Mathematical Programming Based Algorithms for Fairness in Interpretable Learning

A.C. Lumadjeng

A thesis submitted in partial fulfillment for the degree
MASTER IN SCIENCE

in

ECONOMETRICS AND MANAGEMENT SCIENCE
Erasmus University Rotterdam

Supervisor: Dr. M. H. Akyüz
Second assessor: Dr. T.A.B. Dollevoet

March 8, 2022

Contents

Abstract

The use of machine learning methods in decision-making continues to grow in different fields such as health care, education, and law. As the consequences of these decisions are important for society, there is a need for accurate, interpretable, and fair machine learning. Due to the complexity of machine learning methods today, these models can be considered to be black boxes which makes it difficult to explain why a certain decision is made by these models. Hence, machine learning models are not always interpretable. Furthermore, in some data sets, there is discrimination towards a certain group of people present, which can be perpetuated by machine learning models that learn from these data sets. This thesis focuses on incorporating fairness in interpretable learning, and more specifically, in multi-class classification. In tree-based classification algorithms, a predicted class is chosen if it satisfies a set of decision rules. In this work, two algorithms are proposed to construct decision rule sets that result in fair and interpretable classification. The first algorithm, Fair Rule Extraction (FairRUX) extracts fair decision rules from existing classification algorithms by formulating the classification problem as a linear program subject to fairness constraints. These fairness constraints are based on the notions of equal opportunity and equal overall mistreatment. The resulting set of rules can be considered fair and has thus decreased in size, such that we also have an interpretable rule set. The second algorithm, Fair Rule Generation (FairRUG) constructs the rule set itself by following a column generation (CG) procedure. We have tested our fair methods on six data sets and the results show that fair and interpretable classification, with an acceptable test accuracy, is attainable in the multi-class setting.

Acknowledgements

In the past year, I've had the pleasure to work on fairness in artificial intelligence under the daily supervision of Dr. (Hakan) Akyüz. First and foremost, I would like to thank him for his valuable guidance, feedback, and especially his encouragement. Hakan has always made me want to achieve the best I can, and I hope to be able to work with him in the future. Also, thank you to Dr. (Twan) Dollevoet for being the second assessor to this project. His remarks on my thesis were very valuable.

Naturally, I would like to thank my dear friend and fellow master's student, Daphne Stok, who was always able to provide a kind listening ear, even in challenging times. Finally, thank you to my parents for their support, and my boyfriend, who was always ready to hear about the next developments.

1 Introduction

In today's world, machine learning plays a big role in our lives and its use continues to grow. In medicine, education, and law, many decisions are already made by machine learning classification models. The decisions of these models can have a great influence on individuals' personal and societal lives. Consequently, the growing use of these models has a serious impact on society. Therefore, it is important to ensure that the decisions of these models are trustworthy.

The initial focus of machine learning is to train models to make correct decisions without models being specifically programmed to make these decisions. Traditionally, *correct* decisions mean that models are able to accurately predict the target outcomes in historical data. However, the number and complexity of machine learning technologies have since increased, expanding their capabilities (Gerlings et al., 2021). This results in advanced models that can be considered to be *"black boxes"*, wherein it is not possible to exactly follow the data processing cycle for people to interpret and understand the resulting outcome. Naturally, this is crucial to trust the decision-making process. A response to this black box problem is the introduction of interpretable learning, which has emerged to discover knowledge, to debug, or to justify the model and its predictions (Molnar et al., 2020). Gaining insight into the development process helps us to improve the model and it also allows for developers to explain why certain decisions are made by these models to clients, boards, or other stakeholders. It follows that, in addition to accurate models, we also wish for machine learning models to be *interpretable* to trust their decisions.

Thus, the focus of correct decision-making involves accuracy and interpretability. Next to that, we know that machine learning is based on historical evidence. With this, even if we were to consider a model for classification to be accurate and interpretable, it does not necessarily ensure that its decisions are *fair*. This is because historical data reflects demographic disparities through historical prejudices against certain social groups, perpetuating stereotypes or social inequalities. And so, learning from training data that contains bias will reproduce similar dynamics. An example of a mined pattern we would like to avoid learning is the disparity in hiring decisions, where female candidates with similar attributes to male candidates, are less likely to be hired (Barocas et al., 2019). However, because these patterns are the consequence of historical societal standards and moral judgments, learning algorithms have no means of distinguishing between them. This may result in unfair classification outcomes. Hence, the focus of this research is achieving fairness in interpretable learning, and more specifically, we focus on binary and multi-class classification.

1.1 Goal of this thesis

The aim of this thesis is to develop a fair and interpretable classification tool, which works in the binary and multi-class settings. Many common classification tasks can easily be expressed as mathematical optimization problems. Following the introduction of the context of the problem, the key research question is articulated as follows:

(Q) How can we make use of mathematical programming for fairness in interpretable learning?

To keep the research organized to answer this question, we divide the main question into three sub-questions. The first sub-question focuses on the implementation of the well-known column generation (CG) algorithm to achieve fairness in interpretable learning. The answers to second and third sub-questions show us the effects

of implementing fairness and whether an acceptable test accuracy of predictions is achieved. This leads to the following sub-questions.

(SQ1) How can we use the column generation algorithm for fair learning?
(SQ2) What will be the effect of implementing fairness into interpretable learning?
(SQ3) Is it possible to maintain an acceptable test accuracy if we were to implement fairness?

The starting point for achieving fairness in interpretable learning is the interpretable classification framework of Akyüz and Birbil (2021). This model is an interpretable classifier, expressed as a linear program (LP), that extracts and generates interpretable decisions, in which the classification error is minimized to maintain a good accuracy. The main focus of this thesis is making this interpretable classifier *fair* by employing optimization at training time, by adjusting the LP. We propose to add fairness constraints to the LP that are based on different notions of fairness. The resulting LP problem is solved for fair classification, and to extract interpretable information via the well-known column generation (CG) procedure.

1.2 Contribution

To the best of our knowledge, previous studies that have investigated fair and interpretable models lack either fairness or interpretability in the classifiers. Other studies that were able to achieve fairness in interpretable learning, are often limited to binary classification only. A combination of fair and interpretable classifiers in the multi-class setting remains to be found. Hence, the contribution of this thesis is the presentation of a mathematical program for *fair and interpretable multi-class classification*, which can be used for more trustworthy decision making, in terms of accuracy, interpretability, and fairness. More importantly, this research is also of relevance for the groups of people, concerned with these decisions, that are treated unfairly with the current machine learning methods.

The remainder of this thesis is organized as follows. Section 2 analyzes previous studies regarding interpretable learning approaches, approaches to fair learning, and fair and interpretable models for binary classification. Section 3 presents a formal description of the problem in this thesis and what we regard to be *fair*. Section 4 introduces the mathematical formulation of the problem, and the methodology to fair and interpretable learning is presented in section 5. Computational experiments test our methods in section 6 and finally, we come to the conclusion in section 7. Limitations or opportunities for further research are elaborated in the discussion in section 8.

2 Literature Review

Section 2.1 discusses the relevant literature on interpretability and section 2.2 discusses the approaches to fairness and its connection to interpretability.

2.1 Interpretability

The literature describes several different approaches to interpretable learning. Some works focus on specific models that can be considered as interpretable or adjusted to being interpretable. Bien and Tibshirani (2009)

proposes classification by a prototype vector machine model, and claims that dividing the test set into subsets called prototypes will result in interpretable classification. Regression models are also considered to be interpretable, under certain procedures. Schielzeth (2010) argues that linear regression models often yield uninterpretable results, but with some pre-processing, namely by centering and standardizing the input variables, we get interpretable regression models. Another approach to interpretable regression models is the work of Tibshirani (1996). As ordinary least squares can become quite uninterpretable due to many input variables, Tibshirani (1996) proposes the *lasso* technique to shrink the value of certain variables to zero, while maintaining appropriate operator selection.

The approach of this thesis is based on mining interpretable rule sets from ensembles of decision trees. Rules in classification are the decision rules that result in the classification of a sample. The works concerning interpretable rule learning all agree that an interpretable classifier is a model with the least amount of rules while maintaining a good accuracy. The starting point of this work, Akyüz and Birbil (2021), manages to construct an accurate interpretable learner by formulating a mathematical program and optimizing the hinge loss, with respect to a candidate rule set. This problem is solved via rule extraction, and rule generation with column generation, to generate interpretable decision rules. Rule extraction methods have proven to be popular and reliable ways to obtain an interpretable classification. Another work that shares the idea of extracting decision rules from an ensemble of decision trees, is the work of Birbil et al. (2020). A minimum rule covering problem is formulated as a mathematical program with the objective to minimize the total impurity, where the resulting set of rules is also interpretable. Heuristics can also perform rule extraction to obtain the desired set of rules. Adnan and Islam (2017) proposed a heuristic that extracts rules from forests that result in a higher accuracy and training set coverage compared to using the whole set. Another approach of Wanga et al. (2020) performs rule extraction by encoding rules from ensemble methods to chromosomes and carries out a multi-objective optimization process. They optimize for accuracy and interpretability after which they decode the chromosomes again. Liu et al. (2012) proposes interpretable binary classification by combining rule extraction with feature extraction in an iterative manner. A sparse encoding method is introduced to extract representative rules using the binary encoding of the forest. Then, a subset of features is selected, based on changes in the distribution of features that occurred by only considering the subset of rules for classification. This, in turn, results in a smaller set of rules again.

