Bias quantification measures based on fuzzy rough sets

Lisa Koutsoviti Koumeri
STUDENT NUMBER: 2048041

THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE IN COGNITIVE SCIENCE & ARTIFICIAL INTELLIGENCE OR DATA SCIENCE & SOCIETY
DEPARTMENT OF COGNITIVE SCIENCE & ARTIFICIAL INTELLIGENCE
SCHOOL OF HUMANITIES AND DIGITAL SCIENCES
TILBURG UNIVERSITY

Thesis committee:

Dr. Gonzalo Nápoles
Dr. Raquel Garrido Alhama

Tilburg University
School of Humanities and Digital Sciences
Department of Cognitive Science & Artificial Intelligence
Tilburg, The Netherlands
December 2020

Preface

I wish to sincerely thank the supervisor of this thesis, Dr. Gonzalo Nápoles, for inspiring me to work on this topic and for his guidance and support throughout the semester. I also extend my thanks to my peers and friends at Tilburg University without whom I would not have been able to make it through my master's degree. Finally, I would like to thank my mother for her unconditional support and faith.

Bias quantification measures based on fuzzy rough sets

Lisa Koutsoviti Koumeri

The need to measure and mitigate bias in data sets that are used in machine learning tasks has gained wide recognition in the field of Artificial Intelligence (AI) during the past decade. The academic and business community calls for the introduction of new measures of bias as existing ones are highly context specific. This thesis aims at proposing model-independent measures to quantify bias in pattern classification data sets using the fuzzy-rough set theory. The proposed measures attempt to quantify bias in two data sets which have been extensively explored in the context of AI Fairness. The outputs of the proposed measures are compared with the bias quantification measures developed by AI Fairness 360. The advantages of fuzzy-rough sets over existing bias quantification measures are threefold: (1) they do not follow mathematical definitions of fairness which can be context-dependent, (2) they explore how bias is materialized in terms of decision classes and (3) they quantify uncertainty in decision-making. According to the results of the experiment conducted in the framework of this thesis, one of the proposed measures aligns with the baseline measures. The extent and the cause of this alignment need to be further investigated in future research.

1 Introduction

This work aims at introducing measures based on the fuzzy-rough set theory to quantify bias in tabular data. First, a few terms are defined. Then, the motivation behind the need to develop new bias metrics is explained followed by the reasons that fuzzy-rough sets might be able to provide new insight when it comes to bias quantification. Finally, the research questions are stated.

1.1 Definitions

The types of data sets to which the proposed metrics are applicable are tabular data which are used in the context of pattern recognition[1]. This work aims at introducing model-independent measures. The term *model-independent* is related to the amount of data pre-processing needed before the measures are applied. Goal is to create measures that can be applied in any research context and, therefore, only the absolutely minimum amount of data pre-processing should be necessary. Bias is a naturally vague and incomplete concept as there are different sources and types of bias (Bellamy et al., 2018). In this thesis, two types of bias are used, namely the *sample* and *label* bias as defined by Hinnefeld et al. (2018). *Sample* bias is generated when protected and unprotected groups

1 Bishop (2006) defines pattern recognition as a field that deals with the automatic discovery of regularities in data through the use of computer algorithms. These regularities are used to take actions such as classifying the data into different categories.

are sampled in a different way. *Label* bias occurs when the decision making process that produces the class labels takes into account the protected attribute. Protected attributes are personal characteristics such as gender, race or belief. Such features should not put a person at a substantial disadvantage in relation to a relevant matter compared to people with different personal traits (Equality Act 2010, 2016). The intuition is that in a bias-free world personal attributes such as race or gender should not be the decisive factors when it comes to selecting loan applicants or predicting patients' healthcare costs.

1.2 Inherent difficulties in measuring bias in data

AI systems are widely used for decision-making purposes in both the business and the government sector. Some examples include the criminal justice domain where AI systems are used in predictive crime mapping (Završnik, 2020), the banking domain to facilitate loan granting procedures (Al-Blooshi & Nobanee, 2020) or the health-care sector to allocate medical resources to patients (Lysaght et al., 2019). Research revealed that due to imbalanced or unrepresentative data (Barocas et al., 2019), the aforementioned systems often correlate their predictions with protected attributes, a practice that often leads to unfair and discriminatory decisions or even violations of human rights.

This issue gave rise to the notion of Fairness in Artificial Intelligence (AI Fairness) (Foulds & Pan, 2018) that aims at detecting and treating bias in data. In the context of AI Fairness, several mathematical definitions of fairness have been created (Vluymans et al., 2015; Barocas et al., 2019; Hinnefeld et al., 2018) and, along with them, their respective bias quantification metrics. Existing fairness definitions measure different notions of fairness (Verma & Rubin, 2018a) and, therefore, the outputs of the related bias metrics might differ or contradict each other.

Researchers report that no single fairness metric is universally applicable or non-context-specific (Hinnefeld et al., 2018; Barocas et al., 2019). Several parameters need to be taken into account to decide if an existing measure is applicable to a certain case. Such parameters include the origins of the bias in the data, imbalances in the ground truth, and the metric sensitivity on bias type (Hinnefeld et al., 2018). In addition to these, human judgement might also needed. According to the developers of AI Fairness 360, an open-source toolkit developed by IBM Research in an effort to detect and mitigate bias in data and algorithms, all metrics offered in the toolkit are context dependent and should be used in a very limited setting (Bellamy et al., 2018). The authors themselves make an open call for the expansion of the existing bias quantification metrics.

1.3 Why would fuzzy-rough sets be suitable to quantify bias?

To the best of this authors' knowledge, fuzzy-rough sets have yet to be applied in the context of bias quantification. Fuzzy sets (Zadeh, 1965) and rough sets (Pawlak, 1982) are known for their ability to represent imperfect information (Cornelis et al., 2008). They deal with classification of vague and uncertain data by assigning a membership degree between one and zero to an element of a decision class, a process called fuzzification. Rough sets account for incomplete information such as when data does not suffice to discern an instant between two classes (Vluymans et al., 2015) by making use of rough approximations (Bhatt & Gopal, 2007). In other words, fuzzy-rough sets can discern elements to a certain extent by integrating the rule generation technique and ambiguity handling power of rough sets. In the context of bias quantification, they are able to quantify the extent to which an observation belongs to a certain decision

class (Nápoles et al., 2017). If we extend that idea, we can attempt to measure how the membership degrees chance as we exclude protected attributes from a data set.

Another advantage of fuzzy-rough sets it that they can model the strength of individual attributes (Vluymans et al., 2015) and thus select the most significant features without transformation of data. They are currently successfully applied in machine learning on tasks related to attribute selection and data set pre-processing (Cornelis et al., 2008). (Arunkumar & Ramakrishnan, 2018) as they can express how features discern between classes preserving, at the same time, the decision making power of the original set of features (Verbiest et al., 2013). This notion can be used to quantify the extent to which a protected feature influences decision making.

Fuzzy-rough sets are also able to provide an explainable semantic background by quantifying the exact membership degree of an instance to a wider class. In other words, adding or subtracting variables to the total variable set, can be quantified through the change in the membership degree of each particular instance. For that reason, fuzzy-rough set-based algorithms are often used in decision making (Liu et al., 2019). This property is especially useful when it comes to the explainability of the proposed bias quantification measures.

1.4 Data sets and baseline metrics

The bias quantification metrics that are developed by Bellamy et al. (2018) in the AIF360 toolkit are used to evaluate the proposed measures. In particular, the baseline measures are (1) mean difference, (2) disparate impact, (3) equal opportunity difference and (4) average odds difference. These baseline measures were chosen in terms of their popularity as fairness metrics. Furthermore, AIF360's toolkit offers a number of scripts written in Python and R programming languages that the demonstrate the use of the above-mentioned metrics. The goal is to examine whether the proposed fuzzy-rough set-based measures align with the ones introduced by AIF360 and how they differ in terms of the type of knowledge they offer. The two data sets upon which the proposed measures are tested are the *German Credit* data set, retrieved from the UCI Machine Learning Repository (Dua & Graff, 2017a), and the *COMPAS (Correctional Offender Management Profiling for Alternative Sanctions)* data set (Larson et al., 2017) retrieved from ProPublica's repository in GitHub. (Foulds & Pan, 2018; Speicher et al., 2018; Vluymans et al., 2015). *German Credit* aims at classifying bank customers in terms of creditworthiness based on financial and personal information. *COMPAS* is used to predict how likely are criminal defendants to re-offend. Both data sets have been extensively studied by developers of fairness metrics and AIF360 as examples of data that yield biased results without proper balancing actions or bias treatment.

1.5 Research questions

Five bias quantification measures are proposed and each measure is connected to a research question. In order to discuss the intuition behind the proposed measures, it is important to first explain their input: membership values. Nápoles et al. (2017)'s method is adopted to compute the membership values. Membership values are computed for each instance in a data set indicating the extent to which it belongs to the positive, negative and boundary fuzzy-rough region of a decision class. Their values range between one and zero. Observations with higher membership values in the positive regions of a decision class belong to this decision class to a higher extent. The opposite happens with negative regions. If an instance is assigned a higher membership value in

the boundary region of a decision class, then it is unclear whether or not this instance belongs to the decision class in question. Membership values are computed using the original versions of a data set. Then, each of the protected attributes is excluded from the data set and membership values are computed once again. The change between the membership values of the original and the modified data sets where a protected feature is suppressed is explored by the proposed measures. This change is considered to be a bias proxy.

The proposed measures are divided in two groups. The first group concerns the development of local change measures that quantify how much the absence of a protected feature modifies the fuzzy-rough positive and boundary regions as we shift from a decision class to another. If the outputs of the measures are relatively large, then there increased movement of instances from one decision class to another caused by the removal of a protected feature. This means that the protected feature has a larger influence in the decision-making process which serves as an indication of bias. More specifically:

1. First set of research questions: Local measures
    1.1. Would suppressing a certain data set feature cause instances to move from a fuzzy-rough positive region to another?
    1.2. Would suppressing a certain data set feature cause instances to move from a fuzzy-rough positive region to a boundary region?

The second group of measures concerns the development of global change measures that quantify how much the absence of a protected feature modifies the fuzzy-rough regions of the same decision classes. They measure the absolute change in the membership values of the same decision classes. How many instances will be classified differently in the absence of a protected feature? The bigger the change, the larger the bias. More specifically:

2. Second set of research questions: Global measures
    2.1. How much does the absence of a feature change the fuzzy-rough positive regions?
    2.2. How much does the absence of a feature change the fuzzy-rough boundary regions?
    2.3. How much does the absence of a feature change the fuzzy-rough negative regions?

Concluding, the outputs of the proposed measures are compared to the outputs of the baseline metrics in terms of trend and relative magnitude. The outputs per data set are not compared to one another due to their different dimensionalities.

2 Related Work

In this section existing bias quantification attempts are discussed along issues that arose. Then, the baseline measures are presented and examined in respect to their limitations. Finally, related work regarding the bias quantification method that is used in this thesis is presented.

2.1 Bias quantification attempts

Bias is related to the wider concept of fairness in Artificial Intelligence (AI). Attempts to quantify bias are based on the different mathematical definitions of fairness which amount to at least 20 (Verma & Rubin, 2018a) and are highly context-dependent. Mehrabi et al. (2019) offer a comprehensive guide that sums up the bundle of different definitions of AI fairness, present existing bias quantification metrics and bias mitigation algorithms, and recommend data sets for fairness research among them being *COMPAS* and *German credit*.

The efficiency and limitations of existing bias quantification metrics has been extensively studied. In their experiment, Hinnefeld et al. (2018) illustrate the limitations of well-known bias quantification metrics. They create several variations of the same data set by artificially introducing different types of bias into each one of them. After applying the metrics into both the original and the different variations of biased data sets, they report that the metrics perform differently in each data set. They mention that if the ground truth is balanced, then metrics perform well. In the opposite case, the metrics measure dramatic unfairness. They conclude that to use these metrics, researchers need to consider the process that generated the data, the ground truth and the consequences.

Barocas et al. (2019) discuss the relationship between fairness and machine learning. They mention that treating label bias is critical since it is going to bias the predictions. According to their analysis, there are many possible mathematical metrics that measure inequalities without having to consider moral notions of fairness which is also the case for the proposed measures of this thesis.

2.2 Baseline measures

Several open source libraries offering a number of fairness metrics exist and are listed in the works of Mahoney et al. (2020) and Bellamy et al. (2018). AIF360 (Bellamy et al., 2018) unified these efforts and offers a comprehensive open source Python package containing 77 bias detection metrics. Mahoney et al. (2020) and Bellamy et al. (2018) introduce the toolkit AIF360, including the developed bias quantification metrics. They support the view that the metrics included in the toolkit should be carefully chosen based on the subject matter and warn that the metrics do not yet capture the full scope of fairness in all situations.

Among the 77 metrics offered by Bellamy et al. (2018), four are chosen to serve as a baseline for the evaluation of the proposed bias quantification metrics, namely *Disparate impact*, *Statistical parity difference*, *Equal opportunity difference* and *average odds difference*. They are chosen by AIF360 to demonstrate the concepts and capabilities of the open source toolkit. Furthermore, the measures are extensively discussed in the literature (Shrestha & Yang, 2019; Corbett-Davies et al., 2017). The main characteristics and limitations of these baseline metrics are discussed in the following two paragraphs.

Disparate impact captures the unintended discrimination that occurs when a selection process has different outcomes for different groups (Feldman et al., 2015). Feldman et al. (2015) defines the ratio-based mathematical formula of disparate impact which assumes that the ground truth positive rates for both groups are equal. For example, this measure does not work in medical cases where the ground truth dictates that there are proven differences between, for example, breast cancer rates in males and females(Hinnefeld et al., 2018). This is the only fairness metric that is referenced in

US case law and, therefore, finds itself among the most popular metrics (Hinnefeld et al., 2018). An issue with this metric can arise when the sampling process for each group is different which means that between group comparisons are not equal. For example, if a researcher samples every single individual from one group, but only high-risk individuals from another, then the the disparate impact ratios for both groups cannot be compared. *Statistical parity difference* is a measure of group fairness that quantifies whether the demographics of those who belong in one class are identical to the demographics of the population. Chouldechova (2017) states that it works well in settings where equal proportions across demographic groups are desirable such as university admissions or hiring processes but it is not suitable in the case of recidivism prediction that is related to pubic safety. Dwork et al. (2011) demonstrates its limitations in examples where statistical parity is maintained, but from an individual's perspective the outcome remains unfair. Furthermore, by definition, it ignores any possible correlation between positive outcome and protected attributes (Shrestha & Yang, 2019). These two measures are consistent with each other.

Two further notions that measure discrimination based on protected attributes are *equalized odds* and *equal opportunity* as introduced by Hardt et al. (2016). According to the writers, they offer a remedy to the main limitations of demographic parity. These two criteria of non-discrimination make use of predictions generated by a supervised machine learning model. *Equalized odds* compares the true positive rates between the privileged (for example, Caucasian males in Compas data set) and unprivileged groups (for example, African-American males) in the protected attribute. In other words, it prohibits abusing the protected attribute as a proxy for predicting the target class (Hardt et al., 2016). On, the other hand *equal opportunity* has one additional constraint that is considering the predictions only within the 'advantaged' label of the target class (for example, eligibility to receive a loan or unlikely to re-offend). It is a relaxation of equalized odds.

There are a few limitations in the above-mentioned notions. *equalized odds* and *equal opportunity* both require reliably labeled data which is hard to achieve (Hardt et al., 2016; Holstein et al., 2019). The issue here lies again with the credibility of the ground truth: if the labels are biased, then results will also be biased (Hardt et al., 2016). Furthermore, measuring the target variable often requires domain-specific knowledge which places an extra constraint to the research process (Holstein et al., 2019).

2.3 Employing fuzzy-rough set theory in the context of bias quantification

Vluymans et al. (2015) provide an overview of fuzzy-rough set applications in machine learning. Details concerning the mathematical foundations of the fuzzy-rough set theory can be found in the work of Cornelis et al. (2008).

This thesis follows the method applied by Nápoles et al. (2017) in sections 3.2 and 4.1 of their work. The authors offer a way to extract information from data sets in a form that can be used by the proposed bias quantification measures. More specifically, Nápoles et al. (2017) offer a way to quantify the extent to which each observation in a data set belongs to a decision class. The procedure to calculate this quantity is called granulation process. This process assigns membership values to each observation in a data set expressing the extent to which an observation is a member of a decision class. The mathematical formulas adopted are described in the Section 3.

3 Methods

This chapter describes the methods adopted to quantify bias in the two data sets. It is divided into three sub-sections. The first sub-section describes which concepts of the fuzzy-rough set theory are relevant to quantifying bias. The second sub-section introduces the mathematical foundation of the fuzzy-rough set theory that is relevant to transform the data set values into flexible information granules that describe (or associated with) each decision class. This granulation stage is vital as the resulting values will be used as input to the proposed bias quantification measures. The second sub-section introduces the proposed bias quantification measures including their mathematical foundation and the intuition behind their creation which is the contribution of this thesis. The computational algorithms are also presented.

3.1 Relevant fuzzy-rough theory

Fuzzy-rough sets combine two important notions in computing (Cornelis et al., 2008): soft computing and granular computing. Soft computing is related to dealing with imperfect data and knowledge thus adhering closer to the human mind than traditional hard computing notions (Cornelis et al., 2008). Granular computing semantically transforms the data to information granules which can be compared and grouped with each other in terms of similarity or closeness (Yao, 2007). Similarity is an important concept in this thesis because a similarity function will be used to create the fuzzy-rough sets by comparing instances, first, with each other and, second, with the decision classes. In the Future Challenges section of Vluymans et al. (2015)'s work, the authors mention that most approaches apply the same process when comparing data, hence a future research direction is the automatic construction of appropriate similarity metrics based on the application. This thesis also explores bias quantification from that aspect.

Fuzzy-rough sets have been used to improve an existing treatment method of imbalanced data that artificially creates new instances in the underrepresented class (Ramentol et al., 2012). This work was motivated by the fact that standard classification methods are often biased towards the majority class as they do not take data distribution into account. The authors compute a positive fuzzy-rough set region using a membership function to measure the quality of an instance as a typical representative of its class. They report that fuzzy-rough sets were successful in improving the performance of the imbalanced data treatment method.

Computing the similarity between instances as well as between instances and decision classes is a vital part of this thesis and has been explored in the context of fuzzy-rough set nearest neighbour (FRNN) classification (Jensen & Cornelis, 2011). Jensen & Cornelis (2011)'s work further combines FRNN with the concepts of lower and upper approximations which is also applied in this thesis. FRNNs combined with positive regions (or lower approximations) also address the issue of mislabeled neighbours as they are able to calculate the overall similarity between of an instance and a decision class irrespective of the label. In this way, the positive region is used to measure the quality of a neighbour (Verbiest et al., 2012). This notion is extended to address label bias in this thesis.

Measuring the distance or the change between the regions of fuzzy-rough sets has been studied by McCulloch et al. (2013); Yang et al. (2018); Bello et al. (2020, 2019), but never in the context of bias quantification. In all situations, fuzzy-rough sets were successful in improving the efficiency of the applied machine learning methods.

3.2 Granulation stage: creating the membership values

The rationale behind computing the fuzzy-rough regions is described in Cornelis et al. (2008)'s work. The authors mention that the process of information granulation through the lens the fuzzy-rough set theory comprises of two parts. First, a subset A of a given universe X may be generalized to a fuzzy set (inside the same universe) allowing that objects can belong to a concept to varying degrees. Second, these objects can be similar to a certain extent which is expressed by a fuzzy relation R. Based on their similarity to one another, objects are then categorized into classes, or granules, with soft boundaries. The granules in the context of this thesis are three fuzzy-rough regions per class: a positive, a negative and a boundary region. Elements of X can belong simultaneously to the three soft granules, or the fuzzy-rough regions, to a varying degrees. Consequently, these similarity degrees will be used as input into the the bias quantification measures.

