means the likelihood of obtaining a missing value is the same for each attribute. As a result, the underlying distribution does not change significantly, so the fairness may not suffer significantly. Even if the literature usually models sparsity as a random process, it also states it is not unusual for missing values to correlate with a specific attribute [JWC21]. Consequently, tracking the total missing values and their distribution may be more precise. If the distribution changes significantly during a processing step in the preparation pipeline, it may indicate that fairness has suffered. Algorithm 3.4 shows how the total amount of missing values and the distribution of missing values between the attributes can be monitored.

Algorithm 3.4 Optimization of Sparsity Profiling

Input: Dataset D

$missingValues \leftarrow [0, \ldots, 0]$
$totalMissingValues \leftarrow 0$
$size \leftarrow 0$
for each row $r \in D$ **do**
  for each value $v \in d$ **do**
    if v is missing **then**
      $missingValues[indexOf(v)] \leftarrow missingValues[indexOf(v)] + 1$
    end if
  end for
end for
Write $totalMissingValues$ and $missingValues$ into database

Time Efficiency Optimizations

The system relies on multiple profiling algorithms executed at different times. Those are executed in a sequential order at the moment. Upon closer inspection of the algorithms, one can observe a notable resemblance in their structures, with certain values being computed identically. For example, Algorithm 3.2 redundantly computes the variable $size$, despite it having already been computed in Algorithm 3.1. It is clear that by doing that, significant redundancy is produced, which harms the runtime. Instead of computing the algorithms separately as standalone algorithms, they can be transformed into one algorithm. Doing that can prevent redundancy since intermediate results can be used for different measurements. We suggest an improved algorithm following this pattern in the following:

Algorithm 3.5 Optimized Profiling

Input: Dataset D, target class C

$size \leftarrow 0$
$balanceClass1 \leftarrow 0$
$balanceClass2 \leftarrow 0$
$missingValues \leftarrow 0$
for each row $r \in D$ **do**
  if $r[c] = class1$ **then**
    $balanceClass1 \leftarrow balanceClass1 + 1$
  else
    $balanceClass2 \leftarrow balanceClass2 + 1$
  end if
  $size \leftarrow size + 1$
  for each value $v \in r$ **do**
    if v is missing **then**
      $missingValues \leftarrow missingValues + 1$
    end if
  end for
end for
write $size$, $balanceClass1/size$, $balanceClass2/size$ and $missingValues$ into database

If the variable n describes the number of rows and m describes the number of columns in the given dataset, the time complexity of the overall algorithm is $O(m \cdot n)$.

Space Efficiency Optimizations

There is no room for improvement in terms of size since the space complexity of all profiling algorithms is $O(1)$. Furthermore, also the space complexity of the optimized Algorithm 3.5 is $O(1)$ and is by that ideal.

3.3 Fairness-Assessment

As explained previously, we need to assess the fairness of the resulting system to determine the next step. In this section, we focus on how this component works.

Once the training is completed and the ML model is ready for further processing, we need to assess its fairness. If the model shows sufficient fairness, it can be used by users for the intended purpose. In the case that the model cannot satisfy the fairness requirements, the operator identification step is next.

In section 2.1, we gave an overview of commonly used fairness metrics. In practice, these metrics are used to make fairness measurable. We follow this principle and choose the popular metrics, Equalized Odds, Equalized Opportunity, Demographic Parity, and Treatment Equality, to judge the fairness of the resulting ML model.

In practice, fairness metrics are often not fully satisfied. Therefore, we normalize the fairness metrics by computing their ratios based on test data. The closer the value is to 1.0, the fairer the dataset is.

Thresholds are required to determine whether the resulting ML model can be considered fair or unfair. In general, it is not possible to provide thresholds that are always correct. The choice of a suitable threshold depends on the used data and the user's needs. Therefore, we are not providing any concrete thresholds.

We provide a possible realization of the described approach in Algorithm 3.6. First, the algorithm computes the ratio for each fairness metric. Subsequently, it checks if the set thresholds are met for each computed ratio. The model is fair in this case, and the algorithm returns this information. Otherwise, it returns that the model is not fair.

Algorithm 3.6 Fairness-Assessment

Input: $model, thresholds, testData$
Output: returns whether the model is fair

Compute $EqualizedOddsRatio$, $EqualizedOpportunityRatio$, $DemographicParityRatio$, $TreatmentEqualityRatio$ on basis of $model$ and $testData$
if $EqualizedOddsRatio > thresholdEqualizedOdds$
  and $EqualizedOpportunityRatio > thresholdEqualizedOpportunity$
  and $DemographicParityRatio > thresholdDemographicParity$
  and $TreatmentEqualityRatio > thresholdTreatmentEquality$
then
  The model is fair
else
  The mode is unfair
end if

Assuming that the computation of the different fairness metric ratios is possible in $O(n)$ with n as the size of the test data, we also obtain linear runtime ($O(n)$) for the overall time complexity. From the space complexity point of view, we acquire a complexity of $O(1)$, assuming that the computation of the fairness ratios only requires constant space.

3.4 Operator Identification

As introduced in Section 3.1, if the required fairness metric is not satisfied after the training, the system should suggest one or multiple processing steps that are likely responsible for significantly worsening the fairness of the dataset. To accomplish this, it is necessary to utilize an operator identification algorithm that calculates the processing steps that require further investigation. In the following section, we discuss which different realizations we consider and how the operator identification algorithm works for our solution.

Overall, we consider three possibilities for implementing the operator identification algorithm. The challenge for these algorithms is to identify, based on the stored monitoring metadata, which procession steps have likely unintentionally introduced unfairness. We present the three approaches in the following and then provide possible realizations.

3.4.1 Comprehensive Approach

The comprehensive approach involves issuing warnings for all changes detected in the database. Hence, this approach returns a warning for each procession step that has affected at least one observed data characteristic. This means that no prioritization takes place. This approach has the advantage of returning all changes to the users, i.e., if the fairness worsens because of one of our monitored data characteristics changes, this approach captures this. It comes with the drawback that it can be challenging and time-consuming for users to interpret the result correctly and identify the step responsible for introducing unfairness.

3.4.2 Threshold-Based Approach

The threshold-based approach relies on fixed thresholds. Whenever the data characteristics of a processing step in the preparation pipeline reach a specific threshold, the procession step will be marked as improvable. The literature presents three distinct rules of thumb for the data characteristic size that can readily be converted into thresholds:

- The size of the dataset should be at least 50 to 100 times bigger than the number of prediction classes [AvC18; CLS+15; CMS12].
- The size of the dataset should be a minimum of 10 to 100 times greater than the number of features, which refers to the number of columns in the dataset. [AvC18; KM03].
- The dataset size should exceed the number of weights used in the neural network by a factor of at least 10 [AvC18].

With respect to the data characteristic balance, the literature distinguishes between different levels of severity. Google Develops [Dev23] offers the following definition:

Degree of imbalance	Proportion of Minority Class
Mild	20-40 % of the dataset
Moderate	1-20 % of minority class of the dataset
Extreme	<1 % of minority class of the dataset

In the case of data characteristic sparsity, the fairness deteriorates as the number of missing data values increases [Ben01; SG02]. More precisely, it is highly probable for statistical analysis to exhibit notable bias when there is a missing value range of at least 5 % to 10 % [Ben01; SG02].

Hence, this approach informs users about changes in the data characteristics in a step if at least one of the mentioned thresholds is met. This approach has the advantage that the result is easy for users to interpret, even for large pipelines. It also comes with the drawback that the changes in data characteristics of a procession step do not meet the requirements for any of the thresholds but are significant enough to affect the fairness of a dataset negatively. In that case, the users do not get any information about a change in data characteristics of that step and, therefore, will be unable to identify the step as the source that introduces unfairness. In addition, the choice of a suitable threshold may depend on the context.

3.4.3 Ranking-Based Approach

For the ranking-based approach, it is crucial to analyze the change in data characteristics from step S_1 to step S_2. Logging this information is important for later use. In addition to the information on how significantly a data characteristic has changed, heuristics are needed to determine which data characteristic is prioritized compared to another data characteristic. For instance, one rule-based method could prioritize as follows:

1. Dataset gets more imbalanced
2. Size of dataset gets smaller
3. The number of missing values of the dataset increases
4. Dataset gets more balanced
5. The number of missing values of the dataset decreases

One may wonder why metrics such as "Dataset gets more balanced" are also taken into account since this seems to contradict the definition we provided previously. This is because additional data must have been created to better balance the dataset. It may be that these newly created data contain malicious data that bias the dataset further. The same is true for the fifth point in the given enumeration.

In the event that the dataset does not meet a particular fairness metric after training, the system will generate feedback based on the most significant changes in data characteristics prioritized by the discussed metrics. This brings a significant disadvantage because it is challenging to rank data characteristics. Often, it is important to know not only what characteristic has changed but also how much. This disadvantage makes this approach very difficult.

3.4.4 Realizations

Generally, all three presented approaches or combinations (e.g., a combination of threshold-based and ranking-based approach) that could be used for our system. We provide the realization for the comprehensive and the threshold-based approach in the following and leave the ranking-based approach and potential combinations of the different approaches for future work.

The algorithm for the comprehensive approach is shown in Algorithm 3.7. The algorithm loops through the number of steps. This information can be handed to the algorithm as an argument or extracted from the monitored data. In each iteration, the algorithm computes the changes in the data characteristics, size, sparsity, and balance. This can be done by computing the difference between the value of the current step and the value of the previous step. Depending on which characteristics have changed, the algorithm outputs different warnings.

Algorithm 3.7 Comprehensive Approach for Operator Identification

Input: $data, numberOfSteps$
Output: step possibly responsible for reducing fairness and its reason

for $j = 1...numberOfSteps$ **do**
    $sizeMod \leftarrow data[j][size] - data[j-1][size]$
    $sparsityMod \leftarrow data[j][sparsity] - data[j-1][sparsity]$
    $balanceMod \leftarrow data[j][balance1] - data[j-1][balance1]$
    if $sizeMod < 0$ **then**
        Output that the data characteristic size has deteriorated in step j by $sizeMod$
    end if
    if $sizeMod < 0$ **then**
        Output that the data characteristic size has improved in step j by $sizeMod$
    end if
    if $sparsityMod < 0$ **then**
        Output that the data characteristic sparsity has deteriorated in step j by $sparsityMod$
    end if
    if $sparsityMod > 0$ **then**
        Output that the data characteristic sparsity has improved in step j by $sparsityMod$
    end if
    if $balanceMod < 0$ **then**
        Output that the data characteristic balance has deteriorated in step j by $balanceMod$
    end if
    if $balanceMod > 0$ **then**
        Output that the data characteristic balance has improved in step j by $balanceMod$
    end if
end for

We provide a possible realization of the threshold-based approach in Algorithm 3.8. This algorithm is similar to Algorithm 3.7 and extends it using certain thresholds. More concretely, we use thresholds to identify the characteristics of size, sparsity, and balance. It differs from Algorithm 3.7 in that it not only computes the changes of each data characteristic but its absolute value for each data characteristic. We need this information to determine whether one of the thresholds is met. In this algorithm, we only want to focus on deteriorated characteristics since these are most likely to negatively affect fairness. Therefore, we do not consider the improvement cases as in Algorithm 3.7. The algorithm outputs a warning if a data characteristic has deteriorated compared to the previous step and the required threshold is met. The specific threshold values are handed over to this algorithm as arguments and can be set arbitrarily.