Most works achieve an acceptable accuracy by either maximizing the accuracy or by minimizing the classification error. Illustrations of such approaches are developed by Bringmann and Zimmermann (2009), and Rivest (1987), or Akyüz and Birbil (2021) via optimization of the hinge loss. Lakkaraju et al. (2016) starts similar to Akyüz and Birbil (2021), with formalizing rule learning through an objective function that simultaneously optimizes accuracy and interpretability of the rules. However, the problem is solved with a smooth local search algorithm. The approaches of both works apply to binary classification and multi-class classification. Lawless et al. (2021) proposed a similar approach to Akyüz and Birbil (2021), but with the objective function defined as the Hamming loss, and only applicable to binary classification. However, both Akyüz and Birbil (2021) and Lawless et al. (2021) rely on the mathematical programming-based column generation and therefore are great for comparison in binary classification.

In deep learning models, visualization techniques such as weight histograms and saliency maps are also often proposed to gain insight into the data. Shickel and Rashidi (2020) uses weight histograms to inspect and determine the overall distribution learned weights of a deep learning model. Shickel and Rashidi (2020) also uses saliency maps

for interpretability of image processing techniques. Saliency maps, first introduced by Simonyan et al. (2014), visualizes the gradients of outcome labels with respect to input images. This indicates how the classification changes, with respect to small changes in each input image pixel.

2.2 Fairness

To ensure a notion of fairness, we need to establish what we consider to be *fair*. The definition of fairness in the literature is diverse. Not all studies mentioned below employ mathematical programming for rule learning but remain relevant studies on fair learning. Madhavan and Wadhwa (2020) measures fairness by avoiding disparate impact, this means that there is statistical independence between the outcome and the sensitive attributes. Feldman et al. (2015) applies the same metric as Madhavan and Wadhwa (2020) but allows a small percentage of disparate impact. Both works are applicable to fairness for binary classification, but rather than focusing on the classification process itself (via e.g. a mathematical program), Madhavan and Wadhwa (2020) and Feldman et al. (2015) focuses on the data, by applying pre-processing and focus on the input data by making it unbiased. This approach would be considered as a pre-processing approach by Barocas et al. (2019). Another relevant work with a similar goal is the work of Ravichandran et al. (2020). This work aims to develop a fair variant of XGBoost by Chen and Guestrin (2016). The metric used is disparate impact and fairness is ensured via a fairness regularizer to remove correlation between the sensitive attribute and the target value. This can also be considered as a pre-processing approach to improve fairness. With the same metric, but a different approach, Zafar et al. (2019) introduces a constraint-based framework to measure decision boundary unfairness for binary classification. Avoiding disparate impact may not work for all problems and so Hardt et al. (2016) uses the metric of equalized odds. This means that the false positive rate and false negative rate over all different groups of people must be equal. Then, fairness is ensured via a post-processing procedure, by adjusting the learned predictor to remove any unfairness. The equalized odds notion of fairness is a popular fairness definition of choice. Tavakol (2020) optimizes an equalized odds ratio and models fairness-aware binary classification in a counterfactual setting. Counterfactual learning copes with biases in the decision process for resampling purposes. From the literature, we also see that the use of a mathematical programming to fairness is an often desired approach. Agarwal et al. (2018) reduces unfairness in cost-sensitive classification, where the costs are defined as the classification error. In two different mathematical programs, these costs are minimized subject to fairness constraints, based on demographic parity and equalized odds. However, most of these works remain to apply to binary classification only.

Most fairness definitions from Barocas et al. (2019) are defined for binary classification and the generalization of these definitions to multi-class classifications are non-trivial. Tavakol (2020) proposes a fair classification method for binary classification and mentions it is extendable to the multi-class setting. However, actual generalizations, experiments, and results remain to be found. Denis et al. (2021) achieves fairness in multi-class classification by constructing fair plug-in estimators to obtain the distribution of fair data sets. The proposed fair plug-in estimators satisfy demographic parity.

Statistical classification is a popular problem to add fairness into, but fairness guarantees have also been researched for other types of problems. Roman et al. (2020) achieve equalized odds in regression problems by resampling the sensitive attributes on which equalized odds seem to apply. The distribution of the sampled data

set is then compared to the distribution of the whole data set, and the discriminatory and predictive parameters are adjusted accordingly. Roman et al. (2020) also performs their fair classification in multi-class logistic regression problems. However, they state that they cannot formally guarantee fairness for these models. Du et al. (2021) research fairness in deep learning and propose several steps to achieve fairness in neural networks. From pre-processing to avoid discrimination via input to using a fairness regularizer that processes the attributes. Finally, post-processing is also applied via calibration.

There is ample research available for interpretable learning and fairness in binary classification, but there is much less research to be found for fairness in multi-class classification. Moreover, the combination of fair and interpretable learning in multi-class classification remains to be scarce as well. Closer to the approach of this thesis and to the work of Akyüz and Birbil (2021), is the work of Lawless et al. (2021). Lawless et al. (2021) also applies a constraint-based framework for binary classification. Lawless et al. (2021) uses the metric of equalized odds, and its relaxation equal of opportunity, to define fairness constraints for the linear program for rule learning. The solution of this linear program is a set of interpretable decision rules that depicts the classification. Hence, Lawless et al. (2021) achieve fairness in interpretable learning, however, it remains limited to binary classification.

3 Problem Description

In this chapter, we first define the base of our classification tool in section 3.1. Section 3.2 presents the proposed classification framework for interpretable learning. Next, we describe notions of fairness to add to the interpretable learner in section 3.3.

3.1 Decision Tree Ensemble Methods and Mathematical Optimization

A decision tree in machine learning is a predictive modeling approach that consists of input observations that run through branches (conditions) to end up in a leaf, resulting in their predicted outcome (class). Thus, a decision tree consists of a set of leaves, each corresponding to a different *rule*. A rule is an independent if-then statement, which contains one or more conditions that assign a class to a set of samples, resulting in a classification. Example 3.1 illustrates a rule in multi-class classification where the task is to predict the type of flower. When a sample satisfies such a rule, it receives the classification label corresponding to that leaf. If a sample is covered by more than one rule, majority voting among the assigned labels is used to determine the class of the sample. When the classes of a decision tree are discrete, we speak of a classification tree.

Example 3.1. if $(petal\ length \geq 2.75)$ **and** $(petal\ width \leq 1.5)$ **then** *the flower is an* **iris versicolor**

Mathematical optimization plays an important role in many fields, such as criminal justice (Zeng et al., 2017), cybersecurity (Torres et al., 2019) and health care (Bertsimas et al., 2016). In a decision tree, the model is defined by its branch nodes, leaf nodes, and ultimately its prediction structure (Carrizosa et al., 2021). We know that pruning a tree coincides with minimizing the number of rules thus a classification problem is easily formulated as a

mathematical program. Many works have solved classification problems via mathematical optimization, such as by formulating the classification problem as a nonlinear continuous optimization problem (Blanquero et al., 2016), or as a mixed-integer linear problem (Bertsimas and Dunn, 2017). To ensure accurate predictions, these formulations all minimize the costs that are defined as the equivalence of the classification error. Then, constraints are added to the mathematical program to control the path that each individual takes. Thus these previous works inspire a formulation of the classification trees as a linear program (LP).

An enhancement to decision tree models is to build an ensemble of trees. That is, to combine the outputs given by a collection of trees, as opposed to a single one by, for instance, bagging or boosting trees (González et al., 2020). Common types of tree ensemble methods are Random Forest (Biau and Scornet, 2016), Adaptive Boosting (Freund and Schapire, 1997) and Gradient Boosting (Friedman, 2001). Ensemble methods combine multiple decision trees and reduce the chance of overfitting. However, an important drawback to combining more than one tree in a classification model, is that the classification decision becomes more difficult to understand. This causes the interpretability of the model's decision process to decrease. Our solution to this problem and fairness problem is to mine *interpretable* rule sets from ensembles of *fair* decision trees.

3.2 Classification Framework for Interpretable Learning

The fair classification tool of this work builds upon the classification framework of Akyüz and Birbil (2021) for interpretable learning. Akyüz and Birbil (2021) proposes two methods for interpretable classification that utilizes mathematical optimization.