3.2.1 Defining membership and similarity functions

Membership and similarity formulas are needed to create the fuzzy-rough regions. The approach proposed by Nápoles et al. (2017) in sections 3.2 and 4.1 of their work is adopted to that end. Two membership formulas are defined first. They allow for uncertainty management as they assume that different patterns within observations have different probabilities of being correctly labeled. They quantify the extent to which an object could belong to each decision class. In this way, similar instances can be 'grouped together'.

The two different membership functions are calculated as follows. It is assumed that there is a universe of discourse U. There is a fuzzy set $X \in U$ where $\mu_X(x)$ is its membership function that determines the degree to which $X \in U$ is a member of X. It is also assumed that there is a fuzzy binary relation $R \in F(U \times U)$ where $\mu_R(y, x)$ is its membership function that determines the degree to which y is presumed to be a member of X from the fact that x is a member of the fuzzy set X.

Equation (1) measures the degree to which an instance belongs or does not belong to a certain decision class which is important as not all instances belong to the class with the same certainty,

$$\mu_{X_k}(x) = \begin{cases} \frac{1}{2}\left(1 + \frac{\phi(x,C_k)}{\sum_{i=1}^{K}\phi(x,C_i)}\right), x \in X_k \\ \frac{1}{2}\left(\frac{\phi(x,C_k)}{\sum_{i=1}^{K}\phi(x,C_i)}\right), \qquad x \notin X_k \end{cases} \tag{1}$$

such that $0 \leq \phi(x, C_k \leq 1)$ is a similarity function and $C_k = \sum_{x_{i \in X_k}} \frac{x_i}{|X_k|}$ denotes the prototype of each decision class. In simpler words, C_k is a vector containing the means of each numerical attribute. When it comes to nominal variables, the prototype is the mode of all different combinations of nominal attributes within the instances.

The similarity function is defined below. The similarity function follows the Heterogeneous Manhattan-Overlap Metric (HMOM) (Wilson & Martinez, 1997). The function was developed as there is need to compute the distance between mixed-type data. Equation (2) computes the similarity ϕ between two objects $x \in X$ and $y \in X$,

$$\phi(x, y) = 1 - \delta_{HMOM}(x, y) \tag{2}$$

where ϕ denotes the similarity function between two instances that contain both nominal and numerical values. It is computed using the HMOM which was proved by Nápoles et al. (2017) to be the superior distance function among mixed data similarity functions that allowed the proposed classifier in their work to reach better prediction rates. Equation 3 and Equation 4 compute HMOM,

$$\delta_{HMOM}(x,y) = \sum_{j=1}^{|\Phi|} \rho_j(x,y) \tag{3}$$

where

$$\rho_j(x,y) = \begin{cases} 0 & if \;\; \phi_j \;\; is \;\; nominal \wedge x(j) = y(j) \\ 1 & if \;\; \phi_j \;\; is \;\; nominal \wedge x(j) \neq y(j) \\ ||x-y||_{L_1} & if \;\; \phi_j \;\; is \;\; numerical \end{cases} \tag{4}$$

where L_1 is the sum of absolute vector values. HMOM is normalized by dividing by the number of features. Finally, the fuzzy binary relation $\mu_R(y,x)$ is computed as shown in Equation (5)(Nápoles et al., 2017),

$$\mu_R(y,x) = \mu_{X_k}(x)\phi(x,y) \tag{5}$$

where X_k refers to all instances that belong to a certain decision class. As mentioned in the beginning of the subsection, this membership function also accounts for x being a member of a certain decision class. To conclude, the membership functions will assign scores between zero and one to each observation. Each score corresponds to each decision class. These scores will be used to define the lower and upper approximations of the fuzzy-rough sets.

3.2.2 Defining the Lower and upper approximations

In the rough set theory, the lower approximation of a set contains the instances that definitely belong to the decision class in question, whereas the upper approximation of a set contains instances that may or may not belong to the decision class. For a closer look into the intuition behind the formulas, refer to Nápoles et al. (2017). To define the lower approximations, the degree of x being a member of X_k under the knowledge R is used. It is computed in Equation (6),

$$\mu_{R_*(X_k)}(x) = min\{\mu_{X_k}(x), inf_{y \in U} I(\mu_R(y,x), \mu_{X_k}(y))\} \tag{6}$$

where $I(a,b) = \min(1-a+b,1)$ is the Łukasiewicz implication operator. The membership function for the upper approximation is defined in Equation (7),

$$\mu_{R_*(X_k)}(x) = max\{\mu_{X_k}(x), sup_{y \in U} T(\mu_R(x,y), \mu_{X_k}(y))\} \tag{7}$$

where $T(a,b) = \max(a+b-1,0)$ is the Łukasiewicz conjunction operator.

3.2.3 Defining the fuzzy-rough positive, negative and boundary regions

Based on the above elements, three fuzzy-rough regions are defined. This process comprises the granulation stage's core. Equations (8), (9) and (10) display the membership functions associated with the fuzzy-rough positive, negative and boundary regions, respectively using the notions of upper and lower approximation (Nápoles et al., 2017),

$$\mu_{POS(X_k)}(x) = \mu_{R_*(X_k)}(x), \tag{8}$$

$$\mu_{NEG(X_k)}(x) = 1 - \mu_{R^*(X_k)}(x), \tag{9}$$

$$\mu_{BND(X_k)}(x) = \mu_{R^*(X_k)}(x) - \mu_{R_*(X_k)}(x). \tag{10}$$

These functions assign a value between zero and one to each observation. This number showcases how an object can belong to more than one class with varying degrees. These memberships functions allow computing more flexible information granules describing the problem.

3.2.4 Computing the membership values in programming environment

This section explains the process of computing the membership values to the three fuzzy-rough regions for each observation in a programming environment. A class object is created for this purpose that is broken down into some parts: Algorithm 1a to 1f. This class object receives a data set as input. The variables in the data set need to be reordered in the following way: numeric columns are first, categorical columns follow next and the final column is a binary decision class. The decision class values are one and zero Each part is connected with the mathematical theory as explained in the Methods section.

First step in the process of creating the membership values is initializing the variables that will be used later by the functions in the class object as shown in Algorithm (1a). The following variables are initialized: X which stores the values of the

Algorithm 1a Class object: computes membership values per region and decision class

Input: Data set
Output: Membership values in the positive, negative and boundary regions per decision class.

```
1: function INITIALIZE VARIABLES(Input: Data set)
2:     Output: X, C
```

data set in a 2-D array format and C which is an empty array that stores the mean of numeric columns per class.

Algorithm (1b) contains the function that calculates the membership values. First, it divides the data set into two: a data set containing categorical columns (*Xcat*) and another one containing numeric columns (*Xnum*). These two data sets are used to calculate the similarity between categorical and numerical observation values separately only to aggregate them later. Then, a variable to calculate the numeric similarity is created: the mean values of variables per decision class are computed using the numeric data set *Xnum* (refer to parameters of Equation 1) and stored in variable *C*.

Next, the pair-wise similarity between individual observations is calculated. First, the algorithm computes the variables *SC* and *SN* which were defined in Part (1a). *SC* stands for similarity between categorical observations and is a square-form matrix of pair-wise distances between categorical values in observations. It is computed using

Algorithm 1b Class object: computes membership values per region and decision class

3: **function** COMPUTE MEMBERSHIP VALUES(**Input**: data set)
4: Identify numeric and categorical variables in data set
5: Divide data set into $Xnum$ (numeric columns) and $Xcat$ (categorical columns)

 # Process to calculate numeric similarity between instances
6: **for** each decision class in data set **do**
7: Calculate vector of means of each variable in $Xnum$ using $\sum_{x_i \in X_k} \frac{x_i}{|X_k|}$
8: Store in C
9: Compute the pair-wise similarity SN between observations in $Xnum$ with **function** NUMERIC SIMILARITY ▷See Line (25)
10: Compute the pair-wise similarity SC between observations in $Xcat$ with **function** INTERSECTION ▷See Line (22)
11: Compute total similarity $1 - \frac{\text{SN+SC}}{\text{Number of variables}}$ and store in TS ▷See Equation 2

 # Process to calculate similarity between observations and decision classes
12: **for** decision class in data set **do**
13: **for** x in X **do**
14: Calculate numeric similarity $NumSim$ with the decision class vector of means in C using $\sum \frac{|x - C[\text{decision class}]|}{N}$
15: **for** x in X **do**
16: Calculate categorical similarity $CatSim$ with the mode of decision class using the **function**INTERSECTION(X,mode) ▷See Algorithm (1c)
17: Compute total similarity between observations and decision classes using $1 - \frac{\text{CatSim+NumSim}}{\text{Number of variables}}$ and store in TSC

 # Process to calculate membership values in the three fuzzy-rough regions
18: **for** each decision class in data set **do**
19: **for** each x in X **do** ▷See Algorithm (1e)
20: Compute membership values in positive, negative and boundary fuzzy-rough region per decision class using PROCESSOBJECT(x,decision class). Store in POS, NEG, BND
21: **return** POS, NEG, BND

Xcat. The distance function that is used is depicted in Algorithm (1c). *SN* stands for similarity between numeric observations and is a square-form matrix of pair-wise distances between numeric values in observations using *Xnum*. The distance function used is depicted in Algorithm (1d). Then, the two matrices are aggregated based on the HEOM distance formula as depicted in Equation (3). Finally the two matrices are added together and subtracted from 1 as per Equation 2. They store values in the $(0, 1]$ interval. The total similarity values are stored in the variable *ST* which is a square-form matrix of the total pair-wise distances between observations.

Lines (12) to (17) of Algorithm part (1b) calculate the pair-wise similarity between each observation and each decision class. *NumSim* stores the similarity between each observation in *Xnum* and the vector of mean values of the variables of the decision class it belongs to. Similarly, *CatSim* stores the similarity between each observation in *Xcat*

and the mode of each decision class. Finally, the two arrays are aggregated based on the HEOM distance formula as depicted in Equation (3) and the values are exponentiated as depicted in Equation (2). These values are stored in the variable *TSC*.

Finally, the Algorithm (1b) calculates the membership degree of each observation in each fuzzy-rough region per class. First, three empty 2-D arrays are created representing the positive, negative and boundary regions. The first dimension is the number of decision classes and the second dimension is the number of observations. The algorithm returns three two-dimensional arrays of membership values. These are the fuzzy-rough regions. Each array contains the membership values of each observation to the positive, negative and boundary regions. The number of arrays per region corresponds to the number of decision classes in the data set. Each element in the array corresponds to the membership value of each observation to the class in question. The higher the membership value, the more the instance belongs to the region of the assigned class.

Algorithm (1c) computes the distance between categorical observations where x is an observation of the data set. Y can be both an observation or the mode of the decision

Algorithm 1c Class object: computes membership values per region and decision class

Input: Observations x and y
Output: Sum of different values in x and y ▷See Equation 3
22: **function** INTERSECTION(x,y)
23: Measure categorical distance between vectors x and y ▷See Equation 4
24: **return** Sum of different values

class that x is compared to.

Algorithm (1d) computes the distance between numeric observations where x is an observation of the data set. Y can be both an observation or the mean values (defined

Algorithm 1d Class object: computes membership values per region and decision class

Input: Observations x and y
Output: Sum of different values between x and y
25: **function** NUMERIC SIMILARITY(x,y,N)
26: **return** $||x - y||_{L_1}$ ▷See Equation 4

by variable *C* in Algorithm (1a)) of each variable of the decision class that x is compared to.

Algorithm (1e) computes the membership degrees of each observation in the fuzzy-rough regions. This function follows Equation 6, Equation 7, Equation 8, Equation 9 and Equation 10. It makes use of three more functions: MEMBERSHIP, IMPLICATOR and CONJUCTION which are defined by Algorithm (1f). MEMBERSHIP function computes the similarity between individual instances and decision classes following Equation (1).

Listing 7 presents this process in Python environment. It can be found in Appendix 7. It is a class object that takes as input a data set and returns three lists of numbers each corresponding to the positive, negative and boundary region. Each list consists of as many arrays as the decision classes in a data set. Finally, the values of each array correspond to each observation in a data set and indicate the degree to which the observations belong to the positive, negative and boundary regions of a decision class.

Algorithm 1e Class object: computes membership values per region and decision class

Input: decision class and observation x
Output: Membership values to positive, negative and boundary regions
function PROCESSOBJECT(decision class, x)
    inf $\leftarrow 1$ ▷See inf in Equation (6)
    sup $\leftarrow 0$ ▷See sup in Equation (7)
    Compute membership value: MEMBERSHIP(decision class,x) ▷See Alg. (1f)
    Store membership value in $SimXCl$
    for y in X **do**
        $SimXY$ stores similarity between x and y SimXY from TS ▷See Line (11)
        $SimYCL$ computes MEMBERSHIP(decision class,y) ▷See Algorithm (1f)
        inf $\leftarrow min$(inf,IMPLICATOR($SimXCl \times SimXY, SimYCl$)) ▷Al. (1f)
        sup $\leftarrow min$(sup,CONJUNCTION($SimYCl \times SimXY, SimXCl$)) ▷Al. (1f)
    inf $\leftarrow min$(inf,$SimXCl$)
    sup $\leftarrow max$(sup,$SimXCl$)
    return inf, $1-$sup, sup - inf

Algorithm 1f Class object: computes membership values per region and decision class

function IMPLICATOR(a,b) ▷See Equation (6)
    return $\min(1 - \text{a} + \text{b}, 1)$
function CONJUNCTION(a,b) ▷See Equation (7)
    return $\min(\text{a} + \text{b} - 1, 0)$
function MEMBERSHIP(decision class, x) ▷See Equation (1)
    Compute similarity between x and decision class using values from TSC and following Equation 1
    return Membership value of x to decision class

3.3 Proposed bias quantification measures

In this section, two sets of measures that use fuzzy-rough sets to quantify bias in tabular data are proposed. These are the contribution of this thesis. The intuition behind the measures is as follows. The first set of measures is referred to as local change measures, whereas the second set of measures is referred to as global change measures. Both sets were developed on the basis of the lectures notes for the course Knowledge Representation under the MSc Data Science and Society program at Tilburg University (Nápoles, 2020) The intuition behind them is defined in the following sub-sections.

3.3.1 Local change measures

Let Ψ^* denote the set of protected features that might lead to bias and Ψ^+ the set of unprotected features that are not expected to involve any bias. Local change measures aim at quantifying how much the absence of a protected feature $\psi_i \in \Psi^*$ causes instances to move from a fuzzy-rough positive region to another.

The first local change measure quantifies the change between the positive region of the class k using all features and the positive region of the class j using all features in Ψ

but ψ_i. It is expressed in Equation (11),

$$\Theta_{P_k \to P_j}(\psi_i) = \frac{\sum_{x \in U} |\mu_{P_k \cap P_j}(x) - \mu_{P_k \cap \hat{P}_j}(x)|}{\sum_{x \in U} \mu_{P_k \cup P_j \cup \hat{P}_j}(x)} \tag{11}$$

where $P_k = POS_\Psi(X_k)$ denotes the fuzzy-rough positive region of the decision class k using all features of a data set and $\hat{P}_j = POS_{\Psi \setminus \{\psi_i\}}(X_j)$ is the fuzzy-rough positive region of the j-th class using all features in Ψ but ψ_i. The larger the value of Θ, the larger the change from the positive region of class k to the positive region of class j. The measure also reports an intersection that was not caused by the removal of a protected attribute (Nápoles, 2020) as the fuzzy-rough regions have a certain degree of overlap according to the original set of features.

The second local change measure quantifies the change between the positive region of the class k using all features and the boundary region of the class j using all features in Ψ but ψ_i. It is expressed in Equation (12),

$$\Theta_{P_k \to B_j}(\psi_i) = \frac{\sum_{x \in U} |\mu_{P_k \cap B_j}(x) - \mu_{P_k \cap \hat{B}_j}(x)|}{\sum_{x \in U} \mu_{P_k \cup B_j \cup \hat{B}_j}(x)} \tag{12}$$

where $P_k = POS_\Psi(X_k)$ denotes the fuzzy-rough positive region of the decision class k using all features of a data set and $\hat{B}_j = BND_{\Psi \setminus \{\psi_i\}}(X_j)$ is the fuzzy-rough boundary region of class j using all features in Ψ but ψ_i. The larger the value of Θ, the larger the change from the positive region of class k to the boundary region of class j. The intersection of two fuzzy sets X and Y is defined as $\mu_{X \cap Y} = min\{\mu_X(x), \mu_Y(x)\}, \forall x \in U$, whereas their union takes the form $\mu_{X \cup Y} = max\{\mu_X(x), \mu_Y(x)\}, \forall x \in U$.

3.3.2 Global change measures

Global change measures quantify how much the absence of a feature $\psi_i \in \Psi^*$ modifies the fuzzy-rough regions. For the sake of simplicity, difference between the minimum and the maximum of the fuzzy intersection and the fuzzy union respectively is used to normalize the global change measures (Nápoles, 2020).

The first global change measure computes the change between the fuzzy-rough positive region of the class k using all features and the fuzzy-rough positive region of the class k using all features in Ψ but ψ_i. It is expressed in Equation (13),

$$\Omega_P(\psi_i) = \sum_k \Omega_{P_k}(\psi_i) \tag{13}$$

where

$$\Omega_{P_k}(\psi_i) = \frac{\sum_{x \in U} |\mu_{P_k}(x) - \mu_{\hat{P}_k}(x)|}{|U|(\sup_{x \in U}(\mu_{P_k \cup \hat{P}_k}(x)) - \inf_{x \in U}(\mu_{P_k \cap \hat{P}_k}(x)))}$$

The change between the positive regions of both decision classes is aggregated. The larger the change, the larger the bias is expected to be.

The second global change measure computes the change between the fuzzy-rough boundary region of the class k using all features and the fuzzy-rough boundary region of the class k using all features in Ψ but ψ_i. It is expressed in Equation (14),

$$\Omega_B(\psi_i) = \sum_k \Omega_{B_k}(\psi_i) \tag{14}$$

where

$$\Omega_{B_k}(\psi_i) = \frac{\sum_{x \in U} |\mu_{B_k}(x) - \mu_{\hat{B_k}}(x)|}{|U|(\underset{x \in U}{sup}(\mu_{B_k \cup \hat{B_k}}(x)) - \underset{x \in U}{inf}(\mu_{B_k \cap \hat{B_k}}(x)))}$$

Again, the change between the boundary regions of both decision classes is aggregated. The larger the change, the larger the bias is expected to be. The third and last global change measure computes the change between the fuzzy-rough negative region of the class k using all features and the fuzzy-rough negative region of the class k using all features in Ψ but ψ_i. It is expressed in Equation (15),

$$\Omega_N(\psi_i) = \sum_k \Omega_{N_k}(\psi_i) \tag{15}$$

where

$$\Omega_{N_k}(\psi_i) = \frac{\sum_{x \in U} |\mu_{N_k}(x) - \mu_{\hat{N_k}}(x)|}{|U|(\underset{x \in U}{sup}(\mu_{N_k \cup \hat{N_k}}(x)) - \underset{x \in U}{inf}(\mu_{N_k \cap \hat{N_k}}(x)))}$$

The change between the negative regions of both decision classes is aggregated. The larger the change, the larger the bias is expected to be. The intuition behind these measures is that protected features (the ones suspected to cause bias) should not cause significant changes in the granular regions when they are excluded from the granulation step. The extend to which this happens can be understood as a bias indicator.

3.3.3 Computing the measures

This section introduces the algorithm that is created in Python environment to produce the output of the measures. First, a pre-processing algorithm is introduced to transform the data set into a form suitable for the measures algorithms. Next, the two sets of measures algorithms are introduced.

Membership values including and excluding protected attributes

The data-set first needs to be pre-processed in order to measure bias. The intuition behind pre-processing is as follows. The fuzzy-rough set algorithm that is introduced in Section 3.2 is utilized to compute the fuzzy-rough regions of the decision classes using the full set of variables in a data set. These values are stored in a variable. Next, the protected variable is excluded from the original data set. Using this 'reduced' version of the data set, the fuzzy-rough set regions are computed again and stored in a separate variable. The same process can be repeated for all the protected features in a data set.