Algorithm 3.8 Threshold-Based Approach for Operator Identification

Input: $data$, $numberOfSteps$, $numberOfFeatures$, $thresholds$
Output: step possibly responsible for reducing fairness and its reason

for $j = 1...numberOfSteps$ **do**
  $sizeMod \leftarrow data[j][size] - data[j-1][size]$
  $sparsityMod \leftarrow data[j][sparsity] - data[j-1][sparsity]$
  $balanceMod \leftarrow data[j][balance1] - data[j-1][balance1]$
  $size \leftarrow data[j][size]$
  $sparsity \leftarrow data[j][sparsity]$
  $balance \leftarrow data[j][balance1]$
  $totalSize \leftarrow size \cdot numberOfFeatures$
  if $sizeMod < 0$ and $size < sizeThreshold \cdot numberOfFeatures$ **then**
    Output that the data characteristic size has deteriorated in step j by $sizeMod$
  end if
  if $sparsityMod < 0$ and $sparsity > sparsityThreshold \cdot totalSize$ **then**
    Output that the data characteristic sparsity has deteriorated in step j by $sparsityMod$
  end if
  if $balanceMod < 0$ and $(balance1 < balanceThreshold$
   or $balance1 < 1 - balanceThreshold)$
  then
    Output that the data characteristic balance has deteriorated in step j by $balanceMod$
  end if
end for

Algorithm 3.7 and Algorithm 3.8 have the same structure. They iterate through the `for`-loop according to the variable $numberOfSteps$. Assuming that data can be accessed in constant time, only operations and checks are inside the loop with time complexity of $O(1)$. If the number n describes the number of steps our pipeline has, the time complexity of both algorithms is $O(n)$. Since we only store values in variables and are not using any other data structure, we have a space complexity of $O(1)$ for both algorithms.

4 Experiment

In this chapter, we conduct an experiment, which is divided into three segments, to analyze the performance of the solution we introduced in the previous chapters regarding fairness. For this, we use different benchmark datasets. First, we outline the design of the experiment. Then, we execute the experiment. Lastly, we provide the key findings of the experiment.

4.1 Design

In this section, we give an overview of the design of this experiment.

All experiments that we conduct in this chapter have the same design. We are using the programming language Python as it comes with many libraries that are useful for our purposes.

In total, we consider five different datasets from the domains of finance, criminology, and healthcare with various data characteristics to widely analyze the performance of our system. We provide details of the individual datasets in the upcoming Section 4.2.

Before the experiments can start, we have to prepare the datasets and set up the system. More concretely, we need to identify the protected attribute and define arguments that are needed for some pipelines. To obtain more reliable results, we repeat the whole procedure iteratively. Specifcally, we set the number of iterations to 50 for the upcoming experiments. To make the results of the experiments more comparable, we apply the same data preparation pipelines to each dataset. Therefore, we sometimes have to manipulate the datasets beforehand to obtain a dataset we can use. We describe further details in the following Section 4.2 and the repository.

The different pipelines we introduce later are implemented using the `sklearn` module because this provides an easy way to implement pipelines. We define a fundamental pipeline to precisely monitor the impacts of specific steps in the data preparation pipeline. All other pipelines extend this pipeline by at least one additional step. Hence, we can detect precisely how much the fairness has increased or decreased by these additional steps. For further details regarding the implementation of the pipelines, one should consult our repository [Epp23].

After executing the data preparation pipeline, we apply a classification algorithm on the dataset we obtain. To validate how well the resulting model performs in terms of fairness, we split the dataset into a training and test dataset beforehand. More precisely, we split the dataset randomly, whereby 80 % is used as training data and 20 % as test data. We are using the logistic regression classifier as our binary classification algorithm. Conveniently, the module `sklearn` provides an implementation of this model. First, we train the model with the training data. Then, we use the test data to obtain the model's predicted outcomes.

To measure the impact of the steps in the data preparation pipeline, we measure fairness using the fairness metrics we introduced in Chapter 2.1 and specified in Section 3.3. To make these metrics comparable, we have to normalize them. More concretely, we use the ratio of the individual metrics to achieve this. To determine this, we compute the minimum between metric and group for each metric and take the minimum of it [BDE+20]. For more details, one should take a look at the documentation of the module `fairlearn` that we used in our implementation [BDE+20]. It holds: The closer the computed number is to 1, the fairer the dataset is according to the fairness metric.

For this evaluation, we set the threshold of the fairness assessment to 0. Hence, this ensures that the operation identification algorithm is always executed. We decide to do so because the choice of thresholds is user-dependent and depends strongly on the user's needs. Moreover, we want to focus on evaluating how well our approach can detect operators introducing unfairness. Therefore, we need to analyze the changes in the data characteristics size, balance, and sparsity.

As already mentioned, our experiment is divided into three different parts. In the first part, we examine the impact on the fairness of data preparation pipelines, which consist solely of fundamental and simple steps. In Part II, we focus on more advanced pipelines. We use the comprehensive approach for the operator identification algorithm in Part I and Part II. In Part III, we use the threshold-based approach for operator identification to analyze which approach is most reliable and user-friendly.

4.2 Datasets

As mentioned earlier, we use different benchmark datasets frequently used in the literature to assess the performance of our introduced system. Specifically, we use six datasets with different characteristics and contexts. In the following, we concisely introduce the different datasets and present their essential characteristics.

German Credit Dataset

The German Credit Dataset, published in 1994 by Prof. Hans Hofmann, is one of the most popular datasets regarding Fairness in ML and has been used in various studies [Hof94; LRI+22]. It contains information about bank account holders in Germany and classifies whether a person is a good or bad credit risk [FMSS22; Hof94]. The German Credit Dataset encompasses 1 000 entries and contains no missing values. Hence, this dataset is very small compared to many other benchmark datasets. Each instance is expressed by 13 nominal, numerical, and binary features.

In the literature, the features *sex* and *age* are considered as protected attributes [LRI+22]. To be in accordance with most studies, we consider *sex* as the protected attribute. Since the feature *sex* is not directly reflected in the data, we must extract this information from the feature *personal_status_and_sex*. Therefore, we have to add another feature *sex* to our dataset that uses the information of the feature *personal_status_and_sex* to encode the gender of every individual correctly. It is important to mention that the ratio between these groups is 69 % (*male*) to 31 % (*female*). In other words, the dataset is clearly dominated by male bank account holders. Overall, the dataset is imbalanced with a ratio of 2.3 (positive) to 1 (negative) towards the positive class.

Adult Census Dataset

The Adult Census Dataset [BK96] by Ronny Kohavi and Barry Becker was published in 1996. The data were retrieved from the Census Database, hosted by the US Census Bureau [BK96]. The dataset provides information about adult people in the USA [BK96]. It predicts whether an individual's income is less or equal to 50 000 US dollars or greater than 50 000 US dollars. In total, the dataset encompasses 48 842 instances that are described by 15 attributes. Those features are either nominal, numerical, or binary. Furthermore, this dataset contains missing values. More precisely, the feature *workclass* contains 5.73 % (2 799) missing values, the feature *occupation* contains 5,75 % (2 809) missing values, and the feature *native-country* contains 1.75 % (857) missing values. Overall, 7.41 % (3 620) of entries are affected.

Multiple features could serve as protected attributes. In prior studies, the features *sex*, *age*, and *race* have been used as protected attributes [LRI+22]. Correspondingly to most authors, we consider the feature *race* as our protected attribute. Various studies agree that this nominal attribute should be simplified by turning it into a binary attribute with the values *white* and *non-white* [LRI+22]. In other words, the values *black*, *asian-pac-islander*, *amer-indian-eskimo* and *other* are mapped to the value *non-white*. In contrast, the entries with the value *white* remain untouched. Neither group is represented equally in the dataset. More concretely, the ratio between the two groups is 86 % (*white*) to 14 % *non-white*. Hence, the dataset is clearly dominated by the group *white*. In addition to that, the class itself is also skewed. More precisely, the ratio between the two classes is 1 (positive) to 3 (negative), whereby we consider the class as positive if an individual makes more than 50 000 US dollars and negative if an individual makes less or equal to 50 000 US dollars.

COMPAS Dataset

The COMPAS Dataset [Pro24] is another very popular dataset with a real-world application in the justice domain released in 2016 by ProPublica [JA16; LRI+22]. In contrast to other datasets commonly used in the literature, this is relatively recent. More precisely, the data was collected by the Broward County Sheriff's Office in Florida between January 2013 and December 2014 [JA16]. This resulted in the datasets *risk of recidivism* and *Risk of violent recidivism*. The *risk of recidivism* dataset attempts to predict whether an individual will be rearrested within two years after being released [LRI+22]. The dataset *risk of violent recidivism* has a slightly different classification. It predicts whether an individual gets rearrested within two years after being released due to a violent crime [LRI+22]. We decide to use the second dataset *risk of recidivism* because it contains 52 % more data entries. Initially, this dataset consists of 72 14 instances and 52 nominal, binary, and numerical attributes. It stands out that most data rows are affected by missing values. More precisely, 88.6 % (6 395) of the entries contain at least one missing value. In addition, the feature *violent_recid* has only *NULL* values. Accordingly, we dropped this column. As already mentioned previously, we want to apply the same data preparation pipelines to all datasets. Thus, we execute the necessary dropping of this column outside of a data preparation pipeline.

In prior studies, the features *race* or *sex* were typically chosen as the protected attributes. To be in accordance with most other studies, we choose *race* as the protected attribute [LRI+22]. We simplify this attribute to binary, only containing the values *white* and *Non-White*. This means that we map the values *african-american*, *asian*, *Hispanic*, *native american* and *other* to the value *Non-White* while we do not change any entries with the value *white*. The ratio between both groups is 1 (*white*) to 1.9 (*Non-White*). Thereby, the COMPAS Dataset is dominated by people who are not white. As mentioned above, this dataset aims to predict if an individual gets rearrested within two years after being released [LRI+22]. Different from many other datasets, this dataset is fairly balanced with a ratio of 1 (positive) to 1.2 (negative) [LRI+22].

Credit Card Clients Dataset

The Credit Card Clients Dataset [YL09] was collected in October 2005 in Taiwan [YL09]. It captures information about credit card clients in Taiwan and classifies whether a person is credible [Yeh16; YL09]. In total, the Credit Card Clients Dataset encompasses 30 000 instances described by 25 features with the data types nominal, numerical, and binary [LRI+22; Yeh16; YL09]. In contrast to some of the other datasets we introduce in this chapter, this dataset odes not contain missing values.

Typically, the feature *sex* is considered the protected attribute in the literature [LRI+22]. However, also the features *education* and *marriage* were used in some studies [LRI+22]. We choose the feature *sex* as the protected attribute to align with most studies that have used this dataset. The ratio between the two groups *male* and *female* is 1 (*male*) to 1.5 (*female*). Therefore, the dataset is slightly dominated by female people. The attribute *default payment* classifies whether a credit card holder is credible, whereby we consider the case of being credible as positive. The ratio between these two classes is 1 (positive) to 3.5 (negative). Thus, this is an imbalanced dataset.