3.2.1 Rule Extraction (RUX)

The first method for interpretable classification of Akyüz and Birbil (2021) consists of the Rule Extraction (RUX) algorithm. A LP formulation is presented in section 4 that enables the RUX algorithm to select rules from existing tree or rule ensembles for interpretation. The ensemble methods considered are Random Forest classification and Adaptive Boosting. The resulting set of rules is then an interpretable rule set. For example, from a data set with 4069 samples, we wish to classify each sample as either 0 or 1. Using a normal Random Forest Classifier to solve this, with a maximum tree depth of three and no limitations on the number of leaf nodes, we would end up with 798 leaf nodes, or 798 distinct rules. Understanding the predicted outcome of one sample would mean that one has to go through all these 798 rules. Consequently, the predicted outcome is not immediately interpretable. With the RUX algorithm, we are able to reduce the number of rules for this classification to 50 rules. Going through 50 rules seems to be a more achievable task.

The objective function of the LP minimizes the *hinge loss*, a loss function equivalent to the classification error, and the number of used rules simultaneously. Minimizing the hinge loss, i.e. classification error, results in predicted values close to the target values, and thus in a sufficiently accurate classifier. The minimization of the number of used rules creates the interpretable classification. Together we get an LP formulation that is able to mine an interpretable rule set while still maintaining good accuracy. A formal definition of the hinge loss and this LP is given in section 4.

3.2.2 Rule Generation (RUG)

In contrast to the RUX algorithm, the Rule Generation (RUG) algorithm generates rules instead of extracting them from an existing classifier. The RUX algorithm takes an existing ensemble method (e.g. Random Forest, AdaBoost) to make it interpretable, whilst the RUG algorithm is itself a classifier as it generates new rules used for classification. RUG uses the same LP as the RUX algorithm, where the hinge loss and number of used rules are minimized. Naturally, with the same objective function as in the RUX algorithm, the resulting rulesets of the RUG algorithm are also interpretable. The rules of the decision trees correspond to the columns of the linear programming problem. This enables us to generate rules by generating columns. Akyüz and Birbil (2021) propose to do this via the well-known Column Generation approach in optimization (Desaulniers et al., 2005). A formal description of the CG approach is given in section 4.

3.3 Fairness Definitions

In the classification framework for interpretable learning of the previous section, we create our tool for fair classification. In order to ensure fairness, we decide on how to measure fairness. This subsection discusses several notions of fairness to set on a proper definition of fairness such that we can choose a fairness metric. However, we first start by introducing some assumptions and notation. Finally, this allows us to *measure fairness* and adjust the given LP for fairness in interpretable learning. Knowledge on how to define and measure fairness also allows us to assess the quality in terms of fairness of our classifier.

Sensitive Attribute

A data set consists of several samples, each characterized by their attributes and class label, i.e. the target value during prediction. In our case, one sample always represents an individual. To distinguish between two individuals that belong to different groups among those we want to ensure fairness, we choose one of the attributes to be the *sensitive attribute*, that is, the distinct social characteristic of a person. For example, Larson et al. (2016) show that in a classification system used in the United States, where we would like to predict whether a convicted criminal is likely to re-offend, black defendants were often predicted to be at a higher risk of recidivism than they actually were, and white defendants were often predicted to be less risky than they were. In this case, we can consider `race` to be the sensitive attribute with possible values of `black` and `white`. We refer to the values of the sensitive attributes as *groups*.

Accuracy

As we wish to add fairness into an interpretable classifier, we also wish for its accuracy to remain acceptable. There are other measurements to evaluate the performance of a classifier, such as precision and recall, but perhaps the most well-known and most used measurement is *accuracy*. Accuracy is used to describe how close the predicted value is to the true value. We formally define accuracy using the same notation as Barocas et al. (2019). Suppose we have a sample with features $\mathbf{x} \in \mathbb{R}^P$ and class label y. We refer to its predicted label as $\hat{y}$. The accuracy of

the classifier can then be defined as the probability of correctly predicting the target variable:

$$P(y = \hat{y}) \tag{1}$$

The question of what *fairness* actually is, remains to be vague as it can be defined in different ways. In the following sections, we define fairness with the help of the probability definition for accuracy. Introduced by different works and often considered as fairness definitions in other literature, we review the following non-discrimination criteria as definitions of fairness: *Individual Fairness, Fairness through Unawareness, Independence, Separation* and *Sufficiency*.

3.3.1 Individual Fairness

Individual fairness is considered to be the most intuitive notion of fairness and was first introduced by Dwork et al. (2012). Individual fairness is based on the condition that similar individuals should be treated similarly. Hence, given two individuals $i, j \in \mathcal{I}$, where $\mathcal{I}$ is the set of samples, with features $\mathbf{x}_i, \mathbf{x}_j \in \mathbb{R}^P$. We call a *similarity distance* a real-valued function that quantifies how similar two objects are. Let $\mathcal{D}$ and f be such similarity distances, where f describes how similar two individuals i and j are, and $\mathcal{D}$ describes how similar they are treated after prediction. We say that two individuals $i, j \in \mathcal{I}$, satisfy individual fairness if

$$\mathcal{D}(\hat{y}_i, \hat{y}_j) \leq f(\mathbf{x}_i, \mathbf{x}_j) \tag{2}$$

In other words, if two individuals that are similar to each other based on their features (i.e. characteristics), then their treatment should also be similar. In contrast to individual fairness, the following criteria are based on *group fairness*.

3.3.2 Fairness through Unawareness

Another straightforward solution to achieve an impartial classifier would be to simply ignore the sensitive attribute, hence achieving fairness through unawareness. For example, in a job hiring system where we consider `gender` as the sensitive attribute, fairness through unawareness would discard this attribute during the classification process. In the literature, this is also known as satisfying *disparate treatment*. However, as Hardt et al. (2016) explains, this definition will not be great at ensuring fairness as the sensitive attribute is often correlated with other attributes. For example, a race may be linked to a neighborhood within a city, and removing the sensitive attribute `race` but keeping the attributes `neighborhood` or `city`, will not prove to be effective. Hence, fairness through unawareness will result in other attributes serving as a proxy for the sensitive attribute, and so unfairness is still present.

3.3.3 Independence

Fairness based on *independence* is introduced as a formal fairness definition by Barocas et al. (2019) and is one of the most well-known criteria for fairness. Independence requires that a classifier is independent of the sensitive attribute and so all different groups of people should receive the same treatment. In the multi-class setting, we define independence as satisfying the following conditions:

$$P(\hat{y} = k | G = g) = P(\hat{y} = k | G = g') \qquad \forall k \in \mathcal{K} \text{ and } \forall g, g' \in \mathcal{G} \tag{3}$$

where $\mathcal{K}$ is the set of class labels and $\mathcal{G}$ is the set of groups with distinct characteristics. For example, if the sensitive attribute were `religion`, then we could have $g =$`christian` and $g' =$ `islamic`. Independence is also known under different names such as *demographic parity*, *group fairness* or *disparate impact*.

3.3.4 Separation

Introduced by Hardt et al. (2016) for binary classification, *separation* considers the possibility that the sensitive attribute variable is correlated with the target variable. It allows a correlation to the extent that it is justified by the target variable, by requiring equality in error. In binary classification, we define separation as satisfying the following conditions:

$$P(\hat{y} = 0|y = 1, G = g) = P(\hat{y} = 0|y = 1, G = g') \qquad \forall g, g' \in \mathcal{G} \tag{4}$$

$$P(\hat{y} = 1|y = 0, G = g) = P(\hat{y} = 1|y = 0, G = g') \qquad \forall g, g' \in \mathcal{G} \tag{5}$$

where $\hat{y} = 1$ means the predicted class is a beneficial acceptance and $\hat{y} = 0$ is considered a loss. Therefore, separation requires that all groups experience the same false negative rate and false positive rate. Separation is also referred to as the fairness metric *equalized odds* or *avoiding disparate mistreatment*. Naturally, there are relaxations and extensions of this criterion, such as *equal opportunity* and *equalized correlation*. Equal opportunity is an often used relaxation and only imposes an equal false negative rate. Equalized odds is first introduced by Hardt et al. (2016) for binary classification, but is then generalized by Woodworth et al. (2017) for multi-class classification problems under the name *Equalized Correlation*. However, Woodworth et al. (2017) states that learning a predictor under equalized correlations is computationally hard and continues with developing their methods with equal opportunity for binary classification only.