The extent to which the fuzzy-rough regions of the 'full' and 'reduced' data-set differ is explored by the proposed measures.

The pre-processing algorithm is presented in Listing (7) developed in Python environment and can be found in Appendix (7). The pre-processing algorithm is also presented in the form of pseudo-code. Algorithm (2) illustrates its inner workings. Given that a data set has two protected attributes, the algorithm will create a dictionary object that includes three entries: the first represents the three fuzzy-rough regions accounting for the full data set, whereas the other two entries represent the fuzzy-rough regions for the two versions of the same data set where one of the protected attributes is excluded. The proposed measures receive the output of Algorithm (2) as input.

Algorithm 2 Pre-processing the data set

Input: Data set, protected attributes
Output: POS,BND,NEG per data set
function REGIONMAKER(data set,ProtAttr1,ProtAttr2)
  Compute POS, NEG, BND with FRS(data set).REGIONS() ▷See Alg. (1a)
  data set contains all attributes - store in $Full$
  Compute $POS1$, $NEG1$, $BND1$ with FRS(data set).REGIONS() ▷See Alg. (1a)
  data set contains all attributes but protected attribute 1 - store in $Attribute1$
  Compute $POS2$, $NEG2$, $BND2$ with FRS(data set).REGIONS() ▷See Alg. (1a)
  data set contains all attributes but protected attribute 2 - store in $Attribute2$
  Store $Full$, $Attribute1$, $Attribute2$ in Mix
  return Mix

Local Change measures algorithm

The algorithm computing the local change measures is presented in Listing (7) developed in Python environment and can be found in Appendix (7). The local change algorithm is also presented in the form of pseudo-code. Algorithm (3) illustrates its inner workings. Equation (11) is used twice in the algorithm given that there are two

Algorithm 3 Local Change Measures

Input: Output of Algorithm (1), list of protected attributes
Output: First and second local measures
function LOCALCHANGE(Mix,ProtAttr1,ProtAttr2)
  for each decision class **do**
    Compute local measure using POS and $POS1$ ▷See Equation 11
    Compute local measure using POS and $POS2$ ▷See Equation 11
  for each decision class **do**
    Compute local measure using POS and BND and $BND1$ ▷See Equation 12
    Compute local measure using POS and BND and $BND2$ ▷See Equation 12
  return Local measures per decision class

decision classes k and j in the data set. This happens as local change needs to be calculated for both decision classes. The same happens with Equation (12).

Global Change measures algorithm

The algorithm computing the local change measures (See section 3.3.2) is presented in Listing (7) developed in Python environment and can be found in Appendix (7). The local change algorithm is also presented in the form of pseudo-code. Algorithm (3) illustrates its inner workings. It makes use of Equations (13), (14) and (15).

Algorithm 4 Global Change Measures

Input: Output of Algorithm (1), list of protected attributes
Output: First, second and third global measures
function GLOBALCHANGE(Mix,ProtAttr1,ProtAttr2)
  for each decision class **do**
    Compute global measure using POS and $POS1$ ▷See Equation 13
    Compute global measure using POS and $POS2$ ▷See Equation 13
  for each decision class **do**
    Compute global measure using BND and $BND1$ ▷See Equation 14
    Compute global measure using BND and $BND2$ ▷See Equation 14
  for each decision class **do**
    Compute global measure using NEG and $NEG1$ ▷See Equation 15
    Compute global measure using NEG and $NEG2$ ▷See Equation 15
  return Global measures per decision class

4 Experimental Setup

This section presents the steps taken to quantify and measure bias in two data sets. First, the data sets are introduced including the results of exploratory data analysis and the data cleaning process. Both processes were conducted in R programming language (R Core Team, 2013) to explore and treat dependencies in the data, detect missing values and visualize the relationship between protected attributes and decision classes. Second, the experimental procedure is presented comprising of the creation of the three fuzzy-rough regions per data set and decision class followed by the application of the proposed measures.

4.1 Data

The two data sets upon which the proposed measures are tested are *German Credit* data set (Dua & Graff, 2017b) and *COMPAS* data set (Larson et al., 2017).

German Credit aims at classifying bank customers in terms of creditworthiness based on financial and personal information. The data set has 21 variables and 1000 observations. There is a binary decision class: 1 represents people who are eligible to receive a loan and 2 represents people who were categorized as not credit worthy. These values are re-coded to 0 and 1 respectively. Protected attributes are gender and age following Bellamy et al. (2018)'s analysis. During pre-processing, the nominal attribute *sex&marital status* was re-coded by effectively deleting any information regarding marital status and preserving only gender-related information as conducted by AIF360. The rest of the values were kept intact. Table 1 showcases a fraction of the data set. There are no missing values.

Table 1
Original values of *German Credit* data set

Exist. check account	Months	history credit	purpose credit	amount credit	savings account	employed since	installment rate	guarantors	residence since	property
A11	6	A34	A43	1169	A65	A75	4	A101	4	A121
A12	48	A32	A43	5951	A61	A73	2	A101	2	A121
A14	12	A34	A46	2096	A61	A74	2	A101	3	A121
A11	42	A32	A42	7882	A61	A74	2	A103	4	A122
A11	24	A33	A40	4870	A61	A73	3	A101	4	A124

age	installment plans	house ownership	existing credits	job	maintenance no people	phone registered	foreign	Loan eligibility	sex
67	A143	A152	2	A173	1	A192	A201	0	male
22	A143	A152	1	A173	1	A191	A201	1	female
49	A143	A152	1	A172	2	A191	A201	0	male
45	A143	A153	1	A173	2	A191	A201	0	male
53	A143	A153	2	A173	2	A191	A201	1	male

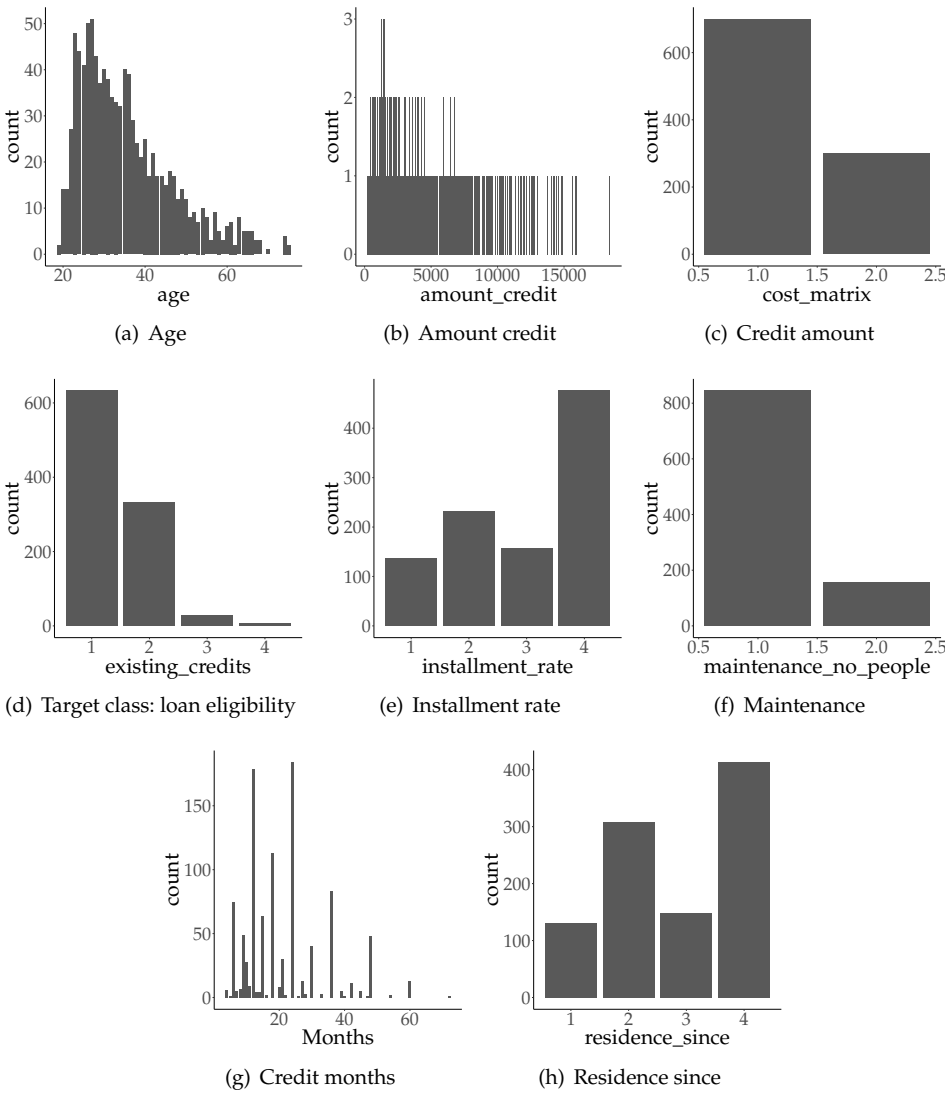
In an effort to visualize the distribution of the numeric variables and look for outliers, histograms are created and displayed in Figure 1. The distribution of categorical variables is illustrated in Figure 2a.

A correlation plot is created to explore dependencies between variables. It is visualized in Figure 3 There is a 8% correlation between applicants' gender and target class and a 9% correlation between age and the target class. The only variable that exhibits a higher correlation degree with the decision class is Existing checking account (35%). The correlation of the rest of the variables with the decision class is less than 25%. There are also a few other variables that correlate with one another by more than 30%. These are the number of months with the amount of credit (62%), historical credit with existing credits (44%), property with house ownership (35%) and with amount of credit (31%), house ownership with age (30%),and, finally, job with phone registration (38%).

The relation between the decision classes and the protected attributes was also explored. Table 4 illustrates that when it comes to gender, 64.8% of female applicants are deemed eligible to receive a loan. The equivalent rate for male applicants was 72.3%. Next, it illustrates that when it comes to age, 58% of applicants below 25 years old are deemed eligible to receive a loan. The equivalent rate for applicants older than 25 years old was 73%.

COMPAS is used to predict how likely are criminal offenders to re-offend. Protected attributes are race and gender (Bellamy et al., 2018). The data set was pre-processed in R programming environment following the steps taken by ProPublica (Larson et al., 2017). The code is exhibited in the Appendix. The original data set has 53 variables and 7214 observations. Target variable is *two year recidivism* where zero represents offenders who are not likely to re-offend and one the opposite. Protected attributes are race and gender following AIF360's example. All attributes containing missing values where removed with the exception of the variable *c charge desc* where the five missing values were replaced with the mode. This resulted in a reduced version of the data set containing 39 variables and 6172 variables. A sample of the post-processed data set is shown in Table 2.

In an effort to visualize the distribution of the numeric variables and look for outliers, histograms are created and displayed in Figure 5.
The distribution of categorical variables is illustrated in Figure 6.



(a) Age (b) Amount credit (c) Credit amount

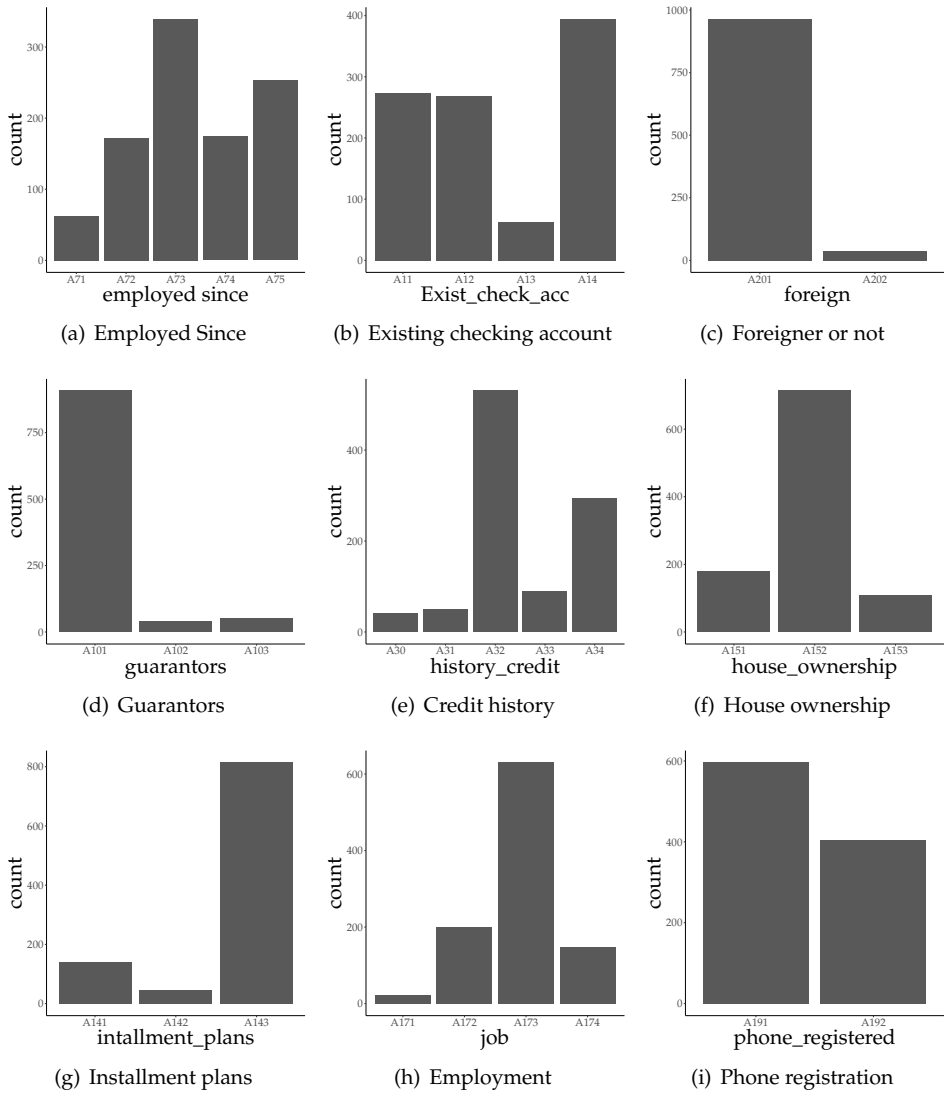
(d) Target class: loan eligibility (e) Installment rate (f) Maintenance

(g) Credit months (h) Residence since

Figure 1
German Credit: histogram of continuous variables

A correlation plot is created to explore dependencies between variables. It is visualized in Figure 7 There is a 1% correlation between offenders' gender and target class and a 13% correlation between race and the target class. Variables that are highly correlated with the target variable are the annual recidivism score (94%), end (78%) and event(79%).

The relation between the decision classes and the protected attributes was also explored. Table 8 illustrates that when it comes to gender, 47.9% of male inmates are more likely to re-offend which is contrasted to an equivalent value of 35.1% in women. Furthermore, when it comes to race, 47.69% of black inmates are likely to re-offend whereas the same figure for all other groups in that category does not exceed 39.09%.



(a) Employed Since (b) Existing checking account (c) Foreigner or not

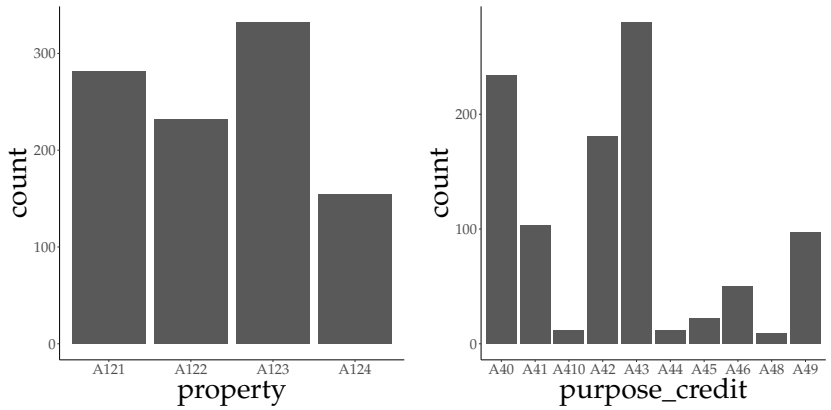
(d) Guarantors (e) Credit history (f) House ownership

(g) Installment plans (h) Employment (i) Phone registration

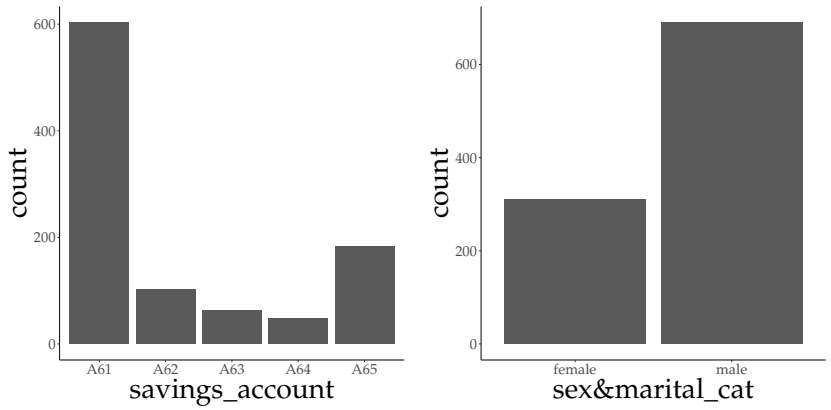
Figure 2
German Credit: histogram of categorical variables

4.2 Data preparation and creation of fuzzy-rough regions

Before creating the fuzzy-rough regions the data set values need to be pre-processed. All numeric attributes are scaled using the function MinMaxScaler (Nápoles, 2020) from the scikit-learn library such that all their values lie between zero and one. Next, the numeric features were standardized by removing the mean and scaling by unit variance using the function fit_transform from scikit-learn library. The instances were reordered: all instances that belong to decision class zero appear first followed by instances that belong to decision class one. Categorical variables are not dummy-coded or otherwise re-coded because that would alter the data set's dimensionality thus affecting the formation of



(a) Property ownership (b) Credit purpose



(c) Saving account (d) Gender

Figure 2a
German Credit: histogram of categorical variables

fuzzy-rough regions. Furthermore, in an effort to create model-independent measures, data pre-processing is kept to the absolute minimum. Listing (7) displays the algorithm that was developed in Python environment for the above-mentioned purpose and can be found in Appendix 7.
The values of the pre-processed data sets are shown in Tables 3 and 4.

The next step is to calculate the fuzzy-rough regions (positive, negative and boundary) per decision class using the pre-processed data sets. Algorithm 1a is used for that purpose. Each observation x in a data set is given a membership score corresponding to each different fuzzy-rough region and each different decision class. The higher the score, the more these instances belong to the decision class. Instances with a high membership score in the positive regions of a decision class belong to in to a higher extent. Conversely, instances with a higher score in the negative regions of a decision class belong to it to a lower extent. Finally, instances who exhibit a higher score in the boundary regions cannot be easily classified as members of the decision class in question.