Diabetes Dataset

The Diabetes Dataset [SDG+14] was collected over ten years between 1999 and 2008 [LRI+22; SDG+14]. It represents the data of patients at 130 hospitals and integrated delivery networks in the USA who were diagnosed with diabetes, stayed at the hospital between one and 14 days, underwent laboratory diagnostics and were administered medication during their stay [SDG+14]. This dataset aims to predict whether the patients will be readmitted within 30 days after discharge [SDG+14]. Overall, the dataset contains 10 1766 instances and 47 features that are either numerical or nominal. It stands out that this dataset contains missing values. More precisely, the feature *race* contains 2 % (204), the feature *weight* contains 97 % (98 713), the feature *payer code* contains 52 % (52 918), the feature *medical specialty* contains 53 % (53 935) and the feature *diagnosis 3* contains 1 % (102) of missing values [SDG+14]. Furthermore, the label attribute *readmitted* does not contain a correct classification for 54864 entries [LRI+22]. Therefore, we removed these entries to ensure that there is no doubt about the correct classification of an individual. As a consequence, our dataset shrunk by over 50 % to a size of only 46 902 data entries. Identically to the COMPAS dataset, we have to remove entries here outside of a data preparation pipeline so that we are able to apply the same pipelines for each considered dataset.

In prior studies, the feature *gender* was chosen as protected attribute [LRI+22]. Therefore, we also choose this as the protected attribute. The two groups *male* and *female* are distributed with the ratio 1 (male) to 1.2 (female). Thus, the two groups are almost reflected equally in the dataset. The class label attribute of the Diabetes Dataset is *readmitted*, which determines whether the patients will be readmitted within 30 days after discharge. We consider the case that the patients will be readmitted within 30 days after discharge as the positive class. The classes are distributed with a ratio of 1 (positive) to 3.1 (negative), which means this dataset is imbalanced.

4.3 Basic Pipeline

We define the steps of the Basic Pipeline as follows:

1. Imputing: Imputes all missing values with a very basic strategy based on the data type of the specific column
2. Encoding: Encodes non-numerical values to numerical values
3. Scaling: Scales the numerical values to a range between -1 and 1

As one can see above, the Basic Pipeline is straightforward. It only consists of Encoding, Scaling, and Imputing. We obtain an encoded and scaled dataset free of missing values by performing these steps. This is often an essential requirement for many data transformations and ML algorithms. This is a standard setup for more advanced algorithms in data preparation pipelines, which makes it a fundamental pipeline. Therefore, we chose this setup for our Basic Pipeline. All other pipelines we present in this work will extend this Basic Pipeline. Moreover, for each upcoming pipeline, we refer to this pipeline and compare the measured fairness metrics. By doing so, we are able to monitor exactly which impact a certain step has on the fairness of a dataset.

The two steps, Encoding and Scaling, are performed in a standardized and well-known manner. Hence, the fairness of the dataset should remain untouched by these procession steps. Contrarily, the imputation of missing values in the dataset may be critical. This is because missing values are imputed based on the existing values in the corresponding column. We chose a simple strategy for computing missing values: If the data type is numeric, the median is used to determine a missing value. Otherwise, the value with the most frequent occurrence is taken. Depending on the missing values that have to be computed, this step could significantly increase the unfairness of the dataset. The more values that have to be imputed, the more significant these changes could be in affecting the fairness of the dataset. More precisely, by imputing missing values, the distribution within a feature can become highly distorted by performing this step, possibly leading to a more unfair model. It is essential to mention that this applies only to datasets that contain missing values. If the dataset contains no missing values, then the imputation step will not change the unfairness.

For these datasets that contain missing values, we expect a warning that informs users that the number of missing values has decreased in step one. In addition, the system should also output how many missing values have been imputed and which proportion this is compared to the total number of entries. We do not expect the system to output any warning for datasets that do not contain any missing values.

For the datasets Census Income Dataset, COMPAS Dataset, and Diabetes Dataset, the system output a warning regarding the decrease of missing values. This is the expected behavior since those are the only datasets with missing values. Out of all these datasets, the COMPAS Dataset imputed the most values by far. More precisely, the number of missing values decreased by 63 699. In other words, 63 699 values were imputed. Considering that this dataset only contains 7 214 data points, it becomes clear that the imputation step may have increased the unfairness. On the other hand, the decrease of missing values is 4 262 for the Census Income Dataset and 1 267 for the Diabetes Dataset. This is more reasonable, considering the datasets contain 48 842 and 54 864 instances, respectively. The system did not alert the users for any other datasets. Therefore, the results fully met our expectations.

The results of the measured fairness metrics are listed in Table 4.1. Notably, the Diabetes Dataset appears to be the fairest dataset after performing the processing steps in the pipeline as it achieved the highest scores in all fairness metrics. Remarkably, it seems that the fairness metric Demographic Parity score is always significantly higher than all other fairness metrics. This could mean it is easier to satisfy Demographic Parity than other fairness metrics. In addition, the fairness metric Treatment Equality seems to be the hardest metric to satisfy since none of our datasets is able to satisfy it entirely.

	Equalized Odds	Equalized Opportunity	Demographic Parity	Treatment Equality
German Credit	0.72	0.87	0.94	0.59
Census Income	0.34	0.45	0.80	0.34
COMPAS	0.40	0.82	0.99	0.21
Credit Card Clients	0.56	0.67	0.82	0.56
Diabetes	1.00	1.00	1.00	0.83

Table 4.1: Fairness Metrics of the Basic Pipeline

4.4 Part I: Foundational Fairness Analysis

In the first part of the experiment, we test how our developed system performs on simple and intuitive pipelines. More precisely, we define data preparation pipelines that contain steps that modify specific data characteristics in a straight and clear manner. For this purpose, we always use the Basic Pipeline as a basis and extend this pipeline. This has the advantage of comparing the measured fairness metrics to the ones we obtained after executing the Basic Pipeline. Hence, we obtain a meaningful result regarding whether the additional steps negatively influenced the corresponding dataset's fairness. Furthermore, we do not discuss the warnings that the system returned because of the first three steps since these were already discussed in Section 4.3. In the first step, we define the pipelines and name the expected results. Secondly, we apply our system with the specified pipelines and record the output, which are warnings, as well as the fairness metrics. Finally, we analyze the results based on our expectations.

As mentioned, each of the following pipelines extends the Basic Pipeline we defined previously. In total, we are taking three pipelines into account in the following.

4.4.1 Pipeline 1

We define the steps of Pipeline 1 as follows:

1. Imputing: Imputes all missing values with a very basic strategy based on the data type of the specific column
2. Encoding: Encodes non-numerical values to numerical values
3. Scaling: Scales the numerical values to a range between -1 and 1
4. Balancing by Deleting: Deletes Instances of the overrepresented class to balance the dataset

Pipeline 1 extends our Basic Pipeline by the step of Balancing by Deleting. This transformation step aims to ensure the balance of the classes by deleting the overrepresented class. In our implementation, we first identify by which factor the dominating class is overrepresented in the dataset and then delete entries so that both classes are equally reflected. If a dataset is not initially equally balanced, this should lead to multiple effects. First, the balance should improve since we delete as many entries as needed to balance the dataset fully. Secondly, the size of the dataset should decrease. Since size is an essential parameter for the fairness of a dataset, it could affect the fairness if a significant part of the dataset gets deleted. In contrast to the other pipelines we have discussed previously, this affects not only datasets that contain missing values but also all datasets that are not fully balanced.

In addition to the warnings caused by the step Imputing, we expect warnings regarding the step Balancing by Deleting. Since none of the datasets considered are fully balanced, we expect warnings for all of them. For each of the datasets, our system should output that, on the one hand, the balance has increased; on the other hand, it should inform the users that the size of the dataset has shrunk. It should also state how many entries were affected and which portion this is in relation to the total dataset.

For all our datasets, the system output the expected warnings regarding the improvement of the balance and the decrease in the size. Therefore, the system fulfilled our expectations and worked as it was supposed.

The system informs the users that the size of the German Credit Dataset shrunk by 40 % (400 instances), meaning that 40 % of the entries were deleted. In addition, the balance improved by a factor of 0.2. Despite the improvement in balance, the significant decrease in size indicates that unfairness may have increased during this step. If one compares the outcome of the fairness metrics shown in Table 4.2 to the ones we obtained after executing the Basic Pipeline (Table 4.1), we notice that this presumption is correct for the most part based on the fairness metrics. The metrics Equalized Odds (from 0.72 to 0.63), Equalized Opportunity (from 0.87 to 0.85), and Treatment Equality (from 0.59 to 0.50) decreased while the metric Demographic Parity (from 0.94 to 0.94) remained the same. Hence, the intuitive interpretation of the warning output by the system was correct. Overall, our system behaved as expected and simplified the assessment critical step Balancing By Deleting.

The system returned a warning that the size of the Census Income Dataset has decreased by 35 % (16 879 instances) in step 4. Similar to the effects on the German Credit Dataset, this is a significant decrease in size. In addition, the system informs the users that the balance has improved by a factor of 0.25. This is a significant improvement that may benefit the fairness of the dataset. Therefore, there are two different interpretations of these warnings. First, it could be that the fairness has deteriorated because a large amount of data entries were deleted. Secondly, fairness could also be increased since the dataset's balance has improved significantly. If we compare the fairness metrics of the Basic Pipeline to this pipeline, it turns out that the fairness was not affected negatively. More precisely, the metrics Equalized Odds (from 0.34 to 0.35), Equalized Opportunity (from 0.45 to 0.46), Demographic Parity (from 0.80 to 0.81), and Treatment Equality (from 0.34 to 0.35) have improved by a little. Hence, the step Balancing by Deleting did not introduce additional unfairness. However, the system returned the warnings it was supposed to output and, therefore, worked as expected.

According to the system, the size of the COMPAS Dataset has shrunk by 10 % (712 instances), and the balance of it has improved by a factor of 0.05. Since only 10 % of the dataset was deleted, it is not likely that the fairness of the dataset suffered by a lot. If one takes a look at the measured fairness metrics in Table 4.2 and compares these to the measured fairness metrics of the Basic Pipeline (Table 4.1), it becomes clear that this assessment is mostly correct. The metrics Equalized Odds (from 0.40 to 0.44), Demographic Parity (0.99 to 0.99), and Treatment Equality (from 0.21 to 0.26) show clearly that the fairness did not decline. Only the metric Equalized Opportunity (from 0.82 to 0.80) declined slightly. Overall, the system output the warnings we expected. Compared to the two previous datasets, it was easier to interpret the system's results correctly.

Our system detected a decrease in the size of the Credit Card Clients Dataset by 55 % (16 728 instances) and an improvement of the balance of the dataset by a factor of 0.28. This is a significant decrease in size and a significant improvement in balance. Therefore, it is very challenging to interpret the impacts on the fairness of this procession step. Compared to the metrics of the Basic Pipeline, all fairness metrics have improved. More precisely, the metrics Equalized Odds (from 0.56 to 0.57), Equalized Opportunity (from 0.67 to 0.68), Demographic Parity (from 0.82 to 0.84), and Treatment Equality (from 0.56 to 0.58) have improved. This indicates that no unfairness was additionally introduced in this step. We can conclude that the system worked as expected since it output the correct warnings.

For the Diabetes Dataset, we again obtain two warnings: On the one hand, the system informs that the size has worsened by 52 % (24 188 instances); on the other hand, it returns the information that the balance has improved by a factor of 0.26. Because of these two significant changes in size and balance, it is very challenging to interpret these warnings correctly. If we compute the fairness metrics, we can see that the fairness metrics for Equalized Odds, Equalized Opportunity, and Demographic Parity are already perfect since the score evaluates to 1.0. Only Treatment Equality has a score of 0.80 and not an ideal score of 1. If we compare these results to the fairness metrics of the base pipeline, we can see that Treatment Equality has deteriorated from 0.83 to 0.80. Since the other fairness metrics are still ideal, we consider that dataset fair. Therefore, the step Balancing by Deleting did not introduce unfairness in this dataset. Even though no unfairness was introduced, our system worked as it was supposed to because many changes in the data characteristics occurred.