3.3.5 Sufficiency

A classifier satisfies *sufficiency* when the prediction subsumes all information about the sensitive attribute that is relevant to the label (Lee et al., 2021). As a result, we look at the features that were predictive of the target value, hence we have sufficient features. Barocas et al. (2019) formally defines that a classifier is sufficient for the sensitive attribute if and only if the following conditions hold:

$$P(y = 1|\hat{y} = k, G = g) = P(y = 1|\hat{y} = k, G = g') \qquad \forall k \in \mathcal{K} \text{ and } \forall g, g' \in \mathcal{G} \tag{6}$$

In contrast to independence and separation, sufficiency is measured after classification. This idea is closely related to *calibration* of a model. Chouldechova (2016) formalizes the connection with an equivalent metric called *calibration within groups*.

3.3.6 Fairness Definition of Choice

We see that there are different choices in defining fairness. To each definition of fairness, there are advantages and disadvantages. Based on these, we choose the appropriate definition for our classification tool.

First, we have seen the notion of individual fairness. In individual fairness, the object to be classified is the individual and requires for similar individuals to be treated similarly. This seems to be a good measure as it

captures the most intuitive definition of fairness. A drawback is that we need two distance functions to depict the similarity between individuals and between outcomes, however choosing what is an appropriate distance function is not trivial (Hardt et al., 2016).

Next, there is group fairness. Using a criterion that considers group fairness is the most researched approach in fair machine learning and is formally introduced by Barocas et al. (2019), with many others following with equivalent and relaxed notions of fairness. Group fairness aims to define fairness in terms of statistical parity conditions between different social groups and is imposed on the decisions of an algorithm. However, Dwork et al. (2012) states that in some cases, produced decisions might appear to be blatantly unfair to individuals involved. Furthermore, Barocas et al. (2019) and Kleinberg et al. (2017) prove that it is not possible to satisfy multiple group fairness criteria at the same time. Thus, in the case of group fairness, we can only pick one fairness criterion at a time to use in quantifying fairness.

We see that fairness through unawareness will not necessarily ensure fairness due to the classifier not actually being unaware of the sensitive attribute. Though it is simple to compute as it only requires removing the sensitive attribute from the training data, there may be many highly correlated features in a sample. In the areas that we consider (e.g. medicine, education, law), we expect a lot of features of a sample to be correlated.

As we can see from the definitions of disparate impact and disparate treatment, we do not need to know any information on the actual target variable value before prediction. This might be an advantage if we have an unlabeled data set. However, independence requires that a classifier and the sensitive attribute are uncorrelated and for the same reason as with fairness through unawareness, the sensitive attribute and the target variable may sometimes be correlated. It follows that perfect classification would not be possible. Hence, independence undermines the utility that we hope to achieve. More importantly, independence does not necessarily ensure fairness. It may result in an equal amount of positive classifications between two groups, but it ignores the possibility that a fair classification is unbalanced between groups. This is because we ignore the target value and only the overall percentage of acceptance between groups has to match. For example, suppose a company hires diligent people, that coincidentally all belong to group a, and another group b is hired that contains more careless people. When independence is required, the hiring rate between these two groups is equal, possibly ignoring truly diligent individuals from group a, to make room for individuals from group b (Barocas et al., 2019).

When using separation as a fairness criterion, we allow the predicted value to depend on the sensitive attribute, but only through the target variable. This means that we need information on the sensitive attribute before prediction, but this is no problem as in our case, we consider labeled data sets. An advantage of separation is that it is to obtain a perfect prediction, that is $\hat{y} = y$, in contrast to independence. More specifically, equalized odds seem like an attractive metric to use as it penalizes laziness, so there is an incentive to reduce errors uniformly in all groups. A drawback of separation, and more specifically, equalized odds, is that it requires an equal false negative rate and false positive rate across groups. Translating the notions of false negative and false positive rates to a definition of equalized odds in the multi-class setting has proven to be difficult during the proposal of this thesis. To our knowledge, we have not come across any works that have succeeded in such an attempt. The need for a stricter condition depends on the social problem at hand. In some cases, only evaluating the false negative rate with equal opportunity is sufficient (e.g. in disease classification problems) and there is a less urgent need for evaluating the false positive rate. The relaxed version, equal opportunity, seems to be doable. Regardless, we

have not come across any works on equal opportunity for multi-class classification either.

Sufficiency also allows for optimality compatibility, which makes it preferable to independence, but in the case of an imbalanced data set, it may not help to reduce the gap between social groups. This is because an imbalance will create a larger and larger gap over time. This results in calibrated scores having a distribution that is concentrated around the group mean, but remains unfair to some individuals when the data set is imbalanced (Corbett-Davies and Goel, 2018). Additionally, there are fewer works available on sufficiency as most works prefer independence or separation.

Based on this evaluation, we define two fairness metrics that are based on the fairness definition separation. It seems very intuitive to take the target value and sensitive attribute into account before prediction, which other definitions fail to do. Due to the difficulties surrounding the extension of equalized odds to the multi-class setting, we focus on equal opportunity for the first metric. As a result, we generalize the conditions (4) on the false negative rate for binary classification to the multi-class setting. For the second metric, we defined a metric ourselves which we will refer to as *equal overall mistreatment*. Even though the definition of equalized odds proved to be difficult during this thesis, equal overall mistreatment is based on the idea of equalized odds, and will be further explained in section 4.3.3.

4 Mathematical model

In this chapter, we present the linear program (LP) for the RUXG algorithm of Akyüz and Birbil (2021) in section 4.2, as well as the mathematical fairness conditions for equal opportunity and equal overall mistreatment to add to this LP, see section 4.3. Section 4.1 describes the preliminaries needed to define the LP of section 4.2.

4.1 Preliminaries

4.1.1 Notation

Table 1 summarizes the used notation of this chapter. For completeness, we include the notation used in later chapters, in Table 2.

Table 1: Summary of notation used

Sets	
$\mathcal{K}$	Set of class labels
$\mathcal{I}$	Set of samples
$\mathcal{J}$	Set of rules
$\mathcal{G}$	Set of groups, also the set different values of the sensitive attribute (e.g. {`asian, black, caucasian, hispanic`})

Parameters	
n	Number of data samples
K	Number of classes
k	A class label
G	A group value
g	Example of a group value (e.g. `asian`)
$\mathbf{x}_i$	Vector of features of sample i
y_i	True class label of sample i
$\hat{y}_i$	Predicted class label of sample i
$\mathbf{y}(\mathbf{x}_i)$	Indicator vector with the k^{th} element equal to one if $y_i = k$, and all other elements equal to $-\frac{1}{K-1}$
$\hat{\mathbf{y}}(\mathbf{x}_i)$	Indicator vector with the index of the largest element as the predicted class label to $\hat{y}_i$
a_{ij}	Indicator parameter equal to one if sample i is covered by rule j
$\mathbf{R}_j(\mathbf{x}_i)$	Indicator vector with the k^{th} element equal to one if sample i is covered by rule j, and all other elements equal to $-\frac{1}{K-1}$
$\hat{a}_{ij}$	Composed parameter to define the Hinge loss
$\hat{c}_j$	Costs of rule j

Variables	
v_i	Continuous variable that describes the classification error of sample i
w_j	Continuous variable that describes the weight of rule j

Table 2: Notation continued, introduced in section 5

Sets	
$\mathcal{P}_{k,g}$	Set of samples that share class label k and belong to group g
$\mathcal{I}_g$	Set of samples that belong to group g
D	Set of all unique pairs of groups in the data set
d	A pair of groups $d := (g, g')$
$\mathcal{J}_0$	Initial set of rules
$\mathcal{J}_t$	Constructed rule set at iteration t
$\mathcal{J}_-$	Rule set to add as columns in the CG procedure
$\mathcal{J}^*$	Interpretable set of rules

Parameters	
ϵ	Allowed level of unfairness
ϵ^*	Optimal allowed level of unfairness
$\bar{c}_j$	Reduced costs corresponding to rule j
p_{kgi}	Indicator parameter equal to one if $i \in \mathcal{P}_{kg}$ and zero otherwise
p_{gi}	Indicator parameter equal to one if $i \in \mathcal{P}_g$ and zero otherwise
α_B	Fairness evaluation function to measure unfairness in binary class and binary group classification, subject to equal opportunity
α_{EO}	Fairness evaluation function to measure unfairness in multi-class and multi-group classification, subject to equal opportunity
α_{EOM}	Fairness evaluation function to measure unfairness in multi-class and multi-group classification, subject to equal overall mistreatment

Variables	
β_i	Continuous dual variable between 0 and 1 that corresponds to the constraint (55)
γ_{kd}	Continuous dual variable that corresponds to fairness constraints
δ_{kd}	Continuous dual variable that corresponds to fairness constraints

4.1.2 Multi-Class Setting

We assume a multi-class classification problem with K classes. Let $\mathcal{K} = \{1, \ldots, K\}$ be the set of class labels. Suppose we have a training set $\mathcal{I}$ of n samples $(\mathbf{x}_i, y_i)$ with features $\mathbf{x}_i \in \mathbb{R}^p$, $p > 0$ and class label $y_i \in \mathcal{K}$. Let $\mathcal{J}$ be the total collection of rules that are used in classification by the original ensemble method, e.g. Random Forest, AdaBoost. For multi-class classification problems, we define a vector $\mathbf{y}(\mathbf{x}_i) \in \mathbb{R}^K$ such that, if $y_i = k$ we have

$$\mathbf{y}(\mathbf{x}_i) = \left(-\frac{1}{K-1}, \ldots, 1, \ldots, -\frac{1}{K-1}\right)^\intercal \tag{7}$$

where only the k^{th} entry of $\mathbf{y}(\mathbf{x}_i)$ has value one.