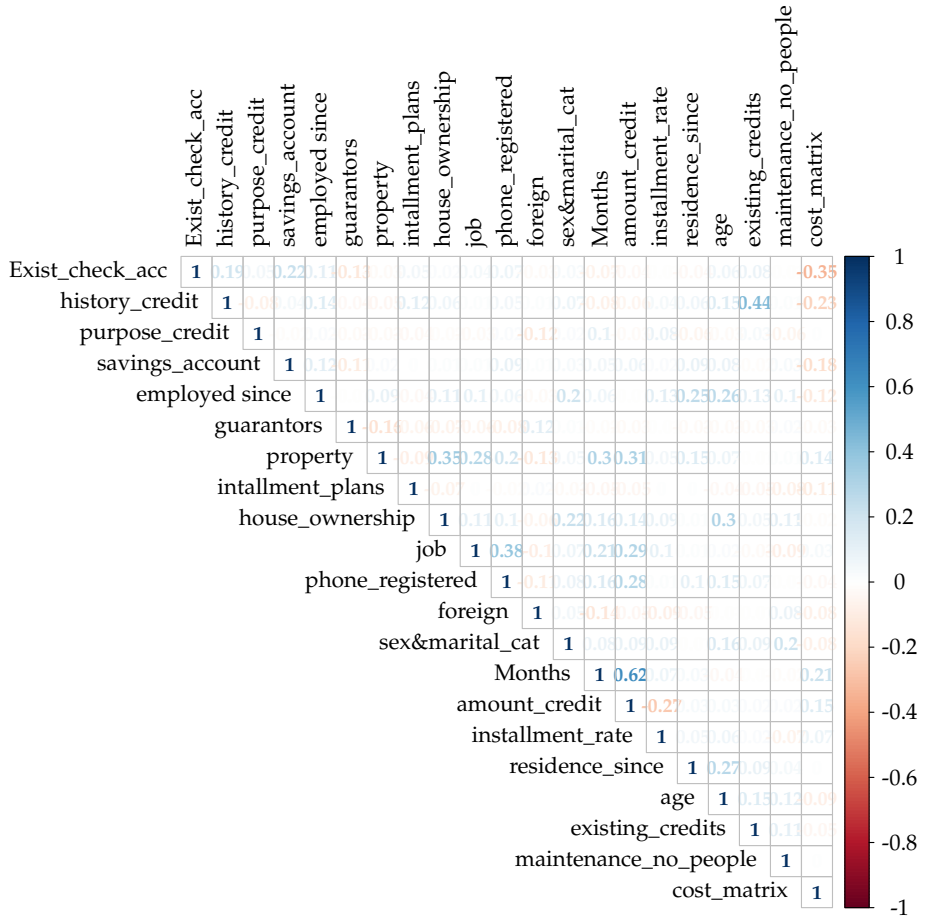
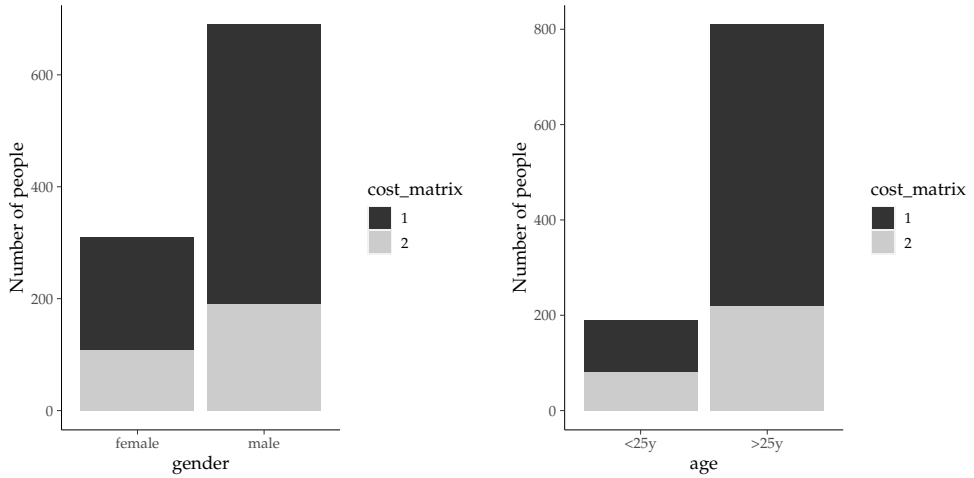


Figure 3
Correlation plot between the variables of *German Credit* data set. The decision class is 'cost_matrix'. Correlation between the protected attributes and the decision class does not exceed 9%.

The output of the algorithm is a dictionary object which contains three entries that correspond to each region: POS, NEG, BND. Each entry of the dictionary has two arrays: the first array contains the membership values of all instances with respect to decision class zero and the second the membership values of all instances with respect to decision class one. Algorithm 2 is used to calculate the fuzzy-rough regions of a version of the data sets that contains the full set of variables and versions that exclude one of the protected attributes each time. More specifically, two different versions of the *German Credit* were created: one were the protected attribute 'sex' was deleted and another where the protected attribute 'age' was deleted. The original *German Credit* data set and its two modified versions are used as inputs to the algorithm which then calculates the positive, negative and boundary regions for of the three data sets. The same process is



(a) Gender and target class (b) Age and target class

Figure 4
Compas: relationship between protected attributes and decision classes. (a) shows that 64.8% of females are eligible to receive a loan, whereas the percentage rises to 72.3% in the case of males. (b) 58% of people younger than 25 years old are eligible to receive a loan, whereas the percentage rises to 73% in the case the applicants are older than 25 years old.

followed for *COMPAS* data set. These membership values will be used as input to the proposed bias quantification measures.

Finally, the Algorithms 3 and 4 are used to measure bias on both data sets per protected attribute. They receive as input the three versions of the *German Credit* data set or, to put it alternatively, the dictionary created by Algorithm 2. The two protected attributes also need to be specified. The membership values calculated using the full set of variables are compared with and the membership values calculated using a reduced set of variables where one of the protected attributes has been omitted (first the attribute *age* and then *gender*). The same process is followed in the case of the *Compas data set* where the protected attributes are *race* and *gender*. Outputs are presented in Results section.

4.3 Baseline measures

The evaluation metrics that will be used as baseline measures are presented in Bellamy et al. (2018)'s work. The four bias quantification metrics that are utilized are presented below.

1. *Disparate impact* is computed by AIF360 as the ratio of the rate of the favorable outcome for the unprivileged group to that of the privileged group. A value below one implies higher benefit for the privileged group and a value above one implies a higher benefit for the unprivileged group.

Table 2
Original values of *COMPAS* data set

sex	age	race	juv fel count	decile score	juv misd count	juv other count	priors count	days before arrest	days from compas	charge degree
Male	69	Other	0	1	0	0	0	-1	1	F
Male	34	African-American	0	3	0	0	0	-1	1	F
Male	24	African-American	0	4	0	1	4	-1	1	F
Male	44	Other	0	1	0	0	0	0	0	M
Male	41	Caucasian	0	6	0	0	14	-1	1	F

c charge desc	decile score.1	score text	screening date	v decile score	v score text	priors count.1	start	end	event	two year recid
Assault	1	Low	2013-08-14	1	Low	0	0	327	0	0
Felony Battery	3	Low	2013-01-27	1	Low	0	9	159	1	1
Possess Cocaine	4	Low	2013-04-14	3	Low	4	0	63	0	1
Battery	1	Low	2013-11-30	1	Low	0	1	853	0	0
Burglary Tools	6	Medium	2014-02-19	2	Low	14	5	40	1	1

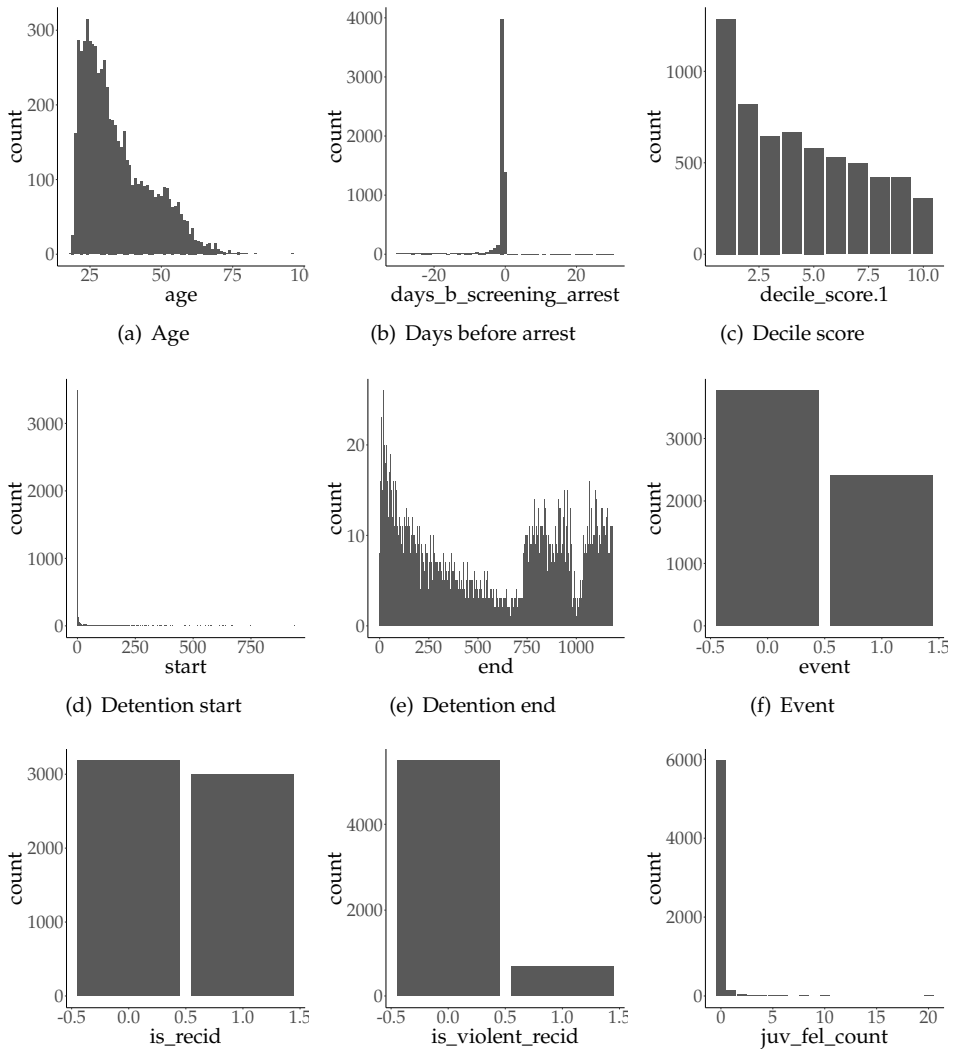
Table 3
Values of the *German Credit* data set after pre-processing

Exist. check account	Months	history credit	purpose credit	amount credit	savings account	employed since	installment rate	guarantors	residence since	property
A11	A34	A43	A65	A75	A101	A121	A143	A152	A173	A192
A14	A34	A46	A61	A74	A101	A121	A143	A152	A172	A191
A11	A32	A42	A61	A74	A103	A122	A143	A153	A173	A191
A14	A32	A46	A65	A73	A101	A124	A143	A153	A172	A192
A14	A32	A42	A63	A75	A101	A122	A143	A152	A173	A191

age	installment plans	house ownership	existing credits	job	maintenance no people	phone registered	foreign	Loan eligibility	sex
A201	male	0.029412	0.050567	1.000000	1.000000	0.857143	0.333333	0.0	0.0
A201	male	0.117647	0.101574	0.333333	0.666667	0.535714	0.000000	1.0	0.0
A201	male	0.558824	0.419941	0.333333	1.000000	0.464286	0.000000	1.0	0.0
A201	male	0.470588	0.484483	0.333333	1.000000	0.285714	0.000000	1.0	0.0
A201	male	0.294118	0.142236	0.666667	1.000000	0.607143	0.000000	0.0	0.0

2. *Statistical parity difference* is computed by AIF360 as the difference of the rate of favorable outcomes received by the unprivileged group to the privileged group. The ideal value of this metric is 0.

3. *Equal opportunity difference* is by AIF360 computed as the difference of true positive rates between the unprivileged and the privileged groups(Verma & Rubin, 2018b). The true positive rate is the ratio of true positives to the total number of actual positives for a given group. It is based on the outcomes of a machine learning model's predictions. The ideal value is 0. A value below zero implies higher benefit for the privileged group and a value above zero implies higher benefit for the unprivileged group (Hardt et al., 2016).



(a) Age (b) Days before arrest (c) Decile score

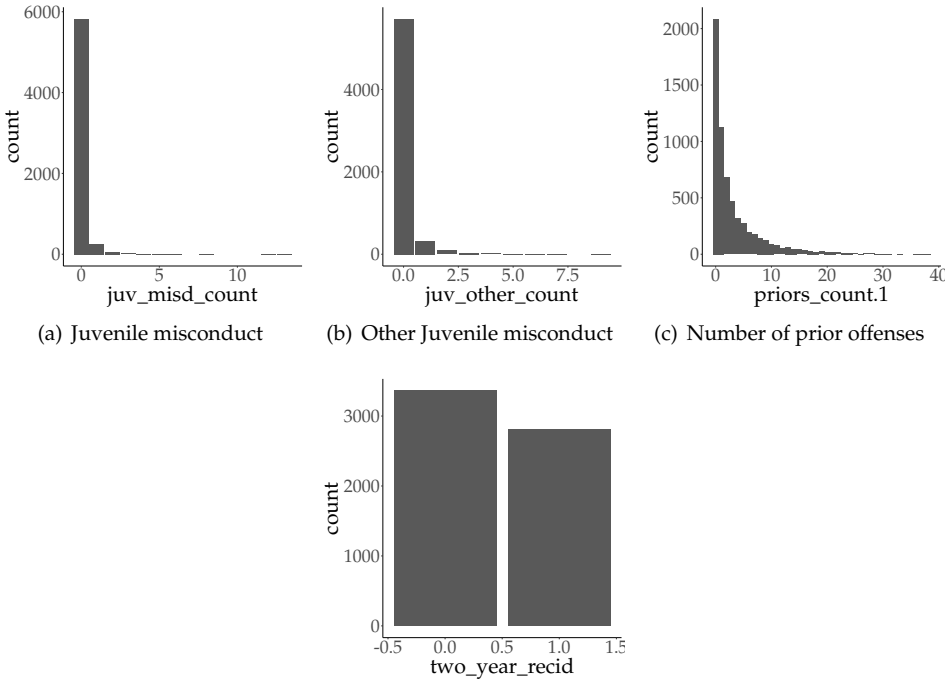
(d) Detention start (e) Detention end (f) Event

(g) Risk of recidivism: 1-yes, 0-no (h) Risk of violent recidivism: 1-yes, 0-no (i) Juvenile felonies

Figure 5
Compas data set: This figure presents the histograms of each numeric variable in the data sets

4. The *average odds difference* is computed as the average difference of false positive rate (false positives divided by the total number of negatives) and true positive rate (true positives divided by the total number of positives) between unprivileged and privileged groups (Bellamy et al., 2018). The ideal value of this metric is 0. A value below zero implies higher benefit for the privileged group and a value above zero implies higher benefit for the unprivileged group.

The above-mentioned measures are translated into code by AIF360 and belong in the *Metric* class and its sub classes. More specifically, the *BinaryLabelDatasetMetric*



(a) Juvenile misconduct (b) Other Juvenile misconduct (c) Number of prior offenses

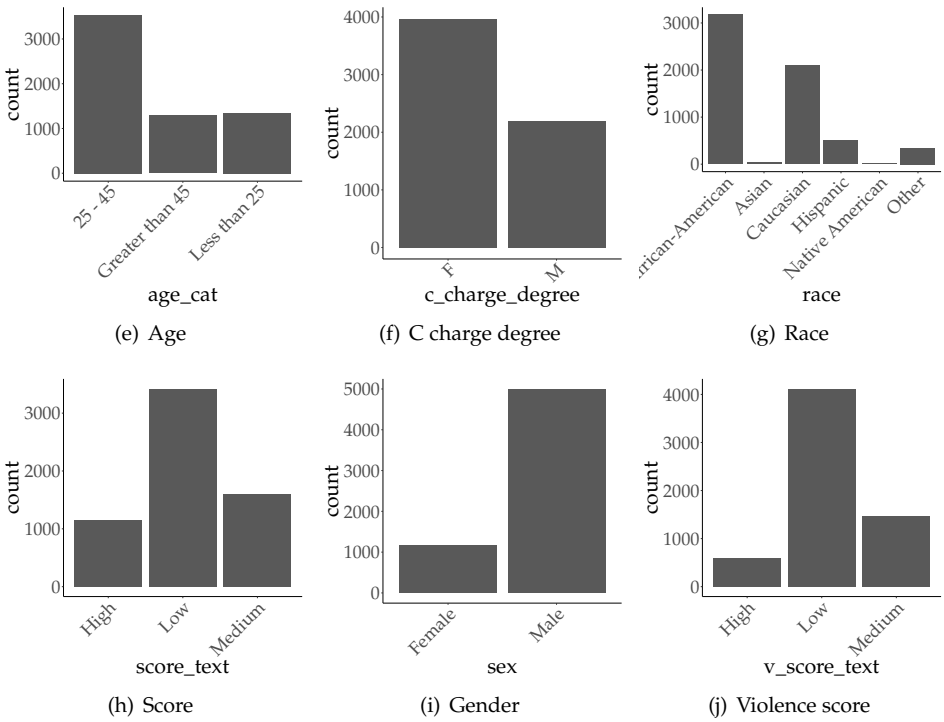
(d) Before

Figure 5a
Compas: histograms of numeric variables

Table 4
Values of *Compas* data set after pre-processing

sex	age	race	juvenile felonies	decile score	juv misd count	juv other	priors count	days b. arrest	days fr. compas	charge degree
Male	Other	F	Firearm Assault	Low	2013-08-14	Low	0.653846	0.0	0.000000	0.0
Male	Other	M	Battery	Low	2013-11-30	Low	0.333333	0.0	0.000000	0.0
Male	Other	F	arrest case no charge	Low	2013-08-30	Low	0.320513	0.0	0.333333	0.0
Female	Caucasian	M	Battery	Low	2014-03-16	Low	0.269231	0.0	0.000000	0.0
Male	Caucasian	F	Poss MDMA	Low	2013-11-26	Low	0.115385	0.0	0.333333	0.0

charge desc	decile score.1	score text	screening date	v decile score	v score text	priors count.1	start	end	event	two year recid
0.0	0.000000	0.483333	0.000105	0.000000	0.000000	0.000000	0.000000	0.275717	0.0	0
0.0	0.000000	0.500000	0.000000	0.000000	0.000000	0.000000	0.001067	0.719224	0.0	0
0.0	0.078947	0.483333	0.000105	0.333333	0.222222	0.078947	0.000000	0.223440	0.0	0
0.0	0.000000	0.483333	0.000105	0.000000	0.000000	0.000000	0.002134	0.629848	0.0	0
0.0	0.000000	0.483333	0.000105	0.333333	0.333333	0.000000	0.000000	0.722597	0.0	0



(e) Age (f) C charge degree (g) Race

(h) Score (i) Gender (j) Violence score

Figure 6
Compas data set: This figure presents the histograms of each categorical variable.

class is used to calculate *disparate_impact* and *statistical_parity_difference*. These two measures are applied in the original version of the data sets. Two additional measures, *average_odds_difference* and *equal_opportunity_difference*, belong to the *ClassificationMetric* class. When the latter two measures are applied, a classifier is trained on a fraction of the original data set (training set) to make predictions regarding the label. Then, the classifier makes predictions on the rest of the data and its outputs are used to compute the metrics. The four bias measures are calculated for each protected attribute in the two data sets in Python environment following example Python scripts[2] offered in AIF360's account in Github (Bellamy et al., 2018). The script can be found in Listing 7 in Appendix 7.