Overall, our system output all the warnings we expected. This means it can always detect the critical steps that users should inspect. It is important to mention that interpreting the system's output can be very challenging. This happens especially if there are huge changes in data characteristics that indicate different impacts on fairness.

	Equalized Odds	Equalized Opportunity	Demographic Parity	Treatment Equality
German Credit	0.63	0.85	0.94	0.50
Census Income	0.35	0.46	0.81	0.35
COMPAS	0.44	0.80	0.99	0.26
Credit Card Clients	0.57	0.68	0.84	0.58
Diabetes	1.0	1.0	1.0	0.80

Table 4.2: Fairness Metrics of Pipeline 1

4.4.2 Pipeline 2

We define the steps of Pipeline 2 as follows:

1. Imputing: Imputes all missing values with a very basic strategy based on the data type of the specific column
2. Encoding: Encodes non-numerical values to numerical values
3. Scaling: Scales the numerical values to a range between -1 and 1
4. Upsampling: Sample data up in order to make it balanced

Pipeline 2 extends our Basic Pipeline through step Upsampling. Similar to the Basic Pipeline, this pipeline aims to balance the dataset. In contrast, it does not balance it by deleting entries of the overrepresented class but by upsampling. In other words, additional entries of the underrepresented class are created to achieve an equal balance between both classes. In our implementation, we first identify the gap between the overrepresented and underrepresented classes. In the next step, we randomly choose samples of the underrepresented class and add them to the dataset until both classes are reflected equally in the dataset. This has two effects for not fully balanced datasets: First, the balance improves so that both classes are reflected equally in the dataset. Secondly, the size of the dataset should increase since we are adding additional instances to it.

Besides the warnings we expect for step 1, already described and discussed in the section for the Basic Pipeline, we expect multiple warnings for all imbalanced datasets due to the step Upsampling. Since none of our datasets are fully balanced, we expect warnings for all of them. More precisely, we expect a warning that informs users about improving the balance of the classes and another warning that informs one about the size increase. The results of the measured fairness metrics are depicted in Table 4.3

For the German Credit Dataset, our system outputs the warning that, on the one hand, the balance of the dataset has improved by the factor of 0.2, and on the other hand, the size has increased by 40 % (400 instances). As discussed in Chapter 2.2, both data characteristics, size, and balance, play a crucial role in fairness. Hence, it appears that based on these warnings, fairness must not worsen but even improve. According to the fairness metrics, this is not true. If one compares the results of the fairness metrics of the Basic Pipeline to this pipeline, it turns out that all metrics except Treatment Equality have significantly deteriorated. More precisely, the metrics Equalized Odds (from 0.72 to 0.62), Equalized Opportunity (from 0.87 to 0.76), Demographic Parity (from 0.94 to 0.89), and

Treatment Equality (from 0.59 to 0.58) deteriorated significantly. This result is surprising since the observed data characteristics were modified in a way that indicates an improvement in fairness. The deterioration in fairness could be due to the small size of this dataset and the relatively high amount of instances that need to be generated (10 %). Hence, it can be challenging to interpret the output of our system correctly. However, the system worked as expected and tracked the data characteristics correctly.

According to our system, the balance of the Census Income Dataset has improved by a factor of 0.23, and the size of this dataset expanded by 34.6 % (16 879 instances). This may indicate that the fairness of this dataset has improved since both data characteristics have changed in the direction that is considered to be fairer. Indeed, all fairness metrics we are computing show an improvement compared to the results for fairness metrics of the Basic Pipeline. More concretely, the metrics Equalized Odds (from 0.34 to 0.40), Equalized Opportunity (from 0.45 to 0.57), Demographic Parity (from 0.80 to 0.83), and Treatment Equality (from 0.34 to 0.40) have significantly improved. This clearly shows that the step Upsampling has improved the fairness of the Census Income Dataset. Our system worked as expected by outputting the changes in the data characteristics in this step. Therefore, it worked well as a tool to assess the changes in fairness based on changes in the data characteristics.

For the COMPAS Dataset, we obtained the warnings that the balance has improved by a factor of 0.05, and its size has increased by 9.9 % (712 instances). Compared to the two other datasets we have discussed previously, it stands out that relatively few samples had to be generated to balance both classes equally. This indicates that fairness has rather improved than deteriorated. This assumption is confirmed if one looks at the measured fairness metrics and compares them to the measured fairness metrics of the Basic Pipeline. More concretely, while the fairness metrics Equalized Odds (from 0.40 to 0.44) and Treatment Equality (from 0.21 to 0.26) improved significantly, the fairness metrics Equalized Opportunity (from 0.82 to 0.80) and Demographic Parity (from 0.99 to 0.99) remained roughly the same. Therefore, the intuitive interpretation was correct. However, our system worked as expected.

Our system reported an improvement of the balance by a factor of 0.28 and an increase of the size of the Credit Card Dataset by 55.76 % (16 728 instances). This is a high number of generated new samples. Therefore, it is challenging for users to interpret the warnings correctly. On the one hand, it may not affect fairness negatively since balance and size changed towards the direction that benefits the fairness of a dataset. On the other hand, and similar to what we observed for the German Credit Dataset, the fairness could deteriorate since some entries could be overrepresented. The results show that the first interpretation is correct. None of the fairness parameters has deteriorated compared to the results of the fairness metrics of the Basic Pipeline. On the contrary, we can observe significant improvement for each metric. More concretely, the fairness metrics Equalized Odds (from 0.56 to 0.63), Equalized Opportunity (from 0.67 to 0.75), Demographic Parity (from 0.82 to 0.86), and Treatment Equality (from 0.56 to 0.63) show a significant increase. This example shows clearly that sometimes it is difficult to assess the information we obtain from the tool correctly. Therefore, the system was still performing correctly and as we expected.

The result for the Diabetes Dataset is similar to the German Credit Dataset and the Credit Card Clients dataset. For these, the system informs users about a significant improvement in balance and size. More precisely, the balance has improved by a factor of 0.26, and the size of the Diabetes Dataset has increased by 51.6 % (24 188 instances). There are two possible interpretations. First, the fairness did not worsen and may have even increased due to the fact that a highly balanced and bigger dataset has a positive impact on fairness, in general. Second, fairness could indeed deteriorate because a relatively high number of samples is generated, which can negatively affect fairness. When one compares the results of the Basic Pipeline with the fairness metrics we measured here, it becomes clear that the second presumption is correct. More precisely, the fairness metrics Equalized Odds (from 1.00 to 0.93), Equalized Opportunity (from 1.00 to 0.94), and Demographic Parity (from 1.00 to 0.94) decreased significantly. An exception is only the fairness metric Treatment Equality. This metric improved from 0.83 to 0.89. However, the measured values are still excellent, indicating that this dataset is fair with respect to the protected attribute. Overall, our system was also able to output warnings here to help users identify this step as a potential source of introduced unfairness, even though it was not easy to interpret the result.

In summary, for this pipeline, we found that our introduced system worked as expected and could be used to detect sources of unfairness effectively. For some datasets, it was challenging to interpret the tool's results correctly.

	Equalized Odds	Equalized Opportunity	Demographic Parity	Treatment Equality
German Credit	0.62	0.76	0.89	0.58
Census Income	0.40	0.57	0.83	0.40
COMPAS	0.45	0.84	1.0	0.16
Credit Card Clients	0.63	0.75	0.86	0.63
Diabetes	0.93	0.94	0.94	0.89

Table 4.3: Fairness Metrics of Pipeline 2

4.4.3 Pipeline 3

We define the steps of Pipeline 3 as follows:

1. Imputing: Imputes all missing values with a very basic strategy based on the data type of the specific column
2. Encoding: Encodes non-numerical values to numerical values
3. Scaling: Scales the numerical values to a range between -1 and 1
4. Halving: The first half of the dataset gets eliminated

As seen above, Pipeline 3 consists of four subsequent process steps. By doing this, it extends our Basic Pipeline by the additional step Halving. Since the steps Imputing, Encoding, and Scaling were already covered in the analysis of the Basic Pipeline, we only focus on the new step, Halving. As the name already says, this step divides the dataset into two equal big datasets and only keeps one. In our concrete implementation, we keep the first half of the dataset. The idea of this approach is to reduce the size of the dataset. In practice, datasets can be huge, and some systems or parts of systems may be unable to cope with giant datasets. Therefore, a method is needed to reduce the amount of data. The Halving step is a very basic and straightforward approach to significantly reducing a dataset's size. Of course, this entails various effects on the characteristics of our observed data. The primary and obvious effect is the significant decrease in size. Besides this, the balance of the dataset is likely modified. The degree to which it deteriorates or enhances, and the specific factor by which this occurs, is entirely determined by the number of instances for each class in the first half of the dataset. This is generally not predictable since each dataset has a different distribution.

First, we expect a warning that informs users that the size of the dataset has decreased by 50 %. Secondly, we expect a warning that informs whether the balance of the dataset has improved or worsened and by which factor. The results of the measured fairness metrics are depicted in Table 4.4

For the German Credit Dataset, the system output two warnings. First, it informed users that the size of the dataset has decreased by 50 % (500 instances). Secondly, it states that the balance of this dataset has deteriorated by a factor of 0.03. Therefore, the system output both of the expected warnings. This information indicates that the fairness may have deteriorated. The comparison of the fairness metrics of this pipeline, outlined in Table 4.4, with the metrics of the Basic Pipeline (Table 4.2), validates this presumption. The fairness metrics Equalized Odds (from 0.72 to 0.66), Demographic Parity (from 0.94 to 0.93), and Treatment Equality (from 0.59 to 0.52) have deteriorated significantly. Only the metric Equalized Opportunity (from 0.87 to 0.87) did not worsen. Hence, the fairness metrics confirmed what we conjectured based on the returned warnings of the system. Therefore, the system detected the step Halving as critical and output the expected warnings. Hence, it was able to help users assess the impact of this step on the fairness of the German Credit Dataset.

Also, for the Census Income Dataset, we obtained two warnings. The first warning was regarding the decrease in size by 50 % (16 281 instances). The second warning was concerning balance, which increased by a factor of 0.05. This entails multiple interpretations. First, the overall fairness may have suffered since fairness is tightly connected to the data characteristic size. On the other hand, the balance has improved, which may indicate that the fairness has improved. However, it turns out that the second presumption is correct. If we compare the results of the measured fairness metrics of this pipeline with the measured fairness metrics of the Basic Pipeline, we can detect that each metric has improved. More precisely, the fairness metrics Equalized Odds (from 0.34 to 0.36), Equalized Opportunity (from 0.45 to 0.52), Demographic Parity (from 0.80 to 0.88) to Treatment Equality (from 0.34 to 0.36) have all significantly improved. Overall, the system output the expected warnings and was able to ease the assessment of this critical step.