Let a_{ij} be an indicator variable that is equal to one if rule $j \in \mathcal{J}$ covers sample i and zero otherwise. If indeed $a_{ij} = 1$, we define another mapping $\mathbf{R}_j(\mathbf{x}_i)$ in the same manner as (7). The predicted class of sample i with features $\mathbf{x}_i$ is then given by

$$\hat{\mathbf{y}}(\mathbf{x}_i) = \sum_{j \in \mathcal{J}} a_{ij} \mathbf{R}_j(\mathbf{x}_i) w_j \tag{8}$$

where $w_j \geq 0$ is a weight associated with rule $j \in \mathcal{J}$. Again we have that $\hat{\mathbf{y}}(\mathbf{x}_i) \in \mathbb{R}^K$, but now the index of the largest entry of $\hat{\mathbf{y}}(\mathbf{x}_i)$ is considered to be the predicted class $\hat{y}_i \in \mathcal{K}$.

The goal of the classification framework for interpretable learning is to produce rule sets that are interpretable while maintaining a good accuracy or equivalently, a low total classification error. In other words, the LP will mine a subset of rules of $\mathcal{J}$ that is interpretable while at the same time it will minimize the total classification error.

4.1.3 Hinge Loss

As we wish to minimize the total loss, we need a loss function to evaluate the classification error. We use the *hinge loss*. Rosasco et al. (2003) argues that the hinge loss is often the loss function of choice for classification as it leads to better accuracy and some sparsity. Sparse solutions are good for us as it leads to a smaller number of rules. The hinge loss is also a convex function which results in a convex optimization problem.

In binary classification where we have class labels $\{0, 1\}$, the hinge loss is defined as

$$\ell(y) = \max(0, 1 - t \cdot y) \tag{9}$$

where t is the actual outcome and y is the output of the classifier. We can apply this to our setting and define the hinge loss as follows. Let $\hat{a}_{ij} = \kappa a_{ij} \mathbf{R}_j(\mathbf{x}_i)^T \mathbf{y}(\mathbf{x}_i)$ and $\kappa = \frac{K-1}{K}$. The hinge loss in the multi-class setting is then given by

$$\ell(\mathbf{y}(\mathbf{x}_i)) = \max\left\{1 - \sum_{j \in \mathcal{J}} \hat{a}_{ij} w_j, 0\right\} \tag{10}$$

If we sum up the hinge loss over al samples $\in \mathcal{I}$ we get total classification error that we wish to minimize:

$$\sum_{i \in \mathcal{I}} \max\left\{1 - \sum_{j \in \mathcal{J}} \hat{a}_{ij} w_j, 0\right\} \tag{11}$$

4.2 Linear Programming Formulation

We define the objective function of the LP. Let $c_j \geq 0$ be the cost of rule $j \in \mathcal{J}$ and let us have auxiliary variable v_i to describe the classification error of sample i. Using the hinge loss, we define

$$v_i \geq \max \left\{ 1 - \sum_{j \in \mathcal{J}} \hat{a}_{ij} w_j, 0 \right\} \tag{12}$$

For the objective function, we wish to minimize the total classification error for accuracy, and the number of rules used for interpretability. The number of rules used in classification is described by the used weights w_j of each rule. This results in the following linear program from Akyüz and Birbil (2021):

$$\text{minimize} \quad \sum_{i \in \mathcal{I}} v_i + \sum_{j \in \mathcal{J}} c_j w_j \tag{13}$$

$$\text{subject to} \quad \sum_{j \in \mathcal{J}} \hat{a}_{ij} w_j + v_i \geq 1 \ \forall i \in \mathcal{I} \tag{14}$$

$$v_i \geq 0 \qquad \forall i \in \mathcal{I} \tag{15}$$

$$w_j \geq 0 \qquad \forall j \in \mathcal{J} \tag{16}$$

The objective function is given by (13). The coefficient c_j in the objective function can be viewed as the cost of a rule j. Minimizing $c_j w_j$ over all rules helps us to avoid many rules of which the weight is nonzero and results in more spare solutions. Additionally, as also stated by Akyüz and Birbil (2021), in the application to other areas, rules in optimization problems have actual costs. In this case, we can consider the number of conditions in a rule to be the cost c_j of a rule. Hence, not only the number of rules influences the interpretability but also the length of a rule (Lakkaraju et al., 2016). The first set of constraints, given by (14), ensures that our definition of the classification error $v_i \geq \max \left\{ 1 - \sum_{j \in \mathcal{J}} \hat{a}_{ij} w_j, 0 \right\}$ holds.

This formulation sets the stage for us to include fairness into it by adding fairness constraints. When we define and add fairness constraints based on the fairness definitions of section 3.3, we can solve this LP according to the RUX and RUG algorithms of Akyüz and Birbil (2021). We refer to our classification tool as the FairRUX and FairRUG algorithms.

4.3 Fairness Conditions

We translate the fairness criteria for equal opportunity and equal overall mistreatment to mathematical conditions. Let $\mathcal{G}$ be the set of groups with distinct sensitive characteristics. We start by associating each data sample $i \in \mathcal{I}$ with a group $g \in \mathcal{G}$. We say that, if samples $i \in \mathcal{I}$ belong to group $g \in \mathcal{G}$ and samples $j \in \mathcal{I}$ belong to group $g' \in \mathcal{G}$, and both group of samples share the same target value, i.e. $y_i = y_j = k$, then the fairness criteria must hold between groups g and g'.

4.3.1 Assumption: One-vs-All Binary Classification

For some of the fairness definitions given in section 3.3, the conditions that the classifier has to satisfy are defined in the binary classification setting only. This is also the case for equal opportunity, where we wish our classifier

to produce an equal false negative rate among groups. However, in the multi-class setting, we do not consider classes to be positive or negative and so there is no distinction between a false (true) negative rate or a false (true) positive rate. To still be able to define such positive or negative rates in the multi-class setting, we assume a class $k \in \mathcal{K}$, the correct class, to be the positive class, and any class in $\mathcal{K} \setminus \{k\}$ to be a negative class. This is equivalent to a one-vs-all binary classification and boils down to an equal misclassification rate for each class.

4.3.2 Metric 1: Equal Opportunity

Equal opportunity requires an equal false negative rate for each class $k \in \mathcal{K}$, among all groups $g, g' \in \mathcal{G}$. With the one-vs-all assumption, the metric equal opportunity equals the condition that we require an equal misclassification rate for each class $k \in \mathcal{K}$, among all groups $g, g' \in \mathcal{G}$ in the multi-class setting. Therefore, we can impose the fairness condition based on equal opportunity as follows: we wish that

$$P(\hat{y} \neq y | y = k, G = g) = P(\hat{y} \neq y | y = k, G = g'), \qquad \forall k \in \mathcal{K} \text{ and } \forall g, g' \in \mathcal{G} \tag{17}$$

It may be unrealistic to find a classifier that satisfies this conditions perfectly while maintaining a good level of accuracy. Hence, we allow for a level of unfairness of $\epsilon \geq 0$. It follows that we can define the mathematical fairness condition of equal opportunity between two groups of people as

$$|P(\hat{y} \neq y | y = k, G = g) - P(\hat{y} \neq y | y = k, G = g')| \leq \epsilon, \qquad \forall k \in \mathcal{K} \text{ and } \forall g, g' \in \mathcal{G} \tag{18}$$

Now that we have translated equal opportunity to mathematical conditions, it remains to create fairness constraints for mathematical optimization out of (18).