4.4 Evaluation criteria

The final step in the experimental process is comparing the the outputs of the bias quantification measures with the baseline measures. To that end, two points need to be considered:

2 The example Python scripts are (1) demo_reject_option_classification.ipynb and (2) demo_reweighing_preproc.ipynb

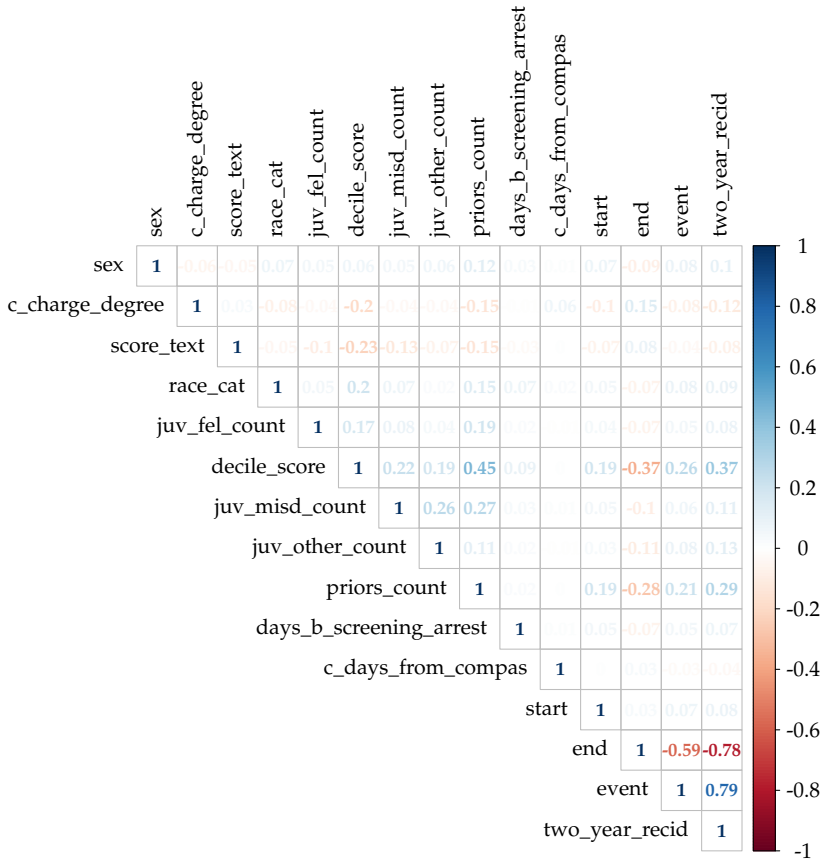
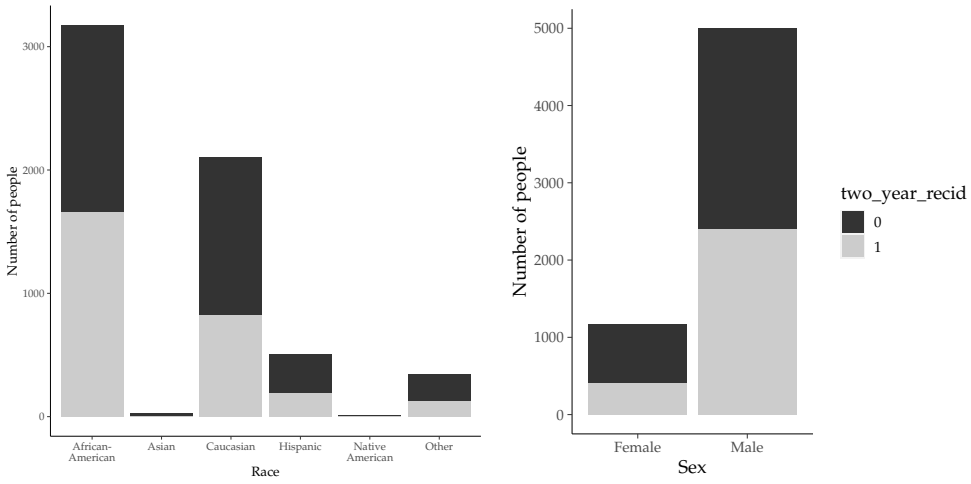


Figure 7
Correlation plot between the variables of *Compas* data set. The decision class is 'two_year_recid'. Correlation between the protected attributes and the decision class does not exceed 12%.

- First, the proposed and the baseline measures are different types of data. The former are computed using the membership values of each observation in a data set. The latter represent either differences among groups (privileged and unprivileged groups considering the protected attribute and the decision class) or differences between the true and false positive prediction rates. Moreover, the scales of all measures differ. The outputs of three of the baseline measures vary between -1 and 1. The values of the fourth baseline measure, *disparate impact*, can be either underneath or above 1. The values of the proposed measures range between 0 and 1. Therefore, it is impossible to directly compare them as they represent different types of data expressed in different scales.
- Second, another factor that needs to be considered is that the proposed measures do not capture the direction of the bias contrary to all four baseline measures. It is



(a) Gender and target class (b) Race and target class

Figure 8
Compas: relationship between protected attributes and decision classes. (a) shows that 47.69% of offenders of African-American origin are likely to re offend, whereas the percentage drops to 39.09% or lower when it comes to Caucasian or other races. (b) shows that 35.1% of females are likely to re-offend, whereas the percentage rises to 47.9% in the case of males.

impossible to tell whether the data set is biased towards the privileged or the unprivileged group since no such distinction is made during the computation of the membership values. Instead, they express the absolute change in the fuzzy-rough set regions per decision class after a protected attribute is excluded. Therefore, the baseline measures are interpreted based on their absolute value.

After considering the above two points, the values of the proposed measures are compared based on their (1) trend and (2) relative magnitude. Comparison in terms of trend means that if baseline measures assign more bias to one attribute than the other, then the proposed measures should also quantify more bias towards the same attribute. The relative magnitude refers to the relative difference of the amount of bias between protected attributes within the same data set. Finally, the outputs of the proposed measures cannot be compared between data sets. The data sets have a different number of variables which means that they differ in terms of dimensionality.

5 Results

This section is dedicated to: (1) presenting the output values to the baseline measures, (2) presenting and visualising the membership values in each fuzzy-rough set region computed for each data set, and, finally, (3) presenting the outputs of the proposed bias quantification metrics for each data set in relation to the baseline measures.

5.1 Baseline measures

The outputs of the four baseline metrics that measure the amount of bias against each of the protected attributes in both data sets are illustrated in Table (**??**). The *'protected*

Table 5
Values of the baseline measures per data set. If there is no bias in the data, the ideal values of all metrics except disparate impact is zero. The ideal value of the disparate impact should be one.

Table (**??**)

Protected attribute	Unprivileged	Statistical Parity Difference	Disparate Impact	Equal Opportunity Difference	Average Odds Difference
German Credit data set					
Sex	Female	-0.002500	0.994800	0.0400	-0.0071
Age	Young	-0.282800	0.469200	-0.287400	-0.25080
Percentage Difference		-196.494918	71.803279	-264.672595	-188.98798
Compas data set					
Sex	Male	-0.272400	0.663100	-0.1392	-0.2439
Race	Non-Caucasian	-0.249400	0.660000	-0.1877	-0.1927

attribute' column indicates the protected attributes towards which the data set is biased. The '*Unprivileged*' column states which population group within the protected attribute is likely to experience discrimination. Negative values (with the exception of '*Disparate Impact*') indicate that discrimination is directed towards the unprivileged group. It is worth noting that '*Equal Opportunity Difference*' shows a slight bias against the privileged group that is males in *German Credit* data set since the value is positive. If the values of '*Disparate Impact*' exceed 1, then the data set is biased against the privileged group which is not the case here. The industry standard dictates that there is an unacceptable amount of bias if the unprivileged group receives an outcome of less than 80% (Gastwirth & Miao, 2009). Finally, the rows '*Percentage Difference*' display the percentage difference between the two protected attributes per data set. Measuring this difference quantifies the amount to which the two protected attributes differ in terms of bias. It merely serves as an indication when comparing the outputs of the baseline measures between each data set.

When comparing the trend of the results of the four baseline measures with each other per data set is it observed they are not fully aligned in terms of trend and magnitude. In *German Credit* data set, all measures exhibit a higher bias towards protected attribute *age* than *sex* but to a different extent as shown by the percentage difference between them. '*Equal Opportunity Difference*' displays the largest difference between the two and '*Disparate impact*' the lowest one. In *Compas* data set, the measures measures generally exhibit a similar amount of bias towards both protected attributes, *race* and *sex*. '*Equal Opportunity Difference*' shows a slightly higher amount of bias towards *race*. The opposite trend is measured by '*Statistical Parity Difference*' and '*Average Odds Difference*'. The trends in the proposed measures per the protected attribute are going to be compared with the above-mentioned trends.

5.2 German data set

All outputs related to *German Credit* data set are exhibited here. A small fraction of the membership values per fuzzy-rough region and decision class is presented first. Then, graphs depicting the trend of all membership values are provided to illustrate the extent of indiscernibility between fuzzy-rough regions. Finally, the outputs of the local and global measures are presented in relation with the baseline measures.

5.2.1 Membership values

As described in Experimental Setup, Algorithms 1a to 1e assign membership values to each observation in a data set. Algorithm 2 computes the membership values of three versions of each data set: one containing the full set of attributes, one containing all but the first protected attribute and one containing all but the second protected attribute. Table 6 illustrates the membership degrees of observations 697 to 702.

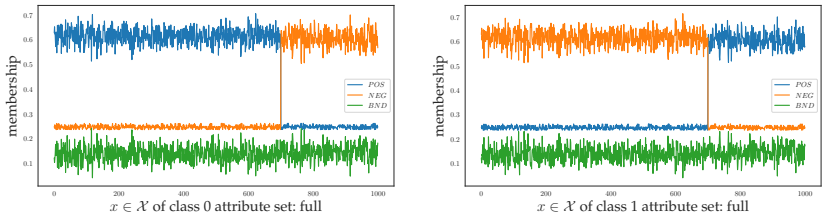
Table 6
German credit data set: membership degrees values per region (positive, negative and boundary) and decision class calculated using three variations of the data set. Decision class 0 represents people eligible to receive a loan and decision class 1 represents the opposite

Data set version 1: Contains the full set of attributes							
Index	POS_class0	POS_class1	NEG_class0	NEG_class1	BND_class0	BND_class1	Decision class
697	0.639972	0.260243	0.260243	0.622881	0.099785	0.116876	0 (Loan)
698	0.577366	0.234289	0.234289	0.605151	0.188345	0.160560	0 (Loan)
699	0.638239	0.251766	0.251766	0.629354	0.109995	0.118880	0 (Loan)
700	0.257274	0.575763	0.560161	0.257274	0.182565	0.166963	1 (No Loan)
701	0.231870	0.645750	0.651921	0.231870	0.116209	0.122380	1 (No Loan)
702	0.239292	0.586011	0.594866	0.239292	0.165843	0.174697	1 (No Loan)
Data set version 2: Excludes the protected attribute *age*							
Index	POS_class0	POS_class1	NEG_class0	NEG_class1	BND_class0	BND_class1	Decision class
697	0.627454	0.261015	0.261015	0.604748	0.111531	0.134237	0 (Loan)
698	0.581545	0.233285	0.233285	0.610934	0.185170	0.155780	0 (Loan)
699	0.645360	0.251904	0.251904	0.635953	0.102736	0.112143	0 (Loan)
700	0.257274	0.573713	0.559721	0.257274	0.183005	0.169013	1 (No Loan)
701	0.230486	0.652954	0.659397	0.230486	0.110117	0.116561	1 (No Loan)
702	0.238480	0.590381	0.599829	0.238480	0.161691	0.171139	1 (No Loan)
Data set version 3: Excludes the protected attribute *gender*							
Index	POS_class0	POS_class1	NEG_class0	NEG_class1	BND_class0	BND_class1	Decision class
697	0.644746	0.261320	0.261320	0.627411	0.093934	0.111269	0 (Loan)
698	0.570437	0.233662	0.233662	0.600052	0.195901	0.166286	0 (Loan)
699	0.645489	0.251492	0.251492	0.636773	0.103019	0.111735	0 (Loan)
700	0.258029	0.576036	0.558836	0.258029	0.183135	0.165936	1 (No Loan)
701	0.230575	0.648567	0.660016	0.230575	0.109409	0.120858	1 (No Loan)
702	0.238968	0.586100	0.594824	0.238968	0.166208	0.174932	1 (No Loan)

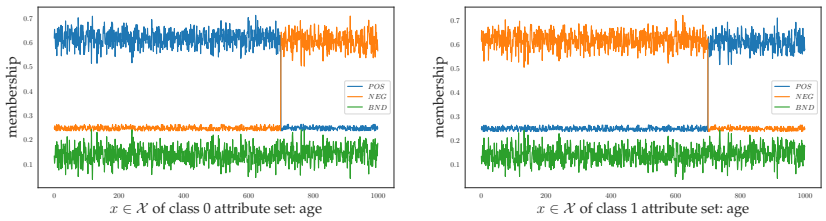
Observations 697 to 699 that belong to decision class 0 (credit-worthy) and observations 700 to 702 to decision class 1 (non credit-worthy). In other words, as the decision classes of both data sets are binary, each observation per data set receives six membership values in total. The first two values refer to the fuzzy-rough positive regions of decision class zero and one, the next two values refer to the negative regions of decision class zero and one, and, finally, the last two values correspond to the boundary regions of the two decision classes. Adding up the membership values the three regions per decision class results to one. Finally, one observes that the membership degrees of each observation

slightly change when the protected attributes are excluded. The extent of this change is explored by the proposed measures.

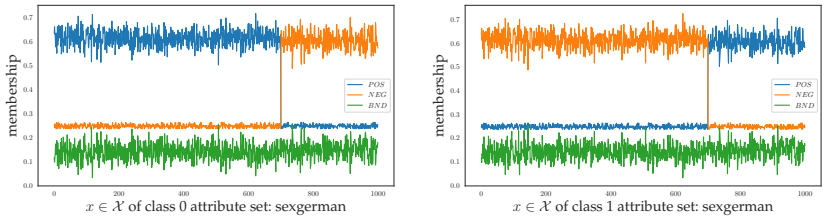
All membership values are vizualized in Figure 9 where x axis represents the membership degree and y axis represents each observation. The graphs depict the



(a) Membership values in the positive, negative and boundary regions computed using the full set of attributes



(b) Membership values in the positive, negative and boundary regions computed using the set of attributes without attribute age



(c) Membership values in the positive, negative and boundary regions computed using the set of attributes without attribute gender

Figure 9
German Credit data set: membership values. Graphs on the left side represent decision class 0 (eligible to receive a loan) and graphs on the right side decision class 1 (not eligible to receive a loan). y axis represents the number of observations and x axis represents the number the membership values.

membership degrees as generated for each different version of *German Credit* data set. Graphs on the left side represent decision class 0 (credit-worthy) and graphs on the right side decision class 1 (non credit-worthy). The blue line represents the extent to which an observation belongs to the decision class, the orange line the extent to which an observation does not belong to the decision class and the green line the extent to which an observation could belong to the decision class.

5.2.2 Measures outputs and comparison with baseline measures

The three versions of *German Credit* data set where used as input to the proposed measures algorithms (Algorithms 3 and 4) and the outputs can be viewed in Table 7. The protected attribute column refers to the attribute that is excluded from the data set

Table 7
German Credit: results of local and global measures measures. Class 0 refers to people eligible to receive a loan and class 1 to people deemed not-credit-worthy. The protected attribute column refers to the attribute that is excluded from the data set. Local measures follow Equation 11 and Equation 12. Global measures follow Equation 13, Equation 14 and Equation 15.

Local measures results				
Protected attribute	POS→POS dec. class 0→1	POS→POS dec. class 1→0	POS→BND dec. class 0→1	POS→BND dec. class 1→0
Age	0.000835	0.000321	0.058769	0.020796
Sex	0.000596	0.000213	0.070511	0.025825

Global measures results			
Protected attribute	POS change dec. class 0 and 1	NEG change dec. class 0 and 1	BND change dec. class 0 and 1
Age	0.009911	0.010001	0.037185
Sex	0.020519	0.020399	0.080683

Baseline metrics					
Protected attribute	Unprivileged	Statistical Parity Diff.	Disparate Impact	Equal Opportunity Diff.	Average Odds Diff.
Age	Young	-0.282800	0.469200	-0.2874	-0.2508
Sex	Female	-0.002500	0.994800	0.0400	-0.0071

when computing the measures. Regarding the local measures, column *'POS→POS dec. class 0→1'* quantifies the extent to which observations move from the positive region of class 0 to the positive region of class 1. When a protected attribute is suppressed, a value closer to zero means that less observations are changing decision classes. Column *'POS→POS dec. class 1→0'* captures the same notion but in the opposite direction. Column *'POS→BND dec. class 0→1'* represents the extent to which observations move from the positive region of class 0 to the boundary region of class 1. Column *'POS→BND dec. class 1→0'* represents the same concept but in the opposite direction. Regarding the global measures, column *'POS change dec. class 0 and 1'* expresses the total change in the membership values between positive regions of the same decision class in the absence of the protected attributes *age* and *sex*. The values of *'NEG change dec. class 0 and 1'* and *'BND change dec. class 0 and 1'* are computed in the same way measuring change in the membership values between the same decision class for the negative and boundary regions.

The main observations based on the results shown in Table 7 are presented below:

- **Local measures**: Local measures follow Equation 11 and Equation 12

- **Comparing the values between protected attributes**: The values generated by each measure differ per protected attribute which means that the data set is more biased towards one of the protected attributes than the other.

  Comparing trend and magnitude of values as we shift between positive regions ('POS→POS'): The protected attribute *age* has larger values in both 'POS→POS dec. class 0→1' and 'POS→POS dec. class 1→0' (class 0→1: 0.000835, class 1→0: 0.000321) than the protected attribute *sex* (class 0→1: 0.000213, class 1→0: 0.000141). This means that as we move from the group of credit-worthy individuals (advantaged group) to the group of non-credit worthy individuals (disadvantaged group) when oppressing attribute *age* more observations are classified as non-credit worthy than when protected attribute *sex*. This aligns with the trend in the baseline measures that show higher bias against *sex*.

  Comparing trend and magnitude of values as we shift from positive to boundary regions ('POS→BND'): The opposite trend is reported when we move from positive to boundary regions. Protected attribute *sex* has larger values in both *'class 0→1'* and *'class 1→0* than the protected attribute *age*.
- **Comparing magnitude of values between 'POS→POS' and 'POS→BND'**: The values show that there is a much larger movement in terms of magnitude as we move from positive to boundary regions compared to the movement between positive regions when suppressing both protected attributes. This means that there is a higher degree of classification uncertainty associated with the boundary regions compared to classification certainty as expressed in the positive regions.
- **Comparing movement from one decision class to another**: When comparing 'POS→POS dec. class 0→1' and 'POS→POS dec. class 1→0, the values of the former (age: 0.000438, sex: 0.000364) are greater than the values of the latter (age: 0.000165, sex: 0.000141) when it comes to both protected attributes. The same trend is evident when comparing 'POS→BND dec. class 0→1' and 'POS→BND dec. class 1→ between protected attributes.

- **Global measures**: Global measures follow Equation 13, 14 and 15.
  - **Total change in the positive regions** The results of the proposed measures contradict with the results of the baseline measures. The total change in the positive regions when protected attribute *sex* is suppressed is larger than the change when protected attribute *age* is suppressed. All baseline measures exhibit the exact opposite trend.
  - **Total change in the negative regions** The results follow a similar trend with the results of the first global measure. Contrary to the trend in the baseline measures they express higher bias towards *sex* than *age*.. Furthermore, the magnitude of the change in the membership values to both the positive and negative regions is similar when both protected attributes are suppressed.
  - **Total change in the boundary regions** The values of this measure are higher than the values of the first two measures thus indicating greater uncertainty than certainty in classifying observations. The values of this measure per protected attribute follow the same trend as the first two measures: there is more uncertainty when protected attribute *sex* is suppressed.

The next sub-section examines the results of *Compas* data set in a similar manner.

5.3 Compas data set

All outputs related to *Compas* data set are exhibited here. First, the membership values of each observation per fuzzy-rough region and decision class are presented and visualized. Then, the outputs of the proposed measures are presented.