Once again, the system reported a decrease in size of 50 % (3 607 instances) for the COMPAS Dataset. In contrast to the German Credit Dataset and the Census Income Dataset, the balance was affected only very little. More precisely, the balance of the COMPAS Dataset worsened by a factor of 0.006. Hence, the step Halving did not significantly impact the balance of the dataset. Therefore, this may indicate that the fairness of the dataset did not improve but rather deteriorated. The measured fairness metrics confirm this, for the most part. More concretely, the fairness metrics Equalized Odds (from 0.40 to 0.32) and Treatment Equality (from 0.21 to 0.13) deteriorated significantly. In contrast, the metrics Equalized Opportunity (from 0.82 to 0.83) and Demographic Parity (from 0.99 to 0.99) did not change significantly. This shows that with the aid of the system, it was straightforward to assess the impact of the step Halving on fairness correctly. In addition, the system output the warnings we expected.

For the Credit Card Clients Dataset, the system informed users about a size decrease by 50 % (1 500 instances). In addition, it also returned a warning stating that this dataset's balance has improved by 0.004. Similar to the balance change of the COMPAS Dataset, this change in balance is very minor and, therefore, not likely to affect the fairness of this dataset significantly. The considerable decrease in size may negatively impact the fairness of the Credit Card Clients Dataset. On the contrary, compared to the fairness metrics of the Basic Pipeline, it turns out that this presumption is incorrect. More concretely, the fairness metrics Equalized Odds (from 0.56 to 0.61), Equalized Opportunity (from 0.67 to 0.74), Demographic Parity (from 0.82 to 0.87), and Treatment Equality (from 0.56 to 0.61) have significantly improved. This shows once again that it can be challenging to interpret the system's results correctly. However, the system output the correct warnings and worked as expected overall.

According to the system, the Diabetes Dataset significantly decreased in size by 50 % (23 451 instances) and barely improved the balance by a factor of 0.006. The significant decrease in size and only a tiny improvement in the balance indicates that the fairness rather worsened than improved. This presumption appears correct if one compares the values of the measures fairness metrics of this pipeline to these of the Basic Pipeline. More precisely, the metrics Equalized Odds, Equalized Opportunity, and Demographic Parity did not change at all and still have the value of 1.00. On the contrary, the fairness metric Treatment Equality deteriorated significantly from 0.83 to 0.76. This shows that the system could help users assess the impact of the step Halving correctly. The system also output the correct warnings and, therefore, worked as expected.

In total, the system was always able to output the correct warnings. However, we also recognized that it can be challenging to interpret the system's warnings correctly. There, it may be necessary to assess not only the system by the obtained warnings but also the structure of the dataset. Furthermore, judging datasets individually and not generalizing the effects of a specific warning is essential.

	Equalized Odds	Equalized Opportunity	Demographic Parity	Treatment Equality
German Credit	0.66	0.87	0.93	0.52
Census Income	0.36	0.52	0.88	0.36
COMPAS	0.32	0.83	0.99	0.13
Credit Card Clients	0.61	0.74	0.87	0.61
Diabetes	1.00	1.00	1.00	0.76

Table 4.4: Fairness Metrics of Pipeline 3

4.5 Part II: Advanced Fairness Analysis

The second part of this experiment is similar to the first. While the first part was focused on simple and intuitive pipelines to test if the system works as expected, the pipelines presented in this part take it beyond that. In other words, we want to try out more complex pipelines on our system and then see how our system can cope. Therefore, we do not formulate any expectations but analyze the output of the system and the influence on the fairness metrics. Similar to the first part of this experiment, we always use the Basic Pipeline as a basis and extend this pipeline. Therefore, we do not discuss the warnings that the system is returning because of the first three steps since these were already discussed in Section 4.3. In the first step, we define the pipelines and explain how the added step(s) work. Secondly, we apply our system with the specified pipelines and record the output, which are warnings, as well as the fairness metrics. Finally, we compare the results to the Basic Pipeline.

As mentioned, each of the following pipelines extends the Basic Pipeline we defined previously. In the following, we are taking three pipelines into account.

4.5.1 Advanced Pipeline 1

We define the steps of the Advanced Pipeline 1 as follows:

1. Imputing: Imputes all missing values with a very basic strategy based on the data type of the specific column
2. Encoding: Encodes non-numerical values to numerical values
3. Scaling: Scales the numerical values to a range between -1 and 1
4. Dimensionality Reduction: Reduces the number of features by using the PCA approach

Our first advanced pipeline extends the Basic Pipeline by a Dimensionality Reduction step. The idea of this step is to reduce the amount of data. Generally, it holds that the bigger a dataset is, the more costly operations are in terms of time and storage. Therefore, it is a common goal to reduce the size of a dataset without losing any quality. One approach to achieve dimensionality reduction is by using the well-known PCA-Algorithm. In our implementation, we are compressing all features down to five features, excluding the sensitive attribute and the label attribute. Hence, we obtain a dataset containing only seven features, i.e., seven columns. How much storage it saves depends on how many features a specific dataset contains. As an example for the COMPAS Dataset, we would save 86 % in storage since it consists of 51 features. This may have a significant effect on the fairness of the datasets.

As already mentioned, we do not consider the warnings regarding the first step of the pipeline (imputation) since the Basic Pipeline already covers this. Our system is not returning a warning for any of the datasets. Hence, the system does not assess the step Dimensionality Reduction as critical. In other words, users feel this step is safe and does not need to be reconsidered. The results of the measured fairness metrics are depicted in Table 4.5.

We notice that none of the fairness metrics for the German Credit Dataset have deteriorated. Quite the contrary, they have even improved. More precisely, the metrics Equalized Odds (from 0.72 to 0.82), Equalized Opportunity (from 0.87 to 0.92), and Treatment Equality (from 0.59 to 0.61) have significantly improved, and the metric Demographic Parity (from 0.94 to 0.94) remained unchanged. This indicates clearly that no unfairness was introduced in the Dimensionality Reduction step.

This is totally different from the Census Income Dataset. In contrast to the results of the German Credit Dataset, all of the measured fairness metrics have deteriorated. More concretely, the metrics Equalized Odds (from 0.34 to 0.24), Equalized Opportunity (from 0.45 to 0.35), Demographic Parity (from 0.80 to 0.67), and Treatment Equality (from 0.34 to 0.24) have declined drastically. Remarkably, all the metrics experienced a decrease of at least 0.1. This is the most substantial decrease in fairness metrics in a step in any of the pipelines we have investigated so far. This indicates a decrease in the fairness of the Census Income Dataset.

The COMPAS Dataset showed results similar to these of the German Credit Dataset. The metrics Equalized Odds (from 0.40 to 0.58), Equalized Opportunity (from 0.82 to 0.80), and Treatment Equality (from 0.21 to 0.5) improved, and the metric Demographic Parity (from 0.99 to 0.99) stayed the same. It stands out that none of the fairness metrics has deteriorated. This indicates that the Dimensionality Reduction step did not negatively affect the fairness of this dataset.

We detect that each fairness metrics we computed have worsened for the Credit Card Clients Dataset. More precisely, the metrics Equalized Odds (from 0.56 to 0.54), Equalized Opportunity (from 0.67 to 0.61), Demographic Parity (from 0.82 to 0.74), and Treatment Equality (from 0.56 to 0.54) have deteriorated significantly. Even though the decrease in the individual metrics is not as drastic as the impact on the fairness metrics of the Census Income Dataset, it is still significant. Hence, it indicates a decrease in fairness for this step.

For the Diabetes Dataset, the results are not as clear as the measured values for the other datasets we have investigated previously. The scores for the metrics Equalized Odds, Equalized Opportunity, and Demographic Parity are still excellent, with a value of 1.00, and did not change. In contrast, the metric Treatment Equality has deteriorated significantly from 0.83 to 0.28. Because the other metrics still show excellent results, it is not clear how strongly fairness is affected by this dataset.

Overall, it becomes clear that the fairness of some datasets suffered severely from the Dimensionality Reduction step. However, the system did not output any warnings. Hence, users of our system would not have been able to identify this step as critical. Therefore, the relationship between the number of columns of a dataset and fairness should be studied in future work. Our system can easily be extended based on this outcome by considering this data characteristic.

	Equalized Odds	Equalized Opportunity	Demographic Parity	Treatment Equality
German Credit	0.82	0.92	0.94	0.61
Census Income	0.24	0.35	0.67	0.24
COMPAS	0.58	0.80	0.99	0.5
Credit Card Clients	0.54	0.61	0.74	0.54
Diabetes	1.00	1.00	1.00	0.28

Table 4.5: Fairness Metrics of Advanced Pipeline 1

4.5.2 Advanced Pipeline 2

We define the steps of the Advanced Pipeline 2 as follows:

1. Imputing: Imputes all missing values with a very basic strategy based on the data type of the specific column
2. Encoding: Encodes non-numerical values to numerical values
3. Scaling: Scales the numerical values to a range between -1 and 1
4. Upsampling with Perturbation: Upsampling and applying perturbation on these

This pipeline extends the Basic Pipeline by the step Upsampling. That pipeline is similar to Pipeline 2 but diverges notably through the introduction of perturbation applied to the generated samples. More concretely, we produce samples exactly like in Pipeline 2, adding noise to them with a standard deviation of 0.1.

The system returns the same warnings as in Pipeline 2, meaning we obtain a warning that the size of the dataset has increased and a warning that the balance has improved. Therefore, we do not repeat these warnings in this analysis. Table 4.6 depicts the measured scores for the different fairness metrics.

Even though the German Credit Dataset shows improved results compared to the Basic Pipeline, probably because of the use of perturbation, all of its fairness metrics have still deteriorated compared to the measured fairness metrics of the Basic Pipeline. More precisely, the metrics Equalized Odds (from 0.72 to 0.66), Equalized Opportunity (from 0.87 to 0.76), and Demographic Parity (from 0.94 to 0.87) have significantly worsened. Only the metric Treatment Equality has slightly improved from (0.59 to 0.62). This shows that the effects are still the same as in Pipeline 2. In other words, the fairness of this dataset decreased in this Upsampling step, meaning that our system output the warnings correctly.

The results for the Census Income Dataset are almost the same as the ones we obtained in the Basic Pipeline. The metrics Equalized Odds (from 0.34 to 0.42), Equalized Opportunity (from 0.45 to 0.58), Demographic Parity (from 0.80 to 0.84), and Treatment Equality (from 0.34 to 0.42) have increased significantly. This indicates that the fairness of this dataset has improved in this step.

For the COMPAS Dataset, we notice an increase in all fairness metrics compared to the results of the Basic Pipeline. More concretely, the metrics Equalized Odds (from 0.40 to 0.40), Equalized Opportunity (from 0.82 to 0.83), Demographic Parity (from 0.99 to 1.00), and Treatment Equality (from 0.21 to 0.22) have either increased or remained at the same score. This means no unfairness has been introduced in this Upsampling step according to the measured metrics. This is the same conclusion we got from Pipeline 2.

The Credit Card Clients Dataset also shows no signs of additional unfairness. The fairness metrics Equalized Odds (from 0.56 to 0.64), Equalized Opportunity (from 0.67 to 0.76), Demographic Parity (from 0.82 to 0.86), and Treatment Equality (from 0.56 to 0.64) have all improved significantly. In other words, not only has no unfairness been introduced in this step, but over and beyond that, the fairness has even strongly increased. This result is similar to the one we obtained from the Basic Pipeline.

For the Diabetes Dataset, we detect an outcome similar to the previous datasets. The metrics fairness Equalized Odds (from 1.00 to 1.00), Equalized Opportunity (from 1.00 to 1.00), and Demographic Parity (from 1.00 to 1.00) did not change at all. Only the metrics Treatment Equality (from 0.83 to 0.66) shows a deterioration. Overall, the excellent scores indicate that the Diabetes Dataset is fair after executing this pipeline.