4.3.3 Metric 2: Equal Overall Mistreatment

From our evaluation of the separation definition for fairness, we define another fairness criterion. We call this metric *equal overall mistreatment*. The definition of this metric is based on the idea of equalized odds. Recall that in equalized odds, we require an equal false positive and equal false negative rate among two groups. This enforces the accuracy between two groups to be equal and punishes models that perform well on the majority (Hardt et al., 2016). As the lack of accuracy is equivalent to mistreatment in the multi-class setting, we define an equal overall mistreatment as a fairness metric. Equal overall mistreatment requires that among all groups, the overall mistreatment should be the same. We assume the target value and sensitive attribute is given and we require a misclassification rate to be equal and so this criterion can be considered as a variation of separation. We can translate this to mathematical conditions by stating that for any pair of groups $g, g' \in \mathcal{G}$ the total classification errors of group g and g' should be the same. Hence, the overall mistreatment of a group can be calculated by accumulating the classification error over all classes in $\mathcal{K}$. And so an equal overall mistreatment among all groups is given by: we wish that

$$\sum_{k \in \mathcal{K}} P(\hat{y} \neq y = k | G = g) = \sum_{k \in \mathcal{K}} P(\hat{y} \neq y = k | G = g'), \qquad \forall g, g' \in \mathcal{G} \tag{19}$$

Again, this seems unrealistic to achieve and we allow for a level of unfairness of $\epsilon \geq 0$. This results in the following fairness condition of equal overall mistreatment as

$$\left| \sum_{k \in \mathcal{K}} P(\hat{y} \neq y = k | G = g) - \sum_{k \in \mathcal{K}} P(\hat{y} \neq y = k | G = g'), \right| \leq \epsilon \qquad \forall g, g' \in \mathcal{G} \tag{20}$$

It remains to create fairness constraints for mathematical optimization out of (20). So far, we have used several different terms to describe the fairness constraints and it might have become confusing what the difference between a fairness definition, condition and constraint is. Table 3 presents a short dictionary for the chosen words.

Table 3: Recap of wording in fairness

Wording	
Fairness Definition	A way to describe what fairness *means*.
Fairness Condition	We refer to the fairness condition as the mathematical way to express the fairness definition.
Fairness Constraint	The fairness constraint is the fairness condition translated to a mathematical optimization constraint. This is the constraint we actually add to the interpretable LP.

5 Methodology

This chapter presents the methodology of adjusting an interpretable classifier to a *fair and interpretable* classifier. We start by defining a set of fairness constraints, derived from the fairness definitions and conditions of section 4.3. Next, we continue with the FairRUX and FairRUG algorithm to obtain fair and interpretable decision rules.

5.1 Fairness Constraints

In this section we define the fairness constraints for both metrics based on the fairness conditions for the metrics equal opportunity and equal overall mistreatment. We use the continuous variable $v_i \geq 0$ as defined in equation (12) for the classification error.

5.1.1 Metric 1: Equal Opportunity

We translate condition (18) to a constraint such that it is applicable to the linear programming problem of section 4.2. Recall that we can view the expression $P(\hat{y} \neq y | y = k, G = g)$ as the classification error rate for samples that have true class label k and belong to group g. In other words, this is equivalent to v_i with $i \in \mathcal{P}_{k,g}$, where $\mathcal{P}_{k,g}$ is the set of samples samples that have true class label k and share sensitive attribute g. Hence, $\mathcal{P}_{k,g}$ for class $k \in \mathcal{K}$ and $g \in \mathcal{G}$ is given by

$$\mathcal{P}_{k,g} = \{i \in \mathcal{I} : y_i = k, G = g\} \tag{21}$$

Then for each pair of groups $g, g' \in \mathcal{G}$, the left-hand-side of the condition (18) for equal opportunity describes the misclassification gap between the classification errors of two samples belong that belong to group g and group

g' respectively, and that share the same class label. It follows that for each pair $g, g' \in \mathcal{G}$ we add the following constraints to the LP:

$$\sum_{i \in \mathcal{P}_{k,g}} v_i - \sum_{i \in \mathcal{P}_{k,g'}} v_i \leq \epsilon \tag{22}$$

$$\sum_{i \in \mathcal{P}_{k,g'}} v_i - \sum_{i \in \mathcal{P}_{k,g}} v_i \leq \epsilon \tag{23}$$

As equation (18) consists of an absolute value, we create two constraints per pair $g, g' \in \mathcal{G}$, to consider the possibility that $\sum_{i \in \mathcal{P}_{k,g}} v_i \geq \sum_{i \in \mathcal{P}_{k,g'}} v_i$, resulting in constraint (22), and the possibility that $\sum_{i \in \mathcal{P}_{k,g'}} v_i \geq \sum_{i \in \mathcal{P}_{k,g}}$, resulting in constraint (23).

5.1.2 Metric 2: Equal Overall Mistreatment

For equal overall mistreatment, we translate condition (19) to a constraint to add to the linear program in the same manner as for equal opportunity. Recall that $\sum_{k \in \mathcal{K}} P(\hat{y} \neq y = k | G = g)$, describes the total classification error of all samples that belong to the same group. It follows that the left-hand side of condition (19) describes the gap between the total misclassification rate between samples that belong to different groups. Hence, we group all data samples that share the sensitive attribute g, which results in the sets

$$\mathcal{I}_g = \{i \in \mathcal{I} : G = g\} \tag{24}$$

for each $g \in \mathcal{G}$. As we are interested in the total misclassification rate of a data sample that carries sensitive attribute g, we again use v_i. It follows that for each pair $g, g' \in \mathcal{G}$ we add the following constraints to the LP:

$$\sum_{i \in \mathcal{I}_g} v_i - \sum_{i \in \mathcal{I}_{g'}} v_i \leq \epsilon \tag{25}$$

$$\sum_{i \in \mathcal{I}_{g'}} v_i - \sum_{i \in \mathcal{I}_g} v_i \leq \epsilon \tag{26}$$

In the same manner, translating a condition that consists of an absolute value results in two optimization constraints per pair of groups.

5.2 Linear Program subject to Fairness Constraints

The fairness constraints allow us to define the fair LP. In case of equal opportunity, we obtain the following linear programming problem:

$$\text{(EO):} \quad \text{minimize} \quad \sum_{i\in\mathcal{I}} v_i + \sum_{j\in\mathcal{J}} c_j w_j \tag{27}$$

$$\text{subject to} \quad \sum_{j\in\mathcal{J}} \hat{a}_{ij} w_j + v_i \geq 1 \qquad \forall i \in \mathcal{I} \tag{28}$$

$$\sum_{i\in\mathcal{P}_{k,g}} v_i - \sum_{i\in\mathcal{P}_{k,g'}} v_i \leq \epsilon \qquad \forall k \in \mathcal{K} \text{ and } \forall g, g' \in \mathcal{G} \tag{29}$$

$$\sum_{i\in\mathcal{P}_{k,g'}} v_i - \sum_{i\in\mathcal{P}_{k,g}} v_i \leq \epsilon \qquad \forall k \in \mathcal{K} \text{ and } \forall g, g' \in \mathcal{G} \tag{30}$$

$$v_i \geq 0 \qquad \forall i \in \mathcal{I} \tag{31}$$

$$w_j \geq 0 \qquad \forall j \in \mathcal{J} \tag{32}$$

where the objective function (27) minimizes the hinge loss and number of used rules simultaneously, with rule costs coefficients $c_j \geq 0, j \in \mathcal{J}$. Constraint (28) defines the classification error in the multi-class setting and (29) and (30) are the added fairness constraints. The addition of the fairness constraints for every class in $\mathcal{K}$, and for every unique pair of groups of $\mathcal{G}$ results in adding a total number of,

$$2 \cdot |\mathcal{K}| \cdot \frac{|\mathcal{G}| \cdot (|\mathcal{G}| - 1)}{2} \tag{33}$$

fairness constraints to the LP of section 4.2. Here, $\frac{|\mathcal{G}|\cdot(|\mathcal{G}|-1)}{2}$ is the number of unique group pairs in $\mathcal{G}$. In the future, we refer to set D as the set of all unique pairs of groups in $\mathcal{G}$. Hence, we add the fairness constraints for every pair $(g, g') \in D$ and $|D| = \frac{|\mathcal{G}|\cdot(|\mathcal{G}|-1)}{2}$.

To obtain the linear programming problem for equal overall mistreatment, we can replace (29) and (30) of the model EO with the constraints (25) and (26) respectively. We write out this LP for the purpose of defining its dual problem in the next section. Thus, this results in a linear program that is subject to a fairness constraint based on equal overall mistreatment:

$$\text{(EOM):} \quad \text{minimize} \quad \sum_{i\in\mathcal{I}} v_i + \sum_{j\in\mathcal{J}} c_j w_j \tag{34}$$

$$\text{subject to} \quad \sum_{j\in\mathcal{J}} \hat{a}_{ij} w_j + v_i \geq 1 \qquad \forall i \in \mathcal{I} \tag{35}$$

$$\sum_{i\in\mathcal{I}_g} v_i - \sum_{i\in\mathcal{I}_{g'}} v_i \leq \epsilon \qquad \forall d \in D \tag{36}$$

$$\sum_{i\in\mathcal{I}_{g'}} v_i - \sum_{i\in\mathcal{I}_g} v_i \leq \epsilon \qquad \forall d \in D \tag{37}$$

$$v_i \geq 0 \qquad \forall i \in \mathcal{I} \tag{38}$$

$$w_j \geq 0 \qquad \forall j \in \mathcal{J} \tag{39}$$

where $d = (g, g')$ in constraints (36) and (37). In the case of equal overall mistreatment, the fair LP again requires two set of fairness constraints for each pair of groups of $\mathcal{G}$. This results in adding $2 \cdot |D|$ constraints to the LP of section 4.2.