5.3.1 Membership values

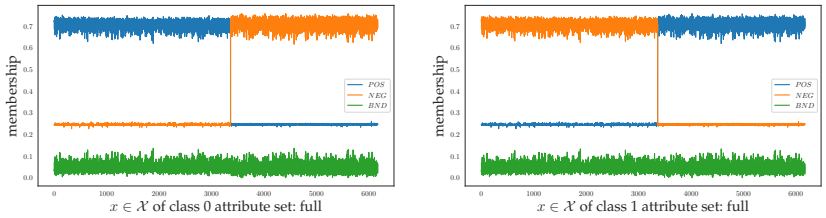
Table 8 illustrates the membership degrees of observations 3360 to 3365. Observations

Table 8
Compas data set: membership values per positive, negative and boundary region and decision class calculated using three versions of the data set. Decision class 0 means 'unlikely to re-offend' and decision class 1 means 'likely to re-offend'

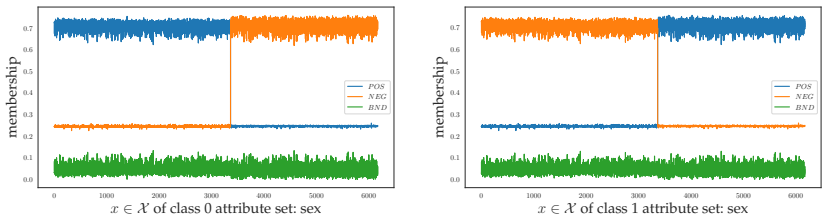
Data set version 1: Contains the full set of attributes							
Index	POS_class0	POS_class1	NEG_class0	NEG_class1	BND_class0	BND_class1	Decision class
3360	0.753021	0.232921	0.232921	0.753113	0.014058	0.013965	0 (No risk of recid.)
3361	0.709259	0.229155	0.229155	0.716705	0.061586	0.054140	0 (No risk of recid.)
3362	0.759577	0.230706	0.230706	0.764157	0.009717	0.005137	0 (No risk of recid.)
3363	0.239759	0.751943	0.745197	0.239759	0.015044	0.008298	1 (Risk of recid.)
3364	0.248769	0.718046	0.706528	0.248769	0.044702	0.033184	1 (Risk of recid.)
3365	0.234091	0.765909	0.765909	0.234091	0.000000	0.000000	1 (Risk of recid.)
Data set version 2: Excludes the protected attribute *race*							
Index	POS_class0	POS_class1	NEG_class0	NEG_class1	BND_class0	BND_class1	Decision class
3360	0.759518	0.232037	0.232037	0.759580	0.008445	0.008383	0 (No risk of recid.)
3361	0.694034	0.229155	0.229155	0.701345	0.076811	0.069499	0 (No risk of recid.)
3362	0.755950	0.229650	0.229650	0.759714	0.014400	0.010636	0 (No risk of recid.)
3363	0.239230	0.738350	0.731758	0.239230	0.029012	0.022419	1 (Risk of recid.)
3364	0.248709	0.703729	0.691934	0.248709	0.059357	0.047562	1 (Risk of recid.)
3365	0.234091	0.765909	0.765909	0.234091	0.000000	0.000000	1 (Risk of recid.)
Data set version 3: Excludes the protected attribute *gender*							
Index	POS_class0	POS_class1	NEG_class0	NEG_class1	BND_class0	BND_class1	Decision class
3360	0.759518	0.232037	0.232037	0.759580	0.008445	0.008383	0 (No risk of recid.)
3361	0.714683	0.228032	0.228032	0.722471	0.057285	0.049497	0 (No risk of recid.)
3362	0.745685	0.230706	0.230706	0.750009	0.023608	0.019285	0 (No risk of recid.)
3363	0.239230	0.758371	0.751377	0.239230	0.009393	0.002398	1 (Risk of recid.)
3364	0.248709	0.723655	0.711689	0.248709	0.039602	0.027636	1 (Risk of recid.)
3365	0.233192	0.766808	0.766808	0.233192	0.000000	0.000000	1 (Risk of recid.)

3360 to 3362 were originally classified to decision class 0 (unlikely to re-offend) and observations 3363 to 3365 to decision class 1 (likely to re-offend). Again, one observes that the membership values slightly change when the protected attributes are excluded. The amount of this change is explored by the proposed measures.

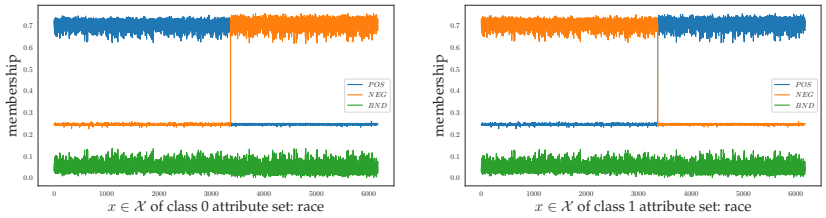
All membership values are vizualized in Figure 10 where x axis represents the membership degree and y axis represents each observation. The graphs depict the membership degrees as generated for each different version of *Compas* data set. Graphs on the left side represent decision class 0 (unlikely to re-offend) and graphs on the right side decision class 1 (likely to re-offend). The blue line represents the extent to which an observation belongs to the decision class, the orange line the extent to which an observation does not belong to the decision class and the green line the extent to which an observation could belong to the decision class.



(a) Membership values in the positive, negative and boundary regions computed using the full set of attributes



(b) Membership values in the positive, negative and boundary regions computed using the set of attributes without attribute age



(c) Membership values in the positive, negative and boundary regions computed using the set of attributes without attribute gender

Figure 10
Compas data set: membership values. Graphs on the left side represent decision class 0 (not likely to re-offend) and graphs on the right side decision class 1 (likely to re-offend). y axis represents the number of observations and x axis represents the number the membership values.

5.3.2 Local and global measures outputs

The three versions of *Compas* data set are used as input to algorithms (Algorithms 3 and 4) and the outputs can be viewed in Table 9. The main observations based on the results are presented below:

- **Local measures**: Local measures follow Equation 11 and Equation 12
  - **Comparing values between protected attributes**: The values generated by each proposed measure differ per protected attribute. The trends also differ between each measure: some measures report more bias towards one of the

Table 9
Compas: results of local and global measures. Class 0 refers offenders who are not likely to re-offend and class 1 the opposite notion. The protected attribute column refers to the attribute that is excluded from the data set. Local measures follow Equation 11 and Equation 12. Global measures follow Equation 13, Equation 14 and Equation 15.

Local measures results				
Protected attribute	POS→POS dec. class 0→1	POS→POS dec. class 1→0	POS→BND dec. class 0→1	POS→BND dec. class 1→0
race	0.000253	0.000272	0.182657	0.007393
sex	0.000480	0.000370	0.178910	0.007339

Global measures results			
Protected attribute	POS change dec. class 0 and 1	NEG change dec. class 0 and 1	BND change dec. class 0 and 1
race	0.009486	0.008988	0.078839
sex	0.008133	0.007750	0.057711

Baseline metrics					
Protected attribute	Unprivileged	Statistical Parity Diff.	Disparate Impact	Equal Opportunity Diff.	Average Odds Diff.
Race	Non-Caucasian	-0.249400	0.660000	-0.1877	-0.1927
Sex	Male	-0.272400	0.663100	-0.1392	-0.2439

protected attributes and others towards another. The baseline measures also exhibit the same inconsistency.

Comparing trend and magnitude as we shift between the positive regions ('POS→POS') The protected attribute *sex* has larger values in '*POS→POS dec.class 0→1*', but a lower value in '*POS→POS dec.class 1→0*. The baseline measures '*Statistical Parity Difference*' and '*Average Odds Difference*' follow the same trend. The same happens in '*POS→POS dec.class 0→1*': the protected attribute *race* seems to be less biased than *sex*.

Comparing trend and magnitude of values as we shift from the positive to boundary regions ('POS→BND') Column '*POS→BND dec.class 0→1*' has slightly larger values for attribute *race* than attribute *sex* which aligns with the baseline measure '*Equal Opportunity Difference*'. Column '*POS→BND dec.class 1→0*' reports the same value towards both protected attributes, a trend that aligns with the baseline measure '*Disparate Impact*'.

- **Comparing the magnitude of values between 'POS→POS and 'POS→BND**: There is a much larger movement in terms of magnitude as we move from positive to boundary regions compared to the movement between positive regions after suppressing both protected attributes. The same trend is also observed in *German Credit*.
- **Comparing movement from one decision class to another**: When comparing 'POS→BND dec. class 0→1' and 'POS→BND dec. class 1→0, the values of the former are greater than the values of the latter when it comes to both

protected attributes. This aligns with the outputs of the respective proposed measures in *German Credit*.

- **Global measures**: Global measures follow Equation 13, 14 and 15.
  - **Total change in the positive regions** The results of the proposed measures align with the results of the baseline measure *Equal Opportunity Difference* as both quantify a larger amount of bias towards *race* than *sex*.
  - **Total change in the negative regions** The results follow a similar trend with the results of the first global measure. The amount of bias per attribute seems to be aligned with the baseline measure '*Equal Opportunity Difference*'.
  - **Total change in the boundary regions** The values of this measure are larger than the values of the first two global measures thus indicating greater uncertainty than certainty in classifying observations. The values of this measure per protected attribute follow the same trend as the first two measures: there is more uncertainty when protected attribute *race* is suppressed.

To summarize, the results of both the local and the global measures align with some of the baseline measures more than with others. In general, a greater alignment is observed between the local measures and most of the base line measures when it comes to both data sets. In addition, as we compare the outputs of the proposed measures between data sets, a common trend is that all values related to the boundary regions are much greater than the values related with the positive or negative regions. These observations are further discussed in the Section 6 section.

6 Discussion

This section is divided into two parts. Firstly, the results of the proposed bias quantification measures are discussed in relation with the baseline measures. Secondly, limitations of this thesis are presented combined with respective suggestions for future research.

6.1 Results and baseline measures

Goal of this study is to propose fuzzy-rough set inspired measures to quantify bias in data sets. More specifically, in this thesis the author investigates, first, whether the outputs of the proposed measures align with the outputs of baseline measures and, second, in which way they differ. To calculate all measures, we compute membership values to three fuzzy-rough regions using two versions of the same data set: one containing all attributes and one containing all attributes but the protected one. Membership values quantify the extent to which an observation belongs to a fuzzy-rough region. The proposed measures attempt to quantify the difference between the membership values of the 'full' and the 'reduced' version of a data set. The four baseline measures employed are widely used in literature and are computed using the AIF360 toolkit(Bellamy et al., 2018). To explore the results of this endeavor, we revert to the research questions.

6.1.1 First set of research questions: Local measures

Local measures attempt to capture the change in membership values between different decision classes when a protected attribute is suppressed.

1. **Would suppressing a certain data set feature cause instances to move from a fuzzy-rough positive region to another?**
   In both data sets, suppressing the protected features causes instances to move

from a fuzzy-rough positive region to another as shown in Table 7 and Table 9. The extent and the direction of this movement are compared between protected attributes. To interpret this measure, first we need to consider that the membership values in a **positive** fuzzy-rough region indicate the extent to which an observation belongs to a decision class **with certainty**. When a protected attribute is suppressed, a value closer to zero means that less observations switch between decision classes and, therefore, the influence of the protected attribute in decision making is low. A larger value means that the protected attribute has a greater influence on the decision making process. In *German Credit*, the values of this first local measure indicate that there is greater movement between decision classes when *age* is suppressed than when *sex* is suppressed. Therefore, this implies that *age* has a greater influence than *age* when classifying loan applicants in terms of credit-worthiness. These results align with the trend in the baseline measures: all four indicate that the data set is more biased against *sex* than *age*.

In *Compas*, one of the baseline measures '*Disparate Impact*' indicates that the data set is equally biased towards *race* and *sex*. '*Equal Opportunity Difference*', '*Average Odds Difference*' and '*Statistical Parity Difference*' report slightly different amounts of bias per protected attribute but their values are close to each other: they all remain within a range of roughly 10%. In other words, they all report a similar amount to bias towards both protected attributes. The results of the proposed local measures also quantify a similar amount of movement between positive regions when each of the protected attributes is suppressed. The greatest alignment is observed between '*Equal Opportunity Difference*' and the local change measures.

2. **Would suppressing a certain data set feature cause instances to move from a fuzzy-rough positive region to a boundary region?**
In both data sets, suppressing the protected features causes instances to move from a fuzzy-rough positive region to a boundary as shown in Table 7 and Table 9. These two measures quantify the amount of *uncertainty* to the decision making process if a protected attribute is suppressed. A value closer to zero indicates that observations there is certainty when classifying observations to a decision class. A larger value indicates that there is increased uncertainty when classifying observations. In *German Credit*, the outputs of this local measure follow the opposite trend as compared to the trend of all the baseline measures. The protected attribute *sex* has larger values in both '*class 0→1*' and '*class 1→0* than the protected attribute *age*. In other words, there is more certainty in classifying observations when *sex* is suppressed.

In *Compas*, as we move from a positive to a boundary region the values when both protected attributes are suppressed are quite similar which generally aligns with the outputs of the baseline measures that report a similar amount of bias towards both protected attributes. A slightly greater movement is reported when attribute *race* is suppressed than when attribute *sex* is suppressed which aligns with '*Equal Opportunity Difference*'. As this second local measure quantifies uncertainty in decision making, it might not be comparable to the baseline measures which measure the amount of bias towards a protected attribute.

Here it is worth noting that in both data sets there is a much larger movement in terms of magnitude as we move from positive to boundary regions (second local measure) compared to the movement between positive regions (first local measure) after suppressing both protected attributes. This means that there is a higher degree of clas-

sification uncertainty associated with the boundary regions compared to classification certainty as expressed in the positive regions. To the author's knowledge, the baseline measures do not express bias related to uncertainty in decision making. More research needs to be made towards that direction.

Finally, when comparing movement from one decision class to another, the local measure outputs are generally greater as we transition from class 0 to class 1 than the opposite for both protected attributes in both datasets. Decision class 0 is the advantaged classification outcome in both data sets (credit-worthy and low recidivism risk) whereas decision class 1 is the disadvantaged outcome. This means that it is more likely for observations to change decision classes as we transition from loan-eligibility to non loan-eligibility than when we move to the opposite direction. To the author's knowledge, measuring bias with regard to the advantaged and disadvantaged classification outcome is not explored by the baseline measures used in this thesis. Future research could explored baseline measures that capture this trend.

6.1.2 Second set of research questions: Global measures

Global measures quantify how much the absence of a protected feature modifies the fuzzy-rough regions. Comparison is made between the same decision class in this second set of measures. The intuition behind these measures is that if a protected attribute is excluded, the membership values should remain the same. The extent to which they might differ is an indication of bias.

1. **How much does the absence of a feature change the fuzzy-rough positive regions?**
   In *German Credit*, the absence of protected feature *age* modifies the positive fuzzy-rough region by 0.1% whereas protected feature *sex* modifies it by 2.05%. Smaller values indicate that the data set is less biased towards the attribute in question. A larger value is indication that the data set is more biased against the protected attribute. To put it differently, the results of this proposed measure in the case of *German Credit* data set contradict the results of the baseline measures which dictate that the data set is more biased towards *age* than towards *sex*.

   In *Compas*, the absence of protected feature *race* modifies the positive fuzzy-rough region by 0.009% whereas protected feature *sex* modifies it by 0.008%. These two percentages are much more similar to each other than in the case of *German Credit*. The baseline measures also measure a similar amount of bias associated with each protected attribute and especially '*Disparate Impact*'. The total change in the positive regions when protected attribute *race* is suppressed is slightly larger in the proposed measures than the change when protected attribute *sex* is suppressed. The opposite happens in two out of four baseline measures: *sex* has a larger bias attached to it than *race*. Nevertheless, there seems to be a general alignment with the baseline measures which also quantify a similar amount of bias towards both attributes.
2. **How much does the absence of a feature change the fuzzy-rough negative regions?**
   The outputs here are similar in terms of trend and magnitude with the results of the first global measure. This alignment with the outputs of the first global measures is expected: the membership values to the fuzzy-rough negative region are computed by subtracting the membership values of the positive region from one.

3. **How much does the absence of a feature change the fuzzy-rough boundary regions?** In *German Credit* the absence of a *age* changes the fuzzy-rough boundary regions by 3.7% whereas the absence of attribute *sex* modifies it by 0.8%. In *Compas* the absence of attribute *race* changes the fuzzy-rough boundary regions by 7.9% whereas the absence of a *sex* modifies it by 5.8%. This measure quantifies uncertainty in classifying an observation to both decision classes. Here, the values of this third measure are higher than the values of the first two measures thus indicating greater uncertainty than certainty in classifying observations in both data sets. The results of this measure do not align with the results of the baseline measures in *German Credit*. The trend is exactly the opposite. In *Compas*, the outputs of this measures also contradict most of the baseline measures with the exception of '*Equal Opportunity Difference*'. The values of this third measure per protected attribute follow the same trend as the first two measures: there is more uncertainty when protected attribute *race* is suppressed.

Comparatively, the magnitude of the values of the first two global measures a lot smaller than the magnitude of the values of the third global measure. This is the case in both protected attributes. This indicates that there is greater uncertainty than certainty when classifying observations.

Finally, it is important to point out that the values of the baseline measures used in this thesis are not absolute. This means they are dependent on the data set pre-processing conducted by AIF360's bias quantification toolkit. Different pre-processing yields slightly different values in the measures whose trend nevertheless remains in the same direction. For example, when it comes to '*Statistical Parity Difference*', AIF360 measures report different values of when tested on a less pre-processed version of the data sets: 0.07% bias against protected attribute *sex* and 0.15% bias against *age* in *German Credit*. These values are different than the values of the baseline measures reported in the Section 5 but, nevertheless, follow the same trend. Therefore, caution is advised computing the baseline measures.

6.2 Contributions, limitations and suggestions for future research

Contributions and limitations of this thesis are listed here along with suggestions for future research. A difference between the proposed measures and the baseline measures is that the former are not able to indicate which group within the protected attribute is being under-represented or discriminated. This means that prior knowledge regarding distinguishing between privileged and under-privileged groups is not required. Therefore, in cases where it is difficult or even unethical to identify the under-privileged group due to the assumptions that are inevitably made the proposed measures might be helpful.

The proposed measures are not tied to a specific fairness definition which means that they are not are not context-dependent. Different fairness definitions satisfy different goals which might be conflicting in different settings such as avoiding racial disparities versus maximizing public safety (Corbett-Davies et al., 2017). Fuzzy-rough set based metrics overcome this hurdle by quantifying the extent to which protected features play a role in classification.

In settings where the direction of the bias needs to be identified in order to be mitigated, the proposed measures do not offer much insight. They quantify the extent to which a protected attribute plays a role in the decision making process but there is no way in specifying which groups within the attribute are treated preferentially. Future

research could focus on developing measures able to quantify the direction of bias using fuzzy-rough sets.

In both data sets, the values of the local measures are generally higher when moving from the advantaged to the disadvantaged decision class. This means that it is easier to classify observations as non credit-worthy or likely to re-offend when protected attributes are suppressed than to to classify observations as credit-worthy or unlikely to re-offend. It seems that protected attributes play a bigger role in classifying observations in the disadvantaged outcome class. The author of this thesis did not identify an existing bias quantification measure that captures this concept. Therefore, in future research emphasis could be placed in identifying existing measures that capture this manifestation of discrimination. Nevertheless, the knowledge that protected attributes negatively influence decision making towards a certain class is beneficiary. Decision-makers can thus proceed with greater caution and avoid inducing additional discrimination.

The proposed measures seem to quantify a form of discrimination which the author was not able to identify in related literature and could serve as contribution of this thesis. This is the uncertainty in classifying observations that is related to the movement in the boundary regions as reported by the second local and third global measures. This quantity always follows the opposite trend in comparison to the baseline measures and its potential application to quantify bias should be further investigated. Since the chosen baseline metrics represent only a small fraction of the existing bias quantification measures, more metrics should be compared with the outputs of the measures in future work.

Another limitation is that the outputs of the proposed measures cannot be compared between data sets. The data sets have a different number of variables which means that they differ in terms of dimensionality. Future research could explore ways that data sets with different dimensions can be compared in terms of the proposed bias quantification measures.

The approach of this thesis does not consider whether combinations of features in a data set might cause bias. Instead this thesis only investigates the influence of individual features in decision making. As combinations of features are likely reveal hidden patterns of bias, future research on bias quantification using fuzzy-rough sets could also move towards this direction.

Finally, a limitation that arises is the absence of treatment of ordinal variables in a data set when it comes to computing the similarity function. In future research, it would be interesting to observe how the membership values of the fuzzy-rough regions would change if ordinal variables are identified and encoded.