In total, we received very similar results to the ones we obtained for the Basic Pipeline. Our system always identified the Upsampling and Perturbation step as critical and output two warnings for each dataset regarding the increase in size and the improvement in balance. By doing this, the system supported users by identifying potential steps that may have introduced unfairness. Therefore, the system worked properly for this pipeline.

	Equalized Odds	Equalized Opportunity	Demographic Parity	Treatment Equality
German Credit	0.66	0.76	0.87	0.62
Census Income	0.42	0.58	0.84	0.42
COMPAS	0.40	0.83	1.00	0.22
Credit Card Clients	0.64	0.76	0.86	0.64
Diabetes	1.00	1.00	1.00	0.67

Table 4.6: Fairness Metrics of Advanced Pipeline 2

4.5.3 Advanced Pipeline 3

We define the steps of the Advanced Pipeline 3 as follows:

1. Imputing: Imputes all missing values with a very basic strategy based on the data type of the specific column
2. Encoding: Encodes non-numerical values to numerical values
3. Scaling: Scales the numerical values to a range between -1 and 1
4. Replace Outliers by NaN: Identifies outliers and replaces them with NaN
5. Imputing: Imputes all missing values with a very basic strategy based on the data type of the specific column

It is a common step in data preparation pipelines to investigate the dataset for outliers. An outlier is typically defined as a sharp change between successive values [SGG10]. Usually, one strives for a dataset with no outliers since this can significantly irritate the performance of the resulting model. Therefore, it is a prevalent approach to use outlier detection methods to eliminate these fraud values and either replace them with better-suited values or cancel out the entry. One realization is presented in the Advanced Pipeline 3. This pipeline extends our primary pipeline in two steps: First, we identify outliers and replace them with NaN values. In the next step, we impute these NaN values to obtain a proper dataset. We use the IsolationForest approach to identify outliers with a contamination of 0.05. The imputation is performed as described in the section of the Basic Pipeline. Unsurprisingly, this leads to changes in the data characteristics we monitor. Step 4 leads to an increase in missing values, i.e., sparsity, if the algorithm identifies outliers since we replace identified outliers with NaN. In step 5, the number of missing values, i.e., sparsity, decreases if outliers were detected in the first place because we are imputing missing values. This may have a significant effect on the fairness of the datasets.

For each dataset, the system returned two warnings. The first warning informs users that the number of missing values has increased in step 4, and the second warning announces a decrease in the number of missing values in step 5. One can see the results of the measured fairness metrics in Table 4.7.

For the German Credit Dataset, our system returned the two mentioned warnings. First, it informs users that the number of missing values was increased by 479 on average. Then, it announces that the number of missing values has decreased by the same number. In other words, the outlier detection algorithm categorized 479 values as outliers. Considering that the German Credit Dataset only compasses 1 000 instances that are described by 13 features, this number seems high. More concretely, 3.4 % of the values were considered an outlier and were then imputed again. As we pointed out in Chapter 2.2, a dataset is likely to be significantly biased if at least 5 % to 10 % of the values are missing [Ben01; SG02]. This is not the case here, and therefore, we do not expect a deterioration in fairness. Indeed, none of the metrics Equalized Odds (from 0.72 to 0.75), Equalized Opportunity (0.87 to 88), or Demographic Parity (0.94 to 0.94) showed a decrease. Treatment Equality (from 0.59 to 0.57) is the only metric that has decreased significantly, but not significantly.

The system informed users that the sparsity has improved and worsened by 17 185 on average in steps 4 and 5 for the Adult Census Dataset. Since this dataset encompasses 48 842 described by 15 attributes, only 2.35 % of the values were affected. Therefore, steps 4 and 5 in this pipeline are not likely to affect the fairness of this dataset negatively because 2.35 % is significantly below the 5 % - 10 % range that is present in the literature [Ben01; SG02]. It turns out that the measured fairness metrics Equalized Odds (from 0.34 to 0.38), Equalized Opportunity (from 0.45 to 0.40), Demographic Parity (from 0,80 to 0.64), and Treatment Equality (from 0.34 to 0.38) do not fully support this. Especially the metrics Equalized Opportunity (from 0.45 to 0.40) and Demographic Parity (from 0,80 to 0.64) deteriorated significantly. This indicates that fairness may have suffered.

According to the system, the COMPAS Dataset experienced an increase and decrease in the number of missing values of 12 518 on average in steps 4 and 5. As mentioned earlier, the COMPAS Dataset comprises 7 214 rows and 51 columns. Hence, only 3.34 % of the values were affected, meaning that this portion was considered outliers and was replaced by imputed values. Since 3.4 % is less than 5 % - 10 %, it should not significantly impact the fairness of this dataset. The measured fairness metrics Equalized Odds (from 0.40 to 0.57), Equalized Opportunity (from 0.82 to 0.82), Demographic Parity (from 0.99 to 0.99), and Treatment Equality (from 0.21 to 0.38) show that this assumption is correct. None of these metrics has deteriorated.

For the Credit Card Clients Dataset, the system returns the information that the number of missing values has increased and then decreased by 25 494 on average in steps 4 and 5. This corresponds to a portion of 3.4 % of this dataset's total number of values. Similar to the other datasets, unfairness has not likely been introduced in the Credit Card Clients Dataset because significantly less than 5 % of its total values were affected. Indeed, none of the measured fairness metrics Equalized Odds (from 0.56 to 0.66), Equalized Opportunity (from 0.67 to 0.73), Demographic Parity (from 0.82 to 0.87), and Treatment Equality (from 0.56 to 0.66) showed any deterioration.

The system provided the information that the sparsity has improved and worsened by 49 866 on average in steps 4 and 5 for the Diabetes Dataset. This dataset contains, after the initial cleansing of 46 902 data entries that are described by 47 features. Hence, 2.3 % were affected, meaning that 2.3 % of all values were identified as outliers and imputed. Since a portion of 2.3 % is quite far away from the range of 5 % - 10 %, likely, the fairness did not deteriorate in steps 4 and 5. The fairness metrics confirm this assumption for the most part. None of the metrics, Equalized Odds, Equalized Opportunity, or Demographic Parity changed. Those stayed at the score of 1.0, which is the best possible result. Only the metric Treatment Equality (from 0.83 to 0.67) shows a decrease in the score. However, the results of the fairness metric are still excellent, meaning that no significant deterioration has occurred here.

Overall, the system provided warnings that we could use to assess the impact the outlier detection steps had on the investigated datasets. This worked perfectly for all datasets except for the Adult Census Dataset. We could see a significant deterioration in the metrics Equalized Opportunity and Demographic Opportunity even though the system's output led to the opposite interpretation. This could be because outliers may only be detected in one row, which may lead to more unfairness. A possible extension of our monitor system could be to monitor not only the number of missing values of a dataset but also the distribution of its features.

	Equalized Odds	Equalized Opportunity	Demographic Parity	Treatment Equality
German Credit	0.75	0.88	0.94	0.57
Census Income	0.38	0.40	0.64	0.38
COMPAS	0.57	0.82	0.99	0.38
Credit Card Clients	0.66	0.73	0.87	0.66
Diabetes	1.00	1.00	1.00	0.67

Table 4.7: Fairness Metrics of Advanced Pipeline 3

4.6 Part III: Threshold-Based Fairness Analysis

In this third part of the experiment, we want to focus on the choice algorithm, i.e., the algorithm that analyzes the monitored data characteristics and outputs warnings based on this analysis. In parts one and two of this experiment, we used the comprehensive approach to analyze the impact of changes in data characteristics on the fairness of datasets. This comprehensive approach always returns detailed feedback to the users. This may be difficult for users to understand, especially when they are inexperienced with this topic. Therefore, we presented the threshold-based approach for the operator identification. This approach only returns deterioration of data characteristics if certain thresholds are met. See Chapter 3.4 for more details. In the following, we want to use this threshold-based approach to determine if this can help users understand our system's output better. As thresholds, we chose the following standard thresholds:

- $sizeThreshold = 100$
- $sparsityThreshold = 0.05$
- $balanceThreshold = 0.2$

For the German Credit Dataset, we only received a warning for Pipelines 1 and 3 that the size has deteriorated by 50 % (500 instances) in the fourth step. If we compare the measured fairness metrics between the pipelines, we can see that the fairness metrics of Pipeline 1, Pipeline 2, Pipeline 3, and Advanced Pipeline 2 indicated a deterioration in fairness. Hence, the system only returned a warning for half of the pipelines that have deteriorated in fairness.

In contrast to the German Credit Dataset, the Census Income Dataset does not output any warnings. This is because none of the thresholds was met after a procession step. At first sight, this may be surprising. Indeed, it is relatively simple to explain. The fairness of the Census Income Dataset only deteriorated significantly for the Advanced Pipeline 1. This deterioration of fairness was because of a significant decrease in dimensionality. Since we are not monitoring dimensionality, i.e., the number of features, we cannot detect this change. Besides this pipeline, the fairness of the Census Income Dataset has not deteriorated significantly in any pipeline. Therefore, it is correct that the system output no other warnings.

We obtained one warning for the COMPAS Dataset. More precisely, we received a warning for Pipeline 3 that the size of the dataset has decreased by 50 % (3 607 instances) in step 4. If we take a look at the fairness metrics we have measured in the previous parts of this experiment, we detect that the fairness of this dataset only suffered significantly in Pipeline 3. Hence, the system analyzed the step that introduced the unfairness correctly.

We did not receive any warnings for the Credit Card Client Dataset. The reason for this is similar to the Census Income Dataset. The fairness metric does not indicate any decrease in fairness of any of the pipelines except for the Advanced Pipeline 1. In this pipeline, unfairness was introduced by restricting the dimensionality. Since our system cannot track this data characteristic, we are also not receiving a warning. Hence, the system worked correctly.

The system did also not return any warnings for the Diabetes Dataset. If we look at the measured fairness metrics, we see that only for Pipeline 2 do the metrics indicate that unfairness has been introduced. It is essential to mention that the fairness metric is still excellent, with 0.93 (Equalized Odds) and 0.94 (Equalized Opportunity and Demographic Parity). Therefore, it may be that the dataset was still too fair to meet any of the thresholds that trigger the operator identification.

To summarize the findings, the system with the threshold-based choice algorithm could return a warning for most pipelines that have introduced unfairness in one of its procession steps. In two cases, the algorithm could not return a warning even though the data characteristic was monitored, and the fairness metric indicated a significant increase in unfairness. Remarkably, the system output no false-positive warnings. It stands out that the system only returned warnings regarding the size. This means that the thresholds regarding balance and sparsity were never met. It is crucial to understand that two conditions must hold so that the algorithm outputs a warning. First, we need a change in at least one of the data characteristics that suggests a decrease in fairness. Second, we need the threshold regarding that specific data characteristic to be met. With this understanding, the results of the third part of this experiment become clearer. In addition, it is significantly easier for users to interpret the system's output than the comprehensive approach for the choice algorithm. This is because drastically fewer warnings or only the important warnings are shown.