5.2.1 Fair Rule Extraction (FairRUX)

The first classification tool is referred to as the Fair Rule Extraction (FairRUX) algorithm. Let for the model EO the *master problem* be given by equations (27)-(32), and for the model EOM by equations (34)-(39). For the existing ensemble methods that are needed to mine an initial collection of rules, we use the Random Forests (RF) and AdaBoost (ADA) classifiers. We train these tree ensemble models on the given data set. This results in several trees with each a set of leaves. As each leaf corresponds to a rule, we can consider the set of leaves to be our set of rules $\mathcal{J}$. Solving the linear program subject to fairness constraints allows us to extract the fair rules that were most critical in the classification with the RF or ADA. The result is an interpretable set of rules $\mathcal{J}^*$ used for fair classification. In other words, $\mathcal{J}^*$ is depicted by all non-zero weights in the final solution of the LP.

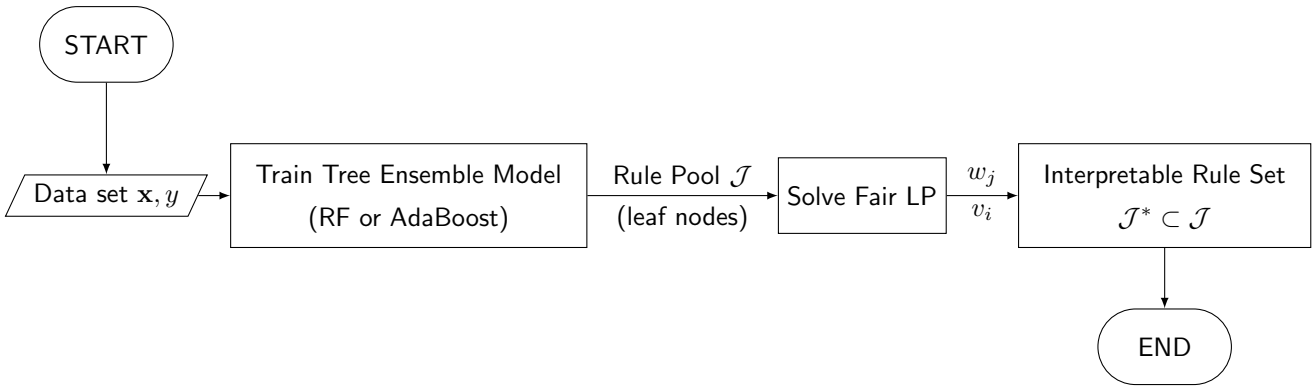


Figure 1: Fair rule extraction algorithm

5.2.2 Fair Rule Generation Algorithm (FairRUG)

In the FairRUX algorithm, we have an existing set of rules $\mathcal{J}$ of which we can mine a smaller subset. However, suppose now that we do not have such a set of rules, or the set of rules is too large to be constructed by an existing machine learning model. This means that we have to generate the rules ourselves iteratively. In the fairness setting, we call this the fair rule generation (FairRUG) algorithm. As explained earlier, the rules of each decision tree correspond to the columns in the fair LP. This allows us to apply the well-known column generation (CG) procedure. Figure 2 shows a simplified version of the FairRUG algorithm.

We see in Figure 2 that the FairRUG actually does start with an initial rule set $\mathcal{J}_0$. We train a decision tree on the training data $\mathbf{x}, y$ and we let its leaves serve as a starting set of rules. In some ways, this looks similar to the FairRUX algorithm. However, in this case, we train one decision tree as opposed to the entire set of rules obtained from forests and ensembles. This results in a faster algorithm with far fewer rules in our rule pool. The next step in the algorithm is to expand this initial rule pool with the CG procedure. The part of the algorithm as shown in Figure 2 that contains the CG procedure is explained in the next section, but note that it is only used for the FairRUG algorithm.

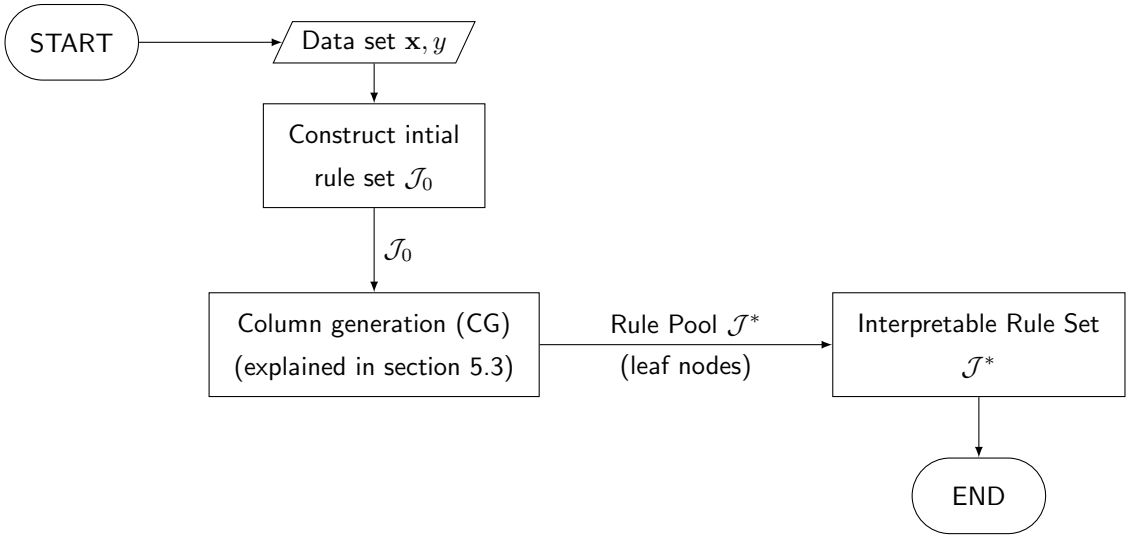


Figure 2: Fair rule generation algorithm

5.3 Column Generation Procedure

The main part of the RUG algorithm consists of a column generation (CG) procedure. Column generation is an efficient approach to solving linear programs where the number of variables is too large to consider explicitly. The idea is to start with a subset of these variables and increase this number iteratively, and thus CG avoids the need for considering all variables. The LP with the restricted number of variables is called the *restricted master problem* (RMP). As the decision variables of a LP make up the columns, generating variables is equivalent to generating columns (Desaulniers et al., 2005). Next, starting with an initially smaller column pool, the choice of which columns to add in the next iteration depends on a subproblem. After solving the restricted master problem, the optimal solution to its dual problem is obtained. With this dual solution, we can define the *pricing problem* (PP) which is solved to identify the columns to add to our column pool in the next iteration. The pricing problem identifies the columns that result in negative *reduced costs* (RC). These columns are the only candidates that may improve the objective function value when they are added to the column pool. The CG procedure continues after adding these columns to the column pool with the next iteration where new columns are identified again. Linear programming theory ensures that when no columns can be found that result in negative reduced costs, optimality is reached and we can stop the CG procedure (Desaulniers et al., 2005).

In our case, we start with a subset of rules which can be expressed by their weights, i.e. columns, and so our rules correspond to the columns of the fair LP. It follows that we define our restricted master problem by starting with a subset of the decision variables w_j, so $j \in \mathcal{J}_t \subset \mathcal{J}$ where t describes the iteration of the CG procedure we are currently in. In the next section, we define the restricted master problem, dual problem, and pricing problem for each fairness metric separately. Figure 3 describes the well-known column generation procedure in general.

Construction of the initial restricted master problem as shown in Figure 3, depends on the construction of a decision tree to create an initial rule pool. Algorithm 1 describes the steps of the complete FairRUG algorithm.