7 Conclusion

It is certain that AI Fairness will continue to attract attention as long as AI systems are used in decision making. Related work has proven that, at least in the past decade, researchers have yet to decide on a universally applicable definition of fairness and, by extension, a metric that can effectively quantify unfairness. Both academia and business continue to call for the exploration of more bias quantification metrics. Following this call, this thesis attempts to use measures based on fuzzy-rough set theory to quantify bias testing them on two data sets that are widely reported to cause biased results, *German Credit* and *Compas* (Bellamy et al., 2018). Fuzzy-rough set theory has been successfully used in decision making when information is vague and incomplete and could potentially quantify bias due to its ability quantify the extent to which observations

belong to decision classes. Four metrics developed in AIF360's toolkit are deployed as baseline metrics.

The experiment that is conducted in the framework of this thesis involves quantifying the amount of bias towards *age* and *sex* in the case of *German Credit* data set (Dua & Graff, 2017b) and *race* and *sex* in the case of *Compas* data set (Larson et al., 2017) using two groups of fuzzy-rough set based measures. Data sets are minimally pre-processed in an attempt to allow the measures to be applied in any setting that involves tabular data. Three versions of each data set are created: one version that includes all the original features and two versions where each of the protected features are excluded (or suppressed). For each version of the data set, each observation received a membership value to three fuzzy-rough regions: a positive, negative and a boundary one. The proposed measures make use of these membership values.

The first group of proposed measures, local measures, attempts to investigate whether (1) suppressing a certain data set feature causes instances to move from a fuzzy-rough positive region to another positive region, and (2) from a fuzzy-rough positive region to a boundary region. These two measures account for the instances shifting from one outcome class to another as a protected attribute is excluded. The second group of proposed measures, global measures, attempts to measure (1) how much the absence of a feature changes the fuzzy-rough positive regions, (2) the fuzzy-rough negative regions and (3) the fuzzy-rough boundary regions. Comparison is made between the same decision class and the extent to which the regions differ is an indication of bias.

According to findings, the first local measure mostly aligns with the outcomes of the baseline measures in both data sets in terms of trend. The second second local measure captures the opposite trend with that of the baseline measures. When it comes to magnitude of values, the outputs of the second local measure are much larger than the ones of the first one which means that there is greater uncertainty than certainty in decision-making when protected attributes are oppressed. This serves as an indication of bias. The results of the global measures follow exactly the opposite trend than that of the baseline measures, but there is a similar pattern when it comes to the magnitude of the outcome values. These results are counter intuitive and further research should be made to investigate the cause of this conflict.

Nevertheless, the baseline measure have limitations. They might assign different amounts of bias to a protected attribute depending on the pre-processing of the data. Moreover, they might not report hidden patterns of bias such as inequalities within the same demographics (Chouldechova, 2017). Therefore, the baseline measures deployed for this thesis are not a panacea and their results should be approached with caution. As there are plenty of other existing bias quantification measures that capture different aspects of discrimination, they should be investigated and compared with the outcomes of the proposed measures. Despite their widespread use, the baseline measures deployed in this thesis capture limited aspects of discrimination and this is one of the major limitations of this thesis.

Despite its limitations, this thesis might offer a few contributions to the field of AI Fairness. These are listed below:

- The proposed measures are not based on a mathematical definition of fairness. Current definitions of fairness have been criticised as highly context-depended and even contradictory (Corbett-Davies et al., 2017). Moreover, to satisfy fairness definitions, assumptions might be made regarding which groups might be underprivileged, a process that can be highly subjective. The proposed measures offer a solution as they quantify the extent to which a protected feature plays a role

in decision-making without depending on a fairness definition or making any assumptions. Developers of AI systems that perform decision-making tasks could thus be more cautious when it comes to sensitive features.

- The proposed measures that involve boundary fuzzy-rough regions measure uncertainty in decision making, a quantity that the baseline measures do not capture. Their outputs generally do not align with the baseline measures which might be attributed to the fact that they measure dis-similar quantities. The contribution of such measures in quantifying bias should be further explored in future research.
- Finally, local measures capture the extent to which observations shift from the advantaged to a disadvantaged decision class in binary classification and this shift is again not expressed in the baseline measures. These results could potentially quantify how easy it is for observations to be classified into the disadvantaged decision class when protected attributes are suppressed. Again, more baseline measures need to be compared with the outcomes to verify that the trend remains similar.

Regardless of the contradictory outcomes, this very first attempt of quantifying bias using fuzzy-rough sets gave rise to some interesting results which might encourage further research into this direction.

References

Al-Blooshi, L. & Nobanee, H. (2020). Applications of artificial intelligence in financial management decisions: A mini-review. *SSRN Electronic Journal*.

Arunkumar, C. & Ramakrishnan, S. (2018). Attribute selection using fuzzy rough set based customized similarity measure for lung cancer microarray gene expression data. *Future Computing and Informatics Journal*, *3*(1), 131 – 142.

Barocas, S., Hardt, M., & Narayanan, A. (2019). *Fairness and Machine Learning*. fairmlbook.org. http://www.fairmlbook.org.

Bellamy, R. K. E., Dey, K., Hind, M., Hoffman, S. C., Houde, S., Kannan, K., Lohia, P., Martino, J., Mehta, S., Mojsilovic, A., Nagar, S., Ramamurthy, K. N., Richards, J., Saha, D., Sattigeri, P., Singh, M., Varshney, K. R., & Zhang, Y. (2018). Ai fairness 360: An extensible toolkit for detecting, understanding, and mitigating unwanted algorithmic bias.

Bello, M., Nápoles, G., Morera, R., Vanhoof, K., & Bello, R. (2020). Outliers detection in multi-label datasets. In Martínez-Villaseñor, L., Herrera-Alcántara, O., Ponce, H., & Castro-Espinoza, F. A. (Eds.), *Advances in Soft Computing*, (pp. 65–75)., Cham. Springer International Publishing.

Bello, M., Nápoles, G., Fuentes Herrera, I., Grau, I., Falcon, R., Bello, R., & Vanhoof, K. (2019). *Fuzzy Activation of Rough Cognitive Ensembles Using OWA Operators*, (pp. 317–335).

Bhatt, R. B. & Gopal, M. (2007). Frct: Fuzzy-rough classification trees.

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. Springer.

Chouldechova, A. (2017). Fair prediction with disparate impact: A study of bias in recidivism prediction instruments.

Corbett-Davies, S., Pierson, E., Feller, A., Goel, S., & Huq, A. (2017). Algorithmic decision making and the cost of fairness. *CoRR*, *abs/1701.08230*.

Cornelis, C., De Cock, M., & Radzikowska, A. (2008). *Fuzzy Rough Sets: From Theory into Practice*, (pp. 533 – 552).

Dua, D. & Graff, C. (2017a). UCI machine learning repository.

Dua, D. & Graff, C. (2017b). UCI machine learning repository.

Dwork, C., Hardt, M., Pitassi, T., Reingold, O., & Zemel, R. (2011). Fairness through awareness.

Equality Act 2010 (2016). Equality act 2010. [Online; accessed 28-November-2020].

Feldman, M., Friedler, S., Moeller, J., Scheidegger, C., & Venkatasubramanian, S. (2015). Certifying and removing disparate impact.

Foulds, J. R. & Pan, S. (2018). An intersectional definition of fairness. *CoRR*, *abs/1807.08362*.

Gastwirth, J. & Miao, W. (2009). Formal statistical analysis of the data in disparate impact cases provides sounder inferences than the u. s. government's 'four-fifths' rule: An examination of the statistical evidence in ricci v. destefano. *Law, Probability and Risk*, *8*.

Hardt, M., Price, E., & Srebro, N. (2016). Equality of opportunity in supervised learning.

Hinnefeld, J. H., Cooman, P., Mammo, N., & Deese, R. (2018). Evaluating fairness metrics in the presence of dataset bias. *arXiv:1809.09245v1*.

Holstein, K., Wortman Vaughan, J., Daumé, H., Dudik, M., & Wallach, H. (2019). Improving fairness in machine learning systems. *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*.

Jensen, R. & Cornelis, C. (2011). Fuzzy-rough nearest neighbour classification and prediction. *Theoretical Computer Science*, *412*(42), 5871 – 5884. Rough Sets and Fuzzy Sets in Natural Computing.

Larson, J., Mattu, S., Kirchner, L., & Angwin, J. (2017). compas-analysis.

Liu, Y., Wang, T., Zhang, H., & Cheutet, V. (2019). Simulation-based fuzzy-rough nearest neighbour fault classification and prediction for aircraft maintenance. *Journal of Simulation*, *0*(0), 1–15.

Lysaght, T., Lim, H., Xafis, V., & Ngiam, K. (2019). Ai-assisted decision-making in healthcare: The application of an ethics framework for big data in health and research. *Asian Bioethics Review*, *11*.

Mahoney, T., Varshney, K. R., & Hind, M. (2020). *AI Fairness How to Measure and Reduce Unwanted Bias in Machine Learning*. O'Reilly Media.

McCulloch, J., Wagner, C., & Aickelin, U. (2013). Measuring the directional distance between fuzzy sets. *2013 13th UK Workshop on Computational Intelligence, UKCI 2013*.

Mehrabi, N., Morstatter, F., Saxena, N., Lerman, K., & Galstyan, A. (2019). A survey on bias and fairness in machine learning. *CoRR*, *abs/1908.09635*.

Nápoles, G. (2020). Fuzzy-rough sets for data analysis. lecture notes in knowledge representation. tilburg university.
Nápoles, G., Mosquera, C., Falcon, R., Grau, I., Bello, R., & Vanhoof, K. (2017). Fuzzy-rough cognitive networks. *Neural Networks*, *97*, 19–27.
Pawlak, Z. (1982). Rough sets. *International Journal of Computer and Information Sciences*, *11*, 341–356.
R Core Team (2013). *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing.
Ramentol, E., Verbiest, N., Bello, R., Caballero, Y., Cornelis, C., & Herrera, F. (2012). Smote-frst: A new resampling method using fuzzy rough set theory.
Shrestha, Y. & Yang, Y. (2019). Fairness in algorithmic decision-making: Applications in multi-winner voting, machine learning, and recommender systems. *Algorithms*, *12*, 199.
Speicher, T., Heidari, H., Grgic-Hlaca, N., Gummadi, K. P., Singla, A., Weller, A., & Zafar, M. B. (2018). A unified approach to quantifying algorithmic unfairness: Measuring individual & group unfairness via inequality indices. *CoRR*, *abs/1807.00787*.
Verbiest, N., Cornelis, C., & Herrera, F. (2013). Frps: A fuzzy rough prototype selection method. *Pattern Recognition*, *46*(10), 2770 – 2782.
Verbiest, N., Cornelis, C., & Jensen, R. (2012). *Fuzzy Rough Positive Region-based Nearest Neighbour Classification*.
Verma, S. & Rubin, J. (2018a). Fairness definitions explained. (pp. 1–7).
Verma, S. & Rubin, J. (2018b). Fairness definitions explained. In *Proceedings of the International Workshop on Software Fairness*, FairWare '18, (pp. 1–7)., New York, NY, USA. Association for Computing Machinery.
Vluymans, S., D'eer, L., Saeys, Y., & Cornelis, C. (2015). Applications of fuzzy rough set theory in machine learning: a survey. *Fundamenta Informaticae*, *142*, 53–86.
Wilson, D. R. & Martinez, T. R. (1997). Improved heterogeneous distance functions. *CoRR*, *cs.AI/9701101*.
Yang, J., Xu, T., & Zhao, F. (2018). Modified uncertainty measure of rough fuzzy sets from the perspective of fuzzy distance. *Mathematical Problems in Engineering*, *2018*, 1–11.
Yao, J. (2007). A ten-year review of granular computing. (pp. 734–739).
Zadeh, L. (1965). Fuzzy sets. *Information and Control*, *8*(3), 338 – 353.
Završnik, A. (2020). Criminal justice, artificial intelligence systems, and human rights. *ERA Forum, Journal of the Academy of European Law*, *20*, 567–583.

Appendix A: First item

```
class FRS_Mix:

    def __init__(self, df):

        self.df = df
        self.X = df.values
        self.D = None
        self.N = None
        self.C = None
        self.my_dic = {}
        self.sim_instance_with_instance_num = pd.DataFrame()
        self.sim_instance_with_instance_cat = pd.DataFrame()
        self.lab = []
        self.normalized_distance = 0

    def regions(self):

        self.D = np.unique(self.X[:,-1])

        POS = np.zeros((len(self.D), len(self.X)))
        NEG = np.zeros((len(self.D), len(self.X)))
        BND = np.zeros((len(self.D), len(self.X)))

        ## Dividing the dataframe into Numeric vars & Cat vars
        # Crate a boolean vector for cat & num columns
        for i in self.df:
            if isinstance(self.df[i][1],str):
                self.lab.append(True)
            else:
                self.lab.append(False)

        # Aggregate the boolean values
        content = Counter(self.lab)
        X_num, X_cat = self.X[:,-content[False]:], self.X[:,:content[True]]

        ## Numeric vars preproc.

        # Prototype for each decision class
        self.C = np.zeros((len(self.D), X_num.shape[1]))
        for k in range(len(self.D)):
            self.C[k] = (X_num[X_num[:,-1] == k]).mean(0)

        ## Similarity of every instance with one another
        self.sim_instance_with_instance_cat = squareform(pdist(
            pd.DataFrame(X_cat), self.intersection))
        self.sim_instance_with_instance_num = squareform(pdist(
            pd.DataFrame(X_num), self.similarity_num))
        dimensions = self.sim_instance_with_instance_num.shape[0]
        self.mat_all = 1 - ((self.sim_instance_with_instance_num +
                             self.sim_instance_with_instance_cat)/
                            (self.X.shape[1]-1))
        self.my_dic['instance_with_instance'] = self.mat_all

        # similarity between instances and each class
        for k in range(len(self.D)):
            # num sim with class
            dists_class = np.zeros(shape=(len(self.X)))
            y = self.C[k][:-1]
            for x,idx in zip(X_num,range(len(self.X))):
                dists_class[idx] = np.sum(np.absolute(x[:-1] - y))

            # cat sim with class
            cat_class = np.zeros(shape=(len(self.X)))
```

```
            y = mode(X_cat[self.X[:,-1]==k])[0][0]
            for x,idx in zip(X_cat,range(len(self.X))):
                cat_class[idx] = self.intersection(x,y)

            self.my_dic[k] = 1.0 - ((dists_class + cat_class)/
                                    (self.X.shape[1]-1))

        for k in range(len(self.D)):
            for idx in range(len(self.X)):
                POS[k][idx], NEG[k][idx], BND[k][idx] = self.process_object(idx
    , k)

        return POS, NEG, BND, self.my_dic

    def process_object(self, idx, k):

        inf = 1
        sup = 0

        mem_x_k = self.membership(idx, k)

        for index in range(len(self.X)):

            sim_x_y = self.my_dic['instance_with_instance'][idx][index]
            mem_y_k = self.membership(index, k)

            inf = min(inf, self.implicator(mem_x_k * sim_x_y, mem_y_k))
            sup = max(sup, self.conjunction(mem_y_k * sim_x_y, mem_y_k))

        inf = min(inf, mem_x_k)
        sup = max(sup, mem_x_k)

        return inf, 1-sup, sup-inf

    def similarity_num(self, x, y):
        return np.sum(np.absolute(x[:-1] - y[:-1]))

    def intersection(self, x, y):
        sum_sim = 0
        for i,u in zip(x,y):
            if i != u:
                sum_sim += 1
        return sum_sim

    def implicator(self, a, b):
        return min(1 - a + b, 1)

    def conjunction(self, a, b):
        return max(a + b - 1, 0)

    def membership(self, idx, k):

        sim_x_Ck = self.my_dic[k][idx]
        acc = sim_x_Ck

        for j in range(len(self.D)):
            if j == k:
                pass
            else:
                acc = acc + self.my_dic[j][idx]

        value = (sim_x_Ck / acc)
        return (1 + value)/2 if self.X[idx:idx+1,-1] == k else value/2
```

Listing 1
Algorithm that computes the fuzzy-rough regions using mixed-type data

```
def fuzzy_regions_maker(df,algorithm,prot_attr1,prot_attr2):

    POS, NEG, BND, data_full = algorithm(df).regions()

    # removing the protected attribute sex
    cols = list(df.columns)
    cols.remove(prot_attr1)
    POS_PA1, NEG_PA1, BND_PA1, data_PA1 = algorithm(df[cols]).regions()

    # removing the protected attribute age
    cols = list(df.columns)
    cols.remove(prot_attr2)
    POS_PA2, NEG_PA2, BND_PA2, data_PA2 = algorithm(df[cols]).regions()

    Mix = {}
    Mix['full'] = [POS, NEG, BND]
    Mix[prot_attr1] = [POS_PA1, NEG_PA1, BND_PA1]
    Mix[prot_attr2] = [POS_PA2, NEG_PA2, BND_PA2]

    return Mix
```

Listing 2
Algorithm computing fuzzy-rough regions considering protected variables

```
def overlap(all,attributes):
  dic = {}

  for protected in attributes:
    dic[protected] = {}
    #norm = np.sum()
    for region,name in zip(range(np.array(all['full']).shape[0]),["POS_change",
    "NEG_change","BND_change"]):
      # Positive to Positive
      if region == 0:
        # class0
        nom = np.sum(np.abs(np.minimum(all['full'][region][0],all['full'][
    region][1]) - np.minimum(all['full'][region][0],all[protected][region][1]))
    )
        denom =  np.sum(np.maximum.reduce([all['full'][region][0],all['full'][
    region][1],all[protected][region][1]]))
        dic[protected][str(name+' class0')] = nom / denom
        # class1
        nom = np.sum(np.abs(np.minimum(all['full'][region][1],all['full'][
    region][0]) - np.minimum(all['full'][region][1],all[protected][region][0]))
    )
        denom =  np.sum(np.maximum.reduce([all['full'][region][1],all['full'][
    region][0],all[protected][region][0]]))
        dic[protected][str(name+' class1')] = nom / denom
      elif region == 2:
        # class0
        nom = np.sum(np.abs(np.minimum(all['full'][0][0],all['full'][region
    ][1]) - np.minimum(all['full'][region][0],all[protected][region][1])))
        denom =  np.sum(np.maximum.reduce([all['full'][region][0],all['full'][
    region][1],all[protected][region][1]]))
        dic[protected][str(name+' class0')] = nom / denom
        # class1
        nom = np.sum(np.abs(np.minimum(all['full'][0][1],all['full'][region
    ][0]) - np.minimum(all['full'][region][1],all[protected][region][0])))
        denom =  np.sum(np.maximum.reduce([all['full'][0][1],all['full'][region
    ][0],all[protected][region][0]]))
        dic[protected][str(name+' class1')] = nom / denom
      else:
        pass
```

```
  return dic
```

Listing 3
Algorithm computing fuzzy-rough regions considering protected variables

```
def glob(all,attributes):
  dic = {}
  for protected in attributes:
    dic[protected] = {}
    for region,name in zip(range(np.array(all['full']).shape[0]),["POS_change",
    "NEG_change","BND_change"]):
      # class 0
      nom = np.sum(np.abs(all['full'][region][0] - all[protected][region][0]))
      denom = (max(np.maximum(all['full'][region][0],all[protected][region][0])
    ) - min(np.minimum(all['full'][region][0],all[protected][region][0])))*all[
    'full'][0][0].shape[0]
      class0 = nom / denom
      # class 1
      nom = np.sum(np.abs(all['full'][region][1] - all[protected][region][1]))
      denom = (max(np.maximum(all['full'][region][1],all[protected][region][1])
    ) - min(np.minimum(all['full'][region][1],all[protected][region][1])))*all[
    'full'][0][0].shape[0]
      class1 = nom / denom
      # sum
      dic[protected][str(name+' sum')] = class0 + class1
  return dic
```