4.7 Results

The first experiment clearly showed that our proposed system worked as we expected and intended. The data characteristics size, balance, and sparsity could be monitored successfully. The system returned a warning whenever one of these data characteristics had changed in a step in the data preparation pipeline. With the returned warnings, users should be able to assess whether a step had likely introduced unfairness. In practice, it turned out that interpreting these warnings is not always straightforward. Mainly, if two data characteristics are directed against each other, i.e., one characteristic indicates an increase in fairness while the other indicates a decrease in fairness, it becomes challenging to interpret the warnings correctly. Therefore, it may be helpful to set thresholds for each characteristic or heuristic that determines what characteristic should be prioritized in a particular procession step of the data preparation pipeline.

The second part of the experiment showed that the system can also cope with more complex pipelines. It also revealed that the monitored data characteristics could be extended by other characteristics so that the system could cope with a wider variety of transformation steps into a data preparation pipeline.

In the third part of the experiment, we could show that the output of the threshold-based approach for the operator identification is significantly easier to interpret than the results of the comprehensive approach. It is important to mention that the comprehensive approach gives more details. Hence, for experienced data scientists, the comprehensive approach may have advantages. For inexperienced users, the threshold-based approach is more straightforward to interpret. In addition, in rare cases, the threshold-based approach cannot return a warning even though unfairness has increased. By using suited threshold values, this error could be minimized.

It is also important to mention that in our experiment, we considered the fairness metrics Equalized Odds, Equalized Opportunity, Demographic Parity, and Treatment Equality. Therefore, the interpretation of warnings could also change depending on which metric users try to optimize. In addition, we always executed the the operator identification to determine whether the fairness metrics have improved. In practice, this should be based on a threshold that users set because we only need to seek introduced unfairness if the dataset's fairness is not good enough (cf. Section 3.3).

5 Conclusion and Outlook

5.1 Contributions

Throughout this study, our primary goal has been to develop a system that aims to facilitate the implementation of fair data preparation pipelines. Based on a detailed review of the available literature, we gave an overview of the various fairness definitions commonly used to determine whether an ML model behaves fairly. In addition, we studied the connection between typical data characteristics and the different fairness metrics.

Building upon this foundational research, we outlined the conceptual framework of our system, including the regarding algorithms. Subsequently, we checked these algorithms for potential improvements, focusing on time and space complexity, and refined them accordingly.

A central part of our work consisted of conducting experiments to evaluate the performance of our system in terms of fairness across diverse benchmark datasets and various data preparation pipelines.

5.2 Findings

As mentioned above, our work encompasses an experimental evaluation of our developed system to assess its performance across various scenarios. Based on the results of this experiment, several key findings emerge.

First, we found that the system we suggested with the comprehensive approach for the operator identification worked as expected on straightforward data preparation pipelines. Specifically, it successfully detected changes in the data characteristics size, balance, and sparsity that have occurred because of specific steps within the pipeline. Based on the detailed feedback, the users obtained valuable feedback and were, in most cases, able to predict the effects on the resulting model's fairness.

Secondly, we found that the system can generally also cope with more advanced pipelines. However, we observed steps that modified data characteristics that have not been monitored by the system. Hence, these unmonitored characteristics, despite their significant influence on fairness, were not detected by the system. Consequently, the system could not provide users with the necessary warnings for pipelines containing these steps.

Lastly, we found that the threshold-based approach for the operator identification works very well. Compared to the comprehensive approach, the feedback provided to the users in the form of warnings was notably more unambiguous. Consequently, the feedback became significantly easier to interpret, particularly benefiting inexperienced users. Thereby, this approach notably enhances the fundamental system since it leads to clear and precise feedback.

5.3 Limitations

Acknowledging several limitations of our system and study design is important. We focus on them in the following.

First, we only considered the fairness metrics Equalized Odds, Equalized Opportunity, Demographic Parity, and Treatment Equality for our system. Even though these are popular metrics, there are also other fairness metrics that we did not consider for our system. It stands out that all fairness metrics we used belong to group fairness. We did not consider any individual fairness metrics.

Secondly, our system is tailored explicitly for binary classification tasks. Therefore, it is unsuitable for datasets encompassing more than two classes. This is a significant limitation since it is not unusual for real-world datasets to have an n-ary classification.

In addition, it is important to mention that our system only monitors the three data characteristics size, balance, and sparsity. Other data characteristics potentially influence fairness and are not monitored in our proposed system.

In our experimental evaluation, we only used small data preparation pipelines with few steps so that we could compare the results easily. Data preparation pipelines are often significantly larger in real-world scenarios, encompassing more steps. We have not tried our system for such pipelines.

By recognizing these limitations, we highlight areas that may be relevant for future work and possible improvements for our system.

5.4 Outlook

We close this thesis with an outlook on research related to our developed system regarding fair data preparation pipelines. Building upon the foundations provided in this work, several promising areas exist for future exploration and refinement.

As we introduced in the literature review, other approaches to debug data preparation pipelines exist, such as MLINSPECT and fair-DAGs, that aim to ensure fairness. A comparison between these systems and the system we suggested may provide valuable insights into the strengths and limitations of each method. Such a comparison could facilitate the further development of our system.

Furthermore, future work could focus on enhancing our proposed system through extensions. More accurately, this may consist of considering more data characteristics. Moreover, our system only yet works for a binary classification. An enhancement to a n-ary system would extend its applicability in practice.

We only developed the proposed system for certain fairness metrics that belong to the category group fairness. Therefore, another research topic in the future could be how to adopt the system so that it can also consider individual fairness.

In order to monitor the system's performance closely, our experiment only used small data preparation pipelines. Data preparation pipelines are usually significantly more complex in a real-world application, encompassing significantly more steps. In addition, which step is affecting which data characteristic is often not apparent. Therefore, it would be important to investigate whether our framework works properly on real-life data preparation pipelines. Such studies may be able to point out the system's effectiveness in various contexts and provide valuable insights for further optimization.

Lastly, we introduced three different approaches that can be used for the operator identification. We also found that the threshold-based approach can significantly reduce irrelevant details and, therefore, improve the interpretability of its output. This finding makes the comparison of the three different approaches appear interesting. This can contribute to the selection of the appropriate approach in a particular use case. Additionally, we found that the ranking-based approach was difficult to realize since a ranking is hard to find. In future research, it would be worthwhile to investigate the feasibility of implementing this approach.

Bibliography

[AAA+21] A. Althnian, D. AlSaeed, H. Al-Baity, A. Samha, A. B. Dris, N. Alzakari, A. Abou Elwafa, H. Kurdi. "Impact of Dataset Size on Classification Performance: An Empirical Evaluation in the Medical Domain". In: *Applied Sciences* 11.2 (2021). ISSN: 2076-3417. DOI: 10.3390/app11020796. URL: https://www.mdpi.com/2076-3417/11/2/796 (cit. on p. 19).

[AvC18] A. Alwosheel, S. van Cranenburgh, C. G. Chorus. "Is your dataset big enough? Sample size requirements when using artificial neural networks for discrete choice analysis". In: *Journal of Choice Modelling* 28 (2018), pp. 167–182. ISSN: 1755-5345. DOI: https://doi.org/10.1016/j.jocm.2018.07.002. URL: https://www.sciencedirect.com/science/article/pii/S1755534518300058 (cit. on p. 30).

[BCFM09] C. Batini, C. Cappiello, C. Francalanci, A. Maurino. "Methodologies for data quality assessment and improvement". In: *ACM Comput. Surv.* 41.3 (July 2009). ISSN: 0360-0300. DOI: 10.1145/1541880.1541883. URL: https://doi.org/10.1145/1541880.1541883 (cit. on p. 15).

[BDE+20] S. Bird, M. Dudík, R. Edgar, B. Horn, R. Lutz, V. Milan, M. Sameki, H. Wallach, K. Walker. *Fairlearn: A toolkit for assessing and improving fairness in AI*. Tech. rep. MSR-TR-2020-32. Microsoft, May 2020. URL: https://www.microsoft.com/en-us/research/publication/fairlearn-a-toolkit-for-assessing-and-improving-fairness-in-ai/ (cit. on p. 36).

[Ben01] D. Bennett. "How can I deal with missing data in my study?" In: *Australian and New Zealand Journal of Public Health* 25 (Oct. 2001), pp. 464–469. DOI: 10.1111/j.1467-842X.2001.tb00294.x (cit. on pp. 15, 20, 30, 52, 53).

[BHJ+21] R. Berk, H. Heidari, S. Jabbari, M. Kearns, A. Roth. "Fairness in criminal justice risk assessments: The state of the art". In: *Sociological Methods & Research* 50.1 (2021), pp. 3–44 (cit. on p. 18).

[BK96] B. Becker, R. Kohavi. *Adult*. UCI Machine Learning Repository. DOI: https://doi.org/10.24432/C5XW20. 1996 (cit. on p. 37).

[BR21] S. Biswas, H. Rajan. "Fair Preprocessing: Towards Understanding Compositional Fairness of Data Transformers in Machine Learning Pipeline". In: *Proceedings of the 29th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*. ESEC/FSE 2021. Athens, Greece: Association for Computing Machinery, 2021, pp. 981–993. ISBN: 9781450385626. DOI: 10.1145/3468264.3468536. URL: https://doi.org/10.1145/3468264.3468536 (cit. on pp. 15, 19, 20).

[C3a23] C3.ai. *Data Granularity*. 2023. URL: https://c3.ai/glossary/features/data-granularity/ (cit. on p. 20).

[Cam23] Cambridge Dictionary. *Meaning of fairness in English*. 2023. URL: https://dictionary.cambridge.org/us/dictionary/english/fairness (cit. on p. 17).

[Cho16] A. Chouldechova. "Fair Prediction with Disparate Impact: A Study of Bias in Recidivism Prediction Instruments". In: *Big Data* 5 (Oct. 2016). DOI: 10.1089/big.2016.0047 (cit. on p. 18).

[CLS+15] J. Cho, K. Lee, E. Shin, G. Choy, S. Do. "How much data is needed to train a medical image deep learning system to achieve necessary high accuracy?" In: *arXiv preprint arXiv:1511.06348* (2015) (cit. on p. 30).

[CMS12] D. C. Cireşan, U. Meier, J. Schmidhuber. "Transfer learning for Latin and Chinese characters with Deep Neural Networks". In: *The 2012 International Joint Conference on Neural Networks (IJCNN)*. 2012, pp. 1–6. DOI: 10.1109/IJCNN.2012.6252544 (cit. on p. 30).

[CPF+17] S. Corbett-Davies, E. Pierson, A. Feller, S. Goel, A. Huq. "Algorithmic Decision Making and the Cost of Fairness". In: *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '17. Halifax, NS, Canada: Association for Computing Machinery, 2017, pp. 797–806. ISBN: 9781450348874. DOI: 10.1145/3097983.3098095. URL: https://doi.org/10.1145/3097983.3098095 (cit. on p. 18).

[Dev23] G. Developers. *Imbalanced Data*. 2023. URL: https://developers.google.com/machine-learning/data-prep/construct/sampling-splitting/imbalanced-data#:~:text=A%20classification%20data%20set%20with,smaller%20proportion%20are%20minority%20classes (cit. on p. 30).

[DHP+12] C. Dwork, M. Hardt, T. Pitassi, O. Reingold, R. Zemel. "Fairness through Awareness". In: *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference*. ITCS '12. Cambridge, Massachusetts: Association for Computing Machinery, 2012, pp. 214–226. ISBN: 9781450311151. DOI: 10.1145/2090236.2090255. URL: https://doi.org/10.1145/2090236.2090255 (cit. on p. 17).