Listing 4
Algorithm computing fuzzy-rough regions considering protected variables

```
class PreProc:

    def __init__(self, datfr):
        self.datfr = datfr
        self.cat = pd.DataFrame()

    def make_df(self):
        num, self.cat, colnum, colcat = self.split_cat_num_df(self.datfr)
        cat2 = copy.deepcopy(self.cat)

        # replacing NAs with None
        for i in self.cat:
            self.cat[i].fillna('None', inplace=True)

        # normalizing the numerical vars
        num_norm = self.regul_num(colnum, num)
        # label encoding & normalizing cat vars
        cat_norm = self.scal_reg_cat(self.cat,colcat)

        # First df - all numeric
        df_reg_full= pd.concat([cat_norm,num_norm], axis = 1)
        # Second df - categoricals left intact & normalized numericals
        df_cat_reg_full = pd.concat([cat2,num_norm], axis = 1)

        # Reordering based on the target variable
        one = df_reg_full[df_reg_full[df_reg_full.columns[-1]] == 0.]
        two = df_reg_full[df_reg_full[df_reg_full.columns[-1]] == 1.]
        df_catreg_ord_full = pd.concat([one,two])

        one_cat = df_cat_reg_full[df_cat_reg_full[df_cat_reg_full.columns[-1]]
    == 0.]
        two_cat = df_cat_reg_full[df_cat_reg_full[df_cat_reg_full.columns[-1]]
    == 1.]
        df_catreg_ord_full_mix = pd.concat([one_cat,two_cat])

```

```
        return df_catreg_ord_full, df_catreg_ord_full_mix

    # spliting the dataframe into categorical and numerical vars
    def split_cat_num_df(self, datfr):
        df_num = pd.DataFrame()
        for i in self.datfr:
            if isinstance(self.datfr[i][1],np.int64):
                df_num[i] = self.datfr[i]
        df_colnames_num = list(df_num.columns)

        df_cat = pd.DataFrame()
        for i in self.datfr:
            if isinstance(self.datfr[i][1],str):
                df_cat[i] = self.datfr[i]
        df_colnames_cat = list(df_cat.columns)

        return df_num, df_cat, df_colnames_num, df_colnames_cat

    def regul_num(self, df_colnames_num, df_num):
        min_max_scaler = preprocessing.MinMaxScaler()
        x_scaled = min_max_scaler.fit_transform(df_num.values)
        df_num_norm = pd.DataFrame(x_scaled)
        df_num_norm.columns = df_colnames_num

        return df_num_norm

    def scal_reg_cat(self, df_cat, df_colnames_cat):
        # label coding
        le = preprocessing.LabelEncoder()
        for i in df_cat:
                if isinstance(df_cat[i][1],str):
                    df_cat[i]  = le.fit_transform(df_cat[i])
        # regularizing
        min_max_scaler = preprocessing.MinMaxScaler()
        x_scaled = min_max_scaler.fit_transform(df_cat.values)
        dfcat_reg = pd.DataFrame(x_scaled)
        dfcat_reg.columns = df_colnames_cat

        return dfcat_reg
```

Listing 5
Algorithm that computes the fuzzy-rough regions using mixed-type data

```
get_ipython().run_line_magic('matplotlib', 'inline')
# Load all necessary packages
import sys
sys.path.append("../")
import numpy as np
from tqdm import tqdm
from warnings import warn

from aif360.datasets import BinaryLabelDataset
from aif360.datasets import AdultDataset, GermanDataset, CompasDataset
from aif360.metrics import ClassificationMetric, BinaryLabelDatasetMetric
from aif360.metrics.utils import compute_boolean_conditioning_vector
from aif360.algorithms.preprocessing.optim_preproc_helpers.
    data_preproc_functions        import load_preproc_data_adult,
    load_preproc_data_german, load_preproc_data_compas
from aif360.algorithms.postprocessing.reject_option_classification
    import RejectOptionClassification
from aif360.algorithms.preprocessing.optim_preproc_helpers.opt_tools import
    OptTools
#from common_utils import compute_metrics
from aif360.algorithms.preprocessing.reweighing import Reweighing

from sklearn.linear_model import LogisticRegression
```

```
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import accuracy_score

from IPython.display import Markdown, display
import matplotlib.pyplot as plt
from ipywidgets import interactive, FloatSlider
from aif360.algorithms.preprocessing.optim_preproc_helpers.distortion_functions
                import get_distortion_adult, get_distortion_compas,
    get_distortion_german
from aif360.algorithms.preprocessing.optim_preproc import OptimPreproc

from collections import OrderedDict
from aif360.metrics import ClassificationMetric

def compute_metrics(dataset_true, dataset_pred,
                    unprivileged_groups, privileged_groups,
                    disp = True):
    """ Compute the key metrics """
    classified_metric_pred = ClassificationMetric(dataset_true,
                                                  dataset_pred,
                                                  unprivileged_groups=
    unprivileged_groups,
                                                  privileged_groups=
    privileged_groups)
    metrics = OrderedDict()
    metrics["Balanced accuracy"] = 0.5*(classified_metric_pred.
    true_positive_rate()+
                                            classified_metric_pred.
    true_negative_rate())
    metrics["Statistical parity difference"] = classified_metric_pred.
    statistical_parity_difference()
    metrics["Disparate impact"] = classified_metric_pred.disparate_impact()
    metrics["Average odds difference"] = classified_metric_pred.
    average_odds_difference()
    metrics["Equal opportunity difference"] = classified_metric_pred.
    equal_opportunity_difference()
    metrics["Theil index"] = classified_metric_pred.theil_index()

    if disp:
        for k in metrics:
            print("%s = %.4f" % (k, metrics[k]))

    return metrics

np.random.seed(1)


# #### 1. Demo Reject option classification
# AIF360/examples/demo_reject_option_classification.ipynb

# In[25]:


## import dataset
dataset_used = "compas"#, "german"
protected_attribute_used = 2# 1, 2

if dataset_used == "adult":
#     dataset_orig = AdultDataset()
    if protected_attribute_used == 1:
        privileged_groups = [{'sex': 1}]
        unprivileged_groups = [{'sex': 0}]
        dataset_orig = load_preproc_data_adult(['sex'])
    else:
        privileged_groups = [{'race': 1}]
        unprivileged_groups = [{'race': 0}]
```

```
        dataset_orig = load_preproc_data_adult(['race'])

elif dataset_used == "german":
#     dataset_orig = GermanDataset()
    if protected_attribute_used == 1:
        privileged_groups = [{'sex': 1}]
        unprivileged_groups = [{'sex': 0}]
        dataset_orig = load_preproc_data_german(['sex'])
    else:
        privileged_groups = [{'age': 1}]
        unprivileged_groups = [{'age': 0}]
        dataset_orig = load_preproc_data_german(['age'])

elif dataset_used == "compas":
#     dataset_orig = CompasDataset()
    if protected_attribute_used == 1:
        privileged_groups = [{'sex': 1}]
        unprivileged_groups = [{'sex': 0}]
        dataset_orig = load_preproc_data_compas(['sex'])
    else:
        privileged_groups = [{'race': 1}]
        unprivileged_groups = [{'race': 0}]
        dataset_orig = load_preproc_data_compas(['race'])


# Metric used (should be one of allowed_metrics)
metric_name = "Average odds difference"

# Upper and lower bound on the fairness metric used
metric_ub = 0.05
metric_lb = -0.05

#random seed for calibrated equal odds prediction
np.random.seed(1)

# Verify metric name
allowed_metrics = ["Statistical parity difference",
                   "Average odds difference",
                   "Equal opportunity difference"]
if metric_name not in allowed_metrics:
    raise ValueError("Metric name should be one of allowed metrics")


# In[26]:


# Get the dataset and split into train and test
dataset_orig_train, dataset_orig_vt = dataset_orig.split([0.7], shuffle=True)
dataset_orig_valid, dataset_orig_test = dataset_orig_vt.split([0.5], shuffle=
    True)


# In[33]:


metric_orig_train = BinaryLabelDatasetMetric(dataset_orig_train,
                                             unprivileged_groups=
    unprivileged_groups,
                                             privileged_groups=
    privileged_groups)
display(Markdown("#### Original training dataset"))
print("Difference in mean outcomes between unprivileged and privileged groups =
     %f" % metric_orig_train.mean_difference())

metric_orig = BinaryLabelDatasetMetric(dataset_orig,
```

```
                                            unprivileged_groups=
    unprivileged_groups,
                                            privileged_groups=
    privileged_groups)
display(Markdown("#### Original training dataset"))
print("Difference in mean outcomes between unprivileged and privileged groups =
     %f" % metric_orig.mean_difference())


# In[34]:


print('Disparate impact:',metric_orig.disparate_impact(),
     '\nStatistical parity difference:',metric_orig.
    statistical_parity_difference())


# In[35]:


# Logistic regression classifier and predictions
scale_orig = StandardScaler()
X_train = scale_orig.fit_transform(dataset_orig_train.features)
y_train = dataset_orig_train.labels.ravel()

lmod = LogisticRegression()
lmod.fit(X_train, y_train)
y_train_pred = lmod.predict(X_train)

# positive class index
pos_ind = np.where(lmod.classes_ == dataset_orig_train.favorable_label)[0][0]

dataset_orig_train_pred = dataset_orig_train.copy(deepcopy=True)
dataset_orig_train_pred.labels = y_train_pred


# In[36]:


dataset_orig_valid_pred = dataset_orig_valid.copy(deepcopy=True)
X_valid = scale_orig.transform(dataset_orig_valid_pred.features)
y_valid = dataset_orig_valid_pred.labels
dataset_orig_valid_pred.scores = lmod.predict_proba(X_valid)[:,pos_ind].reshape
    (-1,1)

dataset_orig_test_pred = dataset_orig_test.copy(deepcopy=True)
X_test = scale_orig.transform(dataset_orig_test_pred.features)
y_test = dataset_orig_test_pred.labels
dataset_orig_test_pred.scores = lmod.predict_proba(X_test)[:,pos_ind].reshape
    (-1,1)


# In[19]:


#Best threshold for classification only (no fairness)
num_thresh = 100
ba_arr = np.zeros(num_thresh)
class_thresh_arr = np.linspace(0.01, 0.99, num_thresh)
for idx, class_thresh in enumerate(class_thresh_arr):

    fav_inds = dataset_orig_valid_pred.scores > class_thresh
    dataset_orig_valid_pred.labels[fav_inds] = dataset_orig_valid_pred.
    favorable_label
    dataset_orig_valid_pred.labels[~fav_inds] = dataset_orig_valid_pred.
    unfavorable_label
```

```
196
197     classified_metric_orig_valid = ClassificationMetric(dataset_orig_valid,
198                                                 dataset_orig_valid_pred,
199                                                 unprivileged_groups=
    unprivileged_groups,
200                                                 privileged_groups=
    privileged_groups)
201
202     ba_arr[idx] = 0.5*(classified_metric_orig_valid.true_positive_rate()
                    +classified_metric_orig_valid.true_negative_rate())
203
204 best_ind = np.where(ba_arr == np.max(ba_arr))[0][0]
205 best_class_thresh = class_thresh_arr[best_ind]
206
207 print("Best balanced accuracy (no fairness constraints) = %.4f" % np.max(ba_arr
    ))
208 print("Optimal classification threshold (no fairness constraints) = %.4f" %
    best_class_thresh)
209
210
211 # In[20]:
212
213
214 ROC = RejectOptionClassification(unprivileged_groups=unprivileged_groups,
215                                  privileged_groups=privileged_groups,
216                                  low_class_thresh=0.01, high_class_thresh=0.99,
217                                   num_class_thresh=100, num_ROC_margin=50,
218                                   metric_name=metric_name,
219                                   metric_ub=metric_ub, metric_lb=metric_lb)
220 ROC = ROC.fit(dataset_orig_valid, dataset_orig_valid_pred)
221
222
223 # In[21]:
224
225
226 # preds from validation set
227 # Metrics for the test set
228 fav_inds = dataset_orig_valid_pred.scores > best_class_thresh
229 dataset_orig_valid_pred.labels[fav_inds] = dataset_orig_valid_pred.
    favorable_label
230 dataset_orig_valid_pred.labels[~fav_inds] = dataset_orig_valid_pred.
    unfavorable_label
231
232 display(Markdown("#### Validation set"))
233 display(Markdown("##### Raw predictions - No fairness constraints, only
    maximizing balanced accuracy"))
234
235 metric_valid_bef = compute_metrics(dataset_orig_valid, dataset_orig_valid_pred,
236                 unprivileged_groups, privileged_groups)
237
238
239 # In[22]:
240
241
242 # Transform the validation set
243 dataset_transf_valid_pred = ROC.predict(dataset_orig_valid_pred)
244
245 display(Markdown("#### Validation set"))
246 display(Markdown("##### Transformed predictions - With fairness constraints"))
247 metric_valid_aft = compute_metrics(dataset_orig_valid,
    dataset_transf_valid_pred,
248                 unprivileged_groups, privileged_groups)
249
250
251 # In[23]:
252
```

```
253
254 # Preds from test set
255 # Metrics for the test set
256 fav_inds = dataset_orig_test_pred.scores > best_class_thresh
257 dataset_orig_test_pred.labels[fav_inds] = dataset_orig_test_pred.
        favorable_label
258 dataset_orig_test_pred.labels[~fav_inds] = dataset_orig_test_pred.
        unfavorable_label
259
260 display(Markdown("#### Test set"))
261 display(Markdown("##### Raw predictions - No fairness constraints, only
        maximizing balanced accuracy"))
262
263 metric_test_bef = compute_metrics(dataset_orig_test, dataset_orig_test_pred,
264                 unprivileged_groups, privileged_groups)
265
266
267 # In[24]:
268
269
270 # Metrics for the transformed test set
271 dataset_transf_test_pred = ROC.predict(dataset_orig_test_pred)
272
273 display(Markdown("#### Test set"))
274 display(Markdown("##### Transformed predictions - With fairness constraints"))
275 metric_test_aft = compute_metrics(dataset_orig_test, dataset_transf_test_pred,
276                 unprivileged_groups, privileged_groups)
277
278
279 # #### 2. Reweighing preproc
280 # AIF360/examples/demo_reweighing_preproc.ipynb
281
282 # In[42]:
283
284
285 ## import dataset
286 dataset_used = "compas"#, "german"
287 protected_attribute_used = 2 # 1, 2
288
289
290 if dataset_used == "adult":
291 #     dataset_orig = AdultDataset()
292     if protected_attribute_used == 1:
293         privileged_groups = [{'sex': 1}]
294         unprivileged_groups = [{'sex': 0}]
295         dataset_orig = load_preproc_data_adult(['sex'])
296     else:
297         privileged_groups = [{'race': 1}]
298         unprivileged_groups = [{'race': 0}]
299         dataset_orig = load_preproc_data_adult(['race'])
300
301 elif dataset_used == "german":
302 #     dataset_orig = GermanDataset()
303     if protected_attribute_used == 1:
304         privileged_groups = [{'sex': 1}]
305         unprivileged_groups = [{'sex': 0}]
306         dataset_orig = load_preproc_data_german(['sex'])
307     else:
308         privileged_groups = [{'age': 1}]
309         unprivileged_groups = [{'age': 0}]
310         dataset_orig = load_preproc_data_german(['age'])
311
312 elif dataset_used == "compas":
313 #     dataset_orig = CompasDataset()
314     if protected_attribute_used == 1:
315         privileged_groups = [{'sex': 1}]
```

```
        unprivileged_groups = [{'sex': 0}]
        dataset_orig = load_preproc_data_compas(['sex'])
    else:
        privileged_groups = [{'race': 1}]
        unprivileged_groups = [{'race': 0}]
        dataset_orig = load_preproc_data_compas(['race'])

all_metrics =  ["Statistical parity difference",
                "Average odds difference",
                "Equal opportunity difference"]

#random seed for calibrated equal odds prediction
np.random.seed(1)


# In[43]:


# Get the dataset and split into train and test
dataset_orig_train, dataset_orig_vt = dataset_orig.split([0.7], shuffle=True)
dataset_orig_valid, dataset_orig_test = dataset_orig_vt.split([0.5], shuffle=
    True)


# In[44]:


# Metric for the original dataset
metric_orig_train = BinaryLabelDatasetMetric(dataset_orig_train,
                                             unprivileged_groups=
    unprivileged_groups,
                                             privileged_groups=
    privileged_groups)

RW = Reweighing(unprivileged_groups=unprivileged_groups,
                privileged_groups=privileged_groups)
RW.fit(dataset_orig_train)
dataset_transf_train = RW.transform(dataset_orig_train)

metric_transf_train = BinaryLabelDatasetMetric(dataset_transf_train,
                                           unprivileged_groups=
    unprivileged_groups,
                                           privileged_groups=privileged_groups)


# In[45]:


# Logistic regression classifier and predictions
scale_orig = StandardScaler()
X_train = scale_orig.fit_transform(dataset_orig_train.features)
y_train = dataset_orig_train.labels.ravel()
w_train = dataset_orig_train.instance_weights.ravel()

lmod = LogisticRegression()
lmod.fit(X_train, y_train,
         sample_weight=dataset_orig_train.instance_weights)
y_train_pred = lmod.predict(X_train)

# positive class index
pos_ind = np.where(lmod.classes_ == dataset_orig_train.favorable_label)[0][0]

dataset_orig_train_pred = dataset_orig_train.copy()
dataset_orig_train_pred.labels = y_train_pred


```

```
# In[46]:



dataset_orig_valid_pred = dataset_orig_valid.copy(deepcopy=True)
X_valid = scale_orig.transform(dataset_orig_valid_pred.features)
y_valid = dataset_orig_valid_pred.labels
dataset_orig_valid_pred.scores = lmod.predict_proba(X_valid)[:,pos_ind].reshape
    (-1,1)

dataset_orig_test_pred = dataset_orig_test.copy(deepcopy=True)
X_test = scale_orig.transform(dataset_orig_test_pred.features)
y_test = dataset_orig_test_pred.labels
dataset_orig_test_pred.scores = lmod.predict_proba(X_test)[:,pos_ind].reshape
    (-1,1)


# In[47]:


num_thresh = 100
ba_arr = np.zeros(num_thresh)
class_thresh_arr = np.linspace(0.01, 0.99, num_thresh)
for idx, class_thresh in enumerate(class_thresh_arr):

    fav_inds = dataset_orig_valid_pred.scores > class_thresh
    dataset_orig_valid_pred.labels[fav_inds] = dataset_orig_valid_pred.
    favorable_label
    dataset_orig_valid_pred.labels[~fav_inds] = dataset_orig_valid_pred.
    unfavorable_label

    classified_metric_orig_valid = ClassificationMetric(dataset_orig_valid,
                                                dataset_orig_valid_pred,
                                                unprivileged_groups=
    unprivileged_groups,
                                                privileged_groups=
    privileged_groups)

    ba_arr[idx] = 0.5*(classified_metric_orig_valid.true_positive_rate()
                    +classified_metric_orig_valid.true_negative_rate())

best_ind = np.where(ba_arr == np.max(ba_arr))[0][0]
best_class_thresh = class_thresh_arr[best_ind]

print("Best balanced accuracy (no reweighing) = %.4f" % np.max(ba_arr))
print("Optimal classification threshold (no reweighing) = %.4f" %
    best_class_thresh)


# In[48]:


display(Markdown("#### Predictions from original testing data"))
bal_acc_arr_orig = []
disp_imp_arr_orig = []
avg_odds_diff_arr_orig = []

print("Classification threshold used = %.4f" % best_class_thresh)
for thresh in tqdm(class_thresh_arr):

    if thresh == best_class_thresh:
        disp = True
    else:
        disp = False

    fav_inds = dataset_orig_test_pred.scores > thresh
```

```
436     dataset_orig_test_pred.labels[fav_inds] = dataset_orig_test_pred.
        favorable_label
437     dataset_orig_test_pred.labels[~fav_inds] = dataset_orig_test_pred.
        unfavorable_label
438 
439     metric_test_bef = compute_metrics(dataset_orig_test, dataset_orig_test_pred
        ,
440                                       unprivileged_groups, privileged_groups,
441                                       disp = disp)
442 
443     bal_acc_arr_orig.append(metric_test_bef["Balanced accuracy"])
444     avg_odds_diff_arr_orig.append(metric_test_bef["Average odds difference"])
445     disp_imp_arr_orig.append(metric_test_bef["Disparate impact"])
```

Listing 6
Computing the baseline measures using the example scripts at AIF360 Github account