[Epp23] P. Epple. *Fairness-monitoring in data preparation pipelines*. https://github.tik.uni-stuttgart.de/IPVS-DE/pipeline-fairness-monitoring. 2023 (cit. on p. 35).

[FMSS22] A. Fabris, S. Messina, G. Silvello, G. A. Susto. "Algorithmic fairness datasets: the story so far". In: *Data Mining and Knowledge Discovery* 36.6 (Sept. 2022), pp. 2074–2152. ISSN: 1573-756X. DOI: 10.1007/s10618-022-00854-z. URL: http://dx.doi.org/10.1007/s10618-022-00854-z (cit. on p. 36).

[FMST20] T. Farrand, F. Mireshghallah, S. Singh, A. Trask. "Neither Private Nor Fair: Impact of Data Imbalance on Utility and Fairness in Differential Privacy". In: *Proceedings of the 2020 Workshop on Privacy-Preserving Machine Learning in Practice*. PPMLP'20. Virtual Event, USA: Association for Computing Machinery, 2020, pp. 15–19. ISBN: 9781450380881. DOI: 10.1145/3411501.3419419. URL: https://doi.org/10.1145/3411501.3419419 (cit. on pp. 15, 19).

[FS20] M. Frye, R. H. Schmitt. "Structured data preparation pipeline for machine learning-applications in production". In: *17th IMEKO TC* 10 (2020), pp. 241–246 (cit. on pp. 3, 5, 15).

[GGR23] A. C. Garcia, M. Garcia, R. Rigobon. “Algorithmic discrimination in the credit domain: what do we know about it?” In: *AI & SOCIETY* (May 2023), pp. 1–40. DOI: 10.1007/s00146-023-01676-3 (cit. on p. 15).

[GGSS21] S. Grafberger, S. Guha, J. Stoyanovich, S. Schelter. “MLINSPECT: A Data Distribution Debugger for Machine Learning Pipelines”. In: *Proceedings of the 2021 International Conference on Management of Data*. SIGMOD ’21. Virtual Event, China: Association for Computing Machinery, 2021, pp. 2736–2739. ISBN: 9781450383431. DOI: 10.1145/3448016.3452759. URL: https://doi.org/10.1145/3448016.3452759 (cit. on p. 21).

[GGSS22] S. Grafberger, P. Groth, J. Stoyanovich, S. Schelter. “Data distribution debugging in machine learning pipelines”. In: *The VLDB Journal* 31.5 (Jan. 2022), pp. 1103–1126. ISSN: 1066-8888. DOI: 10.1007/s00778-021-00726-w. URL: https://doi.org/10.1007/s00778-021-00726-w (cit. on p. 21).

[Goo23] Google Developers. *Imbalanced Data*. 2023. URL: https://developers.google.com/machine-learning/data-prep/construct/sampling-splitting/imbalanced-data?hl=en (cit. on p. 19).

[Gri22] D. Griffin. *Definitions of Fairness in Machine Learning Explained Through Examples*. 2022. URL: https://www.studentvoice.ai/blog/definitions-of-fairness-in-machine-learning-explained-through-examples (cit. on p. 17).

[Hof94] H. Hofmann. *Statlog (German Credit Data)*. UCI Machine Learning Repository. DOI: https://doi.org/10.24432/C5NC77. 1994 (cit. on p. 36).

[HPS16] M. Hardt, E. Price, N. Srebro. “Equality of Opportunity in Supervised Learning”. In: *Proceedings of the 30th International Conference on Neural Information Processing Systems*. NIPS’16. Barcelona, Spain: Curran Associates Inc., 2016, pp. 3323–3331. ISBN: 9781510838819 (cit. on pp. 17, 18).

[JA16] L. K. Jeff Larson Surya Mattu, J. Angwin. *How We Analyzed the COMPAS Recidivism Algorithm*. 2016. URL: https://www.propublica.org/article/how-we-analyzed-the-compas-recidivism-algorithm (cit. on p. 37).

[JPN+20] A. Jain, H. Patel, L. Nagalapatti, N. Gupta, S. Mehta, S. Guttula, S. Mujumdar, S. Afzal, R. Sharma Mittal, V. Munigala. “Overview and Importance of Data Quality for Machine Learning Tasks”. In: *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. KDD ’20. Virtual Event, CA, USA: Association for Computing Machinery, 2020, pp. 3561–3562. ISBN: 9781450379984. DOI: 10.1145/3394486.3406477. URL: https://doi.org/10.1145/3394486.3406477 (cit. on pp. 3, 5, 15).

[JWC21] H. Jeong, H. Wang, F. P. Calmon. “Fairness without Imputation: A Decision Tree Approach for Fair Prediction with Missing Values”. In: *CoRR* abs/2109.10431 (2021). arXiv: 2109.10431. URL: https://arxiv.org/abs/2109.10431 (cit. on p. 26).

[Kle18] J. Kleinberg. “Inherent Trade-Offs in Algorithmic Fairness”. In: *SIGMETRICS Perform. Eval. Rev.* 46.1 (June 2018), p. 40. ISSN: 0163-5999. DOI: 10.1145/3292040.3219634. URL: https://doi.org/10.1145/3292040.3219634 (cit. on p. 18).

[KLRS17] M. Kusner, J. Loftus, C. Russell, R. Silva. "Counterfactual Fairness". In: *Proceedings of the 31st International Conference on Neural Information Processing Systems*. NIPS'17. Long Beach, California, USA: Curran Associates Inc., 2017, pp. 4069–4079. ISBN: 9781510860964 (cit. on pp. 17, 18).

[KM03] T. Kavzoglu, P. Mather. "The use of backpropagating artificial neural networks in land cover classification". In: *International Journal of Remote Sensing - INT J REMOTE SENS* 24 (Dec. 2003), pp. 4907–4938. DOI: 10.1080/0143116031000114851 (cit. on p. 30).

[LGA22] N. Li, N. Goel, E. Ash. "Data-Centric Factors in Algorithmic Fairness". In: *Proceedings of the 2022 AAAI/ACM Conference on AI, Ethics, and Society*. AIES '22. Oxford, United Kingdom: Association for Computing Machinery, 2022, pp. 396–410. ISBN: 9781450392471. DOI: 10.1145/3514094.3534147. URL: https://doi.org/10.1145/3514094.3534147 (cit. on pp. 15, 19).

[LRI+22] T. Le Quy, A. Roy, V. Iosifidis, W. Zhang, E. Ntoutsi. "A survey on datasets for fairness-aware machine learning". In: *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 12.3 (2022), e1452 (cit. on pp. 36–38).

[MMS+21] N. Mehrabi, F. Morstatter, N. Saxena, K. Lerman, A. Galstyan. "A Survey on Bias and Fairness in Machine Learning". In: *ACM Comput. Surv.* 54.6 (July 2021). ISSN: 0360-0300. DOI: 10.1145/3457607. URL: https://doi.org/10.1145/3457607 (cit. on pp. 17–19).

[My 22] My Great Learning. *Understanding Curse of Dimensionality*. 2022. URL: https://www.mygreatlearning.com/blog/understanding-curse-of-dimensionality/ (cit. on p. 20).

[Par22] Parth Shukla. *Dealing With Sparse Datasets In Machine Learning*. 2022. URL: https://www.analyticsvidhya.com/blog/2022/10/dealing-with-sparse-datasets-in-machine-learning/ (cit. on p. 20).

[PCM23] K. Peng, J. Chakraborty, T. Menzies. "FairMask: Better Fairness via Model-Based Rebalancing of Protected Attributes". In: *IEEE Transactions on Software Engineering* 49.4 (2023), pp. 2426–2439. DOI: 10.1109/TSE.2022.3220713 (cit. on p. 17).

[PLW03] L. Pipino, Y. Lee, R. Wang. "Data Quality Assessment". In: *Communications of the ACM* 45 (July 2003). DOI: 10.1145/505248.506010 (cit. on p. 15).

[Pro24] ProPublica. *COMPAS Recidivism Risk Score Data and Analysis*. 2024. URL: https://www.propublica.org/datastore/dataset/compas-recidivism-risk-score-data-and-analysis (cit. on p. 37).

[SDG+14] B. Strack, J. Deshazo, C. Gennings, J. L. Olmo Ortiz, S. Ventura, K. Cios, J. Clore. "Impact of HbA1c Measurement on Hospital Readmission Rates: Analysis of 70,000 Clinical Database Patient Records". In: *BioMed research international* 2014 (Apr. 2014), p. 781670. DOI: 10.1155/2014/781670 (cit. on p. 38).

[SG02] J. Schafer, J. Graham. "Missing Data: Our View of the State of the Art". In: *Psychological Methods* 7 (June 2002), pp. 147–177. DOI: 10.1037/1082-989X.7.2.147 (cit. on pp. 15, 20, 30, 52, 53).

[SGG10] A. Sharma, L. Golubchik, R. Govindan. "Sensor Faults Detection Methods and Prevalence in Real-World Datasets". In: *TOSN* 6 (Jan. 2010) (cit. on p. 52).

[VLA19] I. Valentim, N. Lourenço, N. Antunes. "The Impact of Data Preparation on the Fairness of Software Systems". In: *2019 IEEE 30th International Symposium on Software Reliability Engineering (ISSRE)*. 2019, pp. 391–401. DOI: 10.1109/ISSRE.2019.00046 (cit. on pp. 3, 5, 15).

[VR18] S. Verma, J. Rubin. "Fairness Definitions Explained". In: *Proceedings of the International Workshop on Software Fairness*. FairWare '18. Gothenburg, Sweden: Association for Computing Machinery, 2018, pp. 1–7. ISBN: 9781450357463. DOI: 10.1145/3194770.3194776. URL: https://doi.org/10.1145/3194770.3194776 (cit. on pp. 17, 18).

[Wan21] Y. Wang. "Analyzing the impact of missing values and selection bias on fairness". In: *International Journal of Data Science and Analytics* 12 (Aug. 2021), pp. 1–19. DOI: 10.1007/s41060-021-00259-z (cit. on p. 20).

[Why24] K. Whyte. *Data Quality: Getting the Most Value Out of Your Data*. 2024. URL: https://qualityworkscg.com/data-quality-getting-the-most-value-out-of-your-data/ (cit. on p. 15).

[Yeh16] I.-C. Yeh. *Default of credit card clients*. UCI Machine Learning Repository. DOI: https://doi.org/10.24432/C55S3H. 2016 (cit. on p. 38).

[YHSS20] K. Yang, B. Huang, J. Stoyanovich, S. Schelter. "Fairness-Aware Instrumentation of Preprocessing Pipelines for Machine Learning". In: *Workshop on Human-In-the-Loop Data Analytics (HILDA'20)* (2020). DOI: 10.1145/3398730.3399194. URL: https://par.nsf.gov/biblio/10182459 (cit. on p. 21).

[YL09] I.-C. Yeh, C.-h. Lien. "The comparisons of data mining techniques for the predictive accuracy of probability of default of credit card clients". In: *Expert Syst. Appl.* 36 (2009), pp. 2473–2480 (cit. on p. 38).

All links were last followed on March 27, 2024.

Declaration

I hereby declare that the work presented in this thesis is entirely my own and that I did not use any other sources and references than the listed ones. I have marked all direct or indirect statements from other sources contained therein as quotations. Neither this work nor significant parts of it were part of another examination procedure. I have not published this work in whole or in part before. The electronic copy is consistent with all submitted copies.

place, date, signature