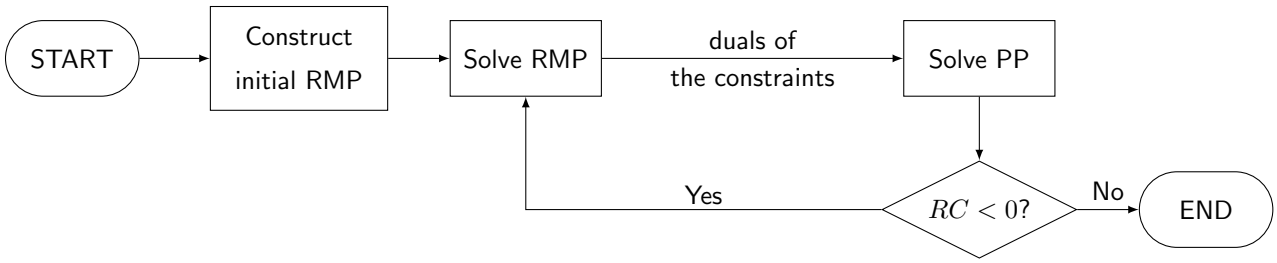


Figure 3: Column Generation Procedure

Algorithm 1 Fair Rule Generation Algorithm

Result: Interpretable rule set $\mathcal{J}^*$ subject to fairness constraints

Input: Training data $(\mathbf{x}_i, y)_{i\in\mathcal{I}}$

$t = 0$

$\mathcal{J}_0 \leftarrow DecisionTree((\mathbf{x}_i, y)_{i\in\mathcal{I}})$

while $RC(\mathcal{J}_t) < 0$ **do**

  $\beta_i, \gamma_{kd}, \delta_{kd} \leftarrow SolveFairLP(\mathcal{J}_t)$

  $\mathcal{J}_- \leftarrow SolvePP(\beta_i, \gamma_{kd}, \delta_{kd})$

  if $\mathcal{J}_- \neq \emptyset$ **then**

    $\mathcal{J}^* \leftarrow \mathcal{J}_-$

    return $\mathcal{J}^*$

  end

  $t \leftarrow t + 1$

  $\mathcal{J}_t = \mathcal{J}_{t-1} \cup \mathcal{J}_-$

end

5.3.1 Metric 1: Equal Opportunity

After the construction of the initial rule pool, we generate the rest of the rules according to the CG procedure. We define the restricted master problem that restricts our number of columns. For equal opportunity, this is equivalent to the fair primal linear program for equal opportunity (EO) of section 5.2. Clearly, instead of considering the whole rule pool $\mathcal{J}$, we consider a subset $\mathcal{J}_t$, where $t = 0$ indicates the start and $t = 1$ is the first iteration of the CG procedure.

Restricted Master Problem

To define the dual problem later on, it is easier to rewrite the fair LP such that the summations on the left-hand-side of constraints of (29) and (30), are summations over all samples in $\mathcal{I}$. We introduce indicator variable p_{kgi} defined as

$$p_{kgi} = \begin{cases} 1 & \text{if sample } i \in \mathcal{P}_{kg} \\ 0 & \text{otherwise} \end{cases}$$

Recall that $\mathcal{P}_{kg}$ is the set of samples that share class label k and belong to group g. With the definition of p_{kgi} and $\mathcal{J}_t \subset \mathcal{J}, t \geq 0$, the primal restricted master problem subject to equal opportunity conditions becomes

$$
\begin{aligned}
(\text{EO}_{RMP}): \quad \text{minimize} \quad & \sum_{i\in\mathcal{I}} v_i + \sum_{j\in\mathcal{J}_t} c_j w_j && && (40)\\
\text{subject to} \quad & \sum_{j\in\mathcal{J}_t} \hat{a}_{ij} w_j + v_i \geq 1 && \forall i \in \mathcal{I} && (41)\\
& \sum_{i\in\mathcal{I}} (p_{kgi} - p_{kg'i}) v_i \leq \epsilon && \forall k \in \mathcal{K} \text{ and } \forall g, g' \in \mathcal{G} && (42)\\
& \sum_{i\in\mathcal{I}} (p_{kg'i} - p_{kgi}) v_i \leq \epsilon && \forall k \in \mathcal{K} \text{ and } \forall g, g' \in \mathcal{G} && (43)\\
& v_i \geq 0 && \forall i \in \mathcal{I} && (44)\\
& w_j \geq 0 && \forall j \in \mathcal{J}_t && (45)
\end{aligned}
$$

Dual problem

To continue with the pricing problem in order to identify candidate columns, we need the dual variables corresponding to the constraints of the dual problem. Hence, we define the dual problem. Given the restricted master problem for equal opportunity, let its dual variables $\beta_i, i \in \mathcal{I}$ correspond to (41). Recall that D is the set of each unique pair of groups. Then we denote the dual variables γ_{kd} and δ_{kd} that correspond to the set of constraints (42) and (43) respectively, for each $k \in \mathcal{K}$ and $(g, g') =: d \in D$. Furthermore, let $p_{kdi} := p_{kgi} - p_{kg'i}$ and so $-p_{kdi} = p_{kg'i} - p_{kgi}$, then the *dual restricted master problem* at iteration t, subject to equal opportunity conditions becomes

$$
\begin{aligned}
(\text{EO-D}): \quad \text{maximize} \quad & \sum_{i\in\mathcal{I}} \beta_i - \epsilon \cdot \sum_{k\in\mathcal{K}} \sum_{d\in D} (\gamma_{kd} + \delta_{kd}) && && (46)\\
\text{subject to} \quad & \sum_{i\in\mathcal{I}} \hat{a}_{ij} \beta_i \leq c_j && \forall j \in \mathcal{J}_t && (47)\\
& \beta_i + \sum_{k\in\mathcal{K}} \sum_{d\in D} p_{kdi} (\gamma_{kd} + \delta_{kd}) \leq 1 && \forall i \in \mathcal{I} && (48)\\
& \beta_i \geq 0 && \forall i \in \mathcal{I} && (49)\\
& \gamma_{kd} \geq 0 && \forall k \in \mathcal{K}, \forall d \in D && (50)\\
& \delta_{kd} \geq 0 && \forall k \in \mathcal{K}, \forall d \in D && (51)
\end{aligned}
$$

Note that the optimal dual solution is equal to the optimal primal solution due to strong duality as the fair primal linear program is a convex linear optimization problem. We use this property in our motivation to define and solve the pricing problem.

Pricing Problem

To identify which rule j' or set of rules $\mathcal{J}_-$ to add to our rule pool $\mathcal{J}_t$ in the next iteration, we define and solve the pricing problem. The pricing problem identifies which rules would improve the objective function value of the dual problem. Let the optimal solution of EO-D, at iteration t, be denoted by $\boldsymbol{\beta}_{EO}^{(t)}, \boldsymbol{\gamma}_{EO}^{(t)}$ and $\boldsymbol{\delta}_{EO}^{(t)}$.

Then the objective function value of the dual problem improves if we are able to find at least one rule $j' \in \mathcal{J} \setminus \mathcal{J}_t$ such that

$$\bar{c}_{j'} = c_{j'} - \sum_{i \in \mathcal{I}} \hat{a}_{ij'} \beta_{EO_i}^{(t)} < 0 \tag{52}$$

The left-hand side of (52) describes the reduced costs $\bar{c}_j$ of column j' and is thus associated with the decision variable $w_{j'}$. Recall that linear programming theory ensures that if we cannot find any rules that result in negative reduced costs, then we can terminate the column generation algorithm, and the solution to the restricted master problem is also optimal for the LP. To find the rule j' such that (52) holds, we define the following *pricing problem*:

$$\text{(PP-EO):} \qquad \text{minimize}_{j \in \mathcal{J} \setminus \mathcal{J}_t} \qquad c_j - \sum_{i \in \mathcal{I}} \hat{a}_{ij} \beta_{EO_i}^{(t)} \tag{53}$$

In the first iteration of the column generation procedure, we can solve the pricing problem by computing the reduced costs of each rule in the initial rule set $\mathcal{J}_0$ and adding a rule with negative reduced costs. This initial rule pool $\mathcal{J}_0$ is created by training a decision tree with a Decision Tree Classifier from the Python programming library scikit-learn (Pedregosa et al., 2011). After adding a rule, the next iteration starts and we go back to the restricted master problem, only now do we have a rule pool and so there is no need to train a decision tree. We repeat the steps of computing reduced costs to identify rules until no negative reduced costs can be found. If we find multiple rules at once that result in negative reduced costs, we add these rules as a set $\mathcal{J}_-$.

Note that the dual variables $\boldsymbol{\gamma}_{EO}^{(t)}$ and $\boldsymbol{\delta}_{EO}^{(t)}$ do not explicitly appear in the reduced costs associated with variable w_j, i.e. the rules, as these are the dual multipliers that correspond to the fairness constraints of the fair LP. However, $\boldsymbol{\beta}_{EO_i}^{(t)}$ is dependent on these fairness multipliers and so $\bar{c}_j$ is still implicitly affected by the fairness constraints.

5.3.2 Metric 2: Equal Overall Mistreatment

Restricted Master Problem

For equal overall mistreatment, we define the restricted master problem by rewriting the summation in the fairness constraints (36) and (37) as a summation over all samples in $\mathcal{I}$. As for EO$_{RMP}$, this makes it easier to write down the dual problem later on. We introduce indicator variable p_{gi} defined as

$$p_{gi} = \begin{cases} 1 & \text{if sample } i \in \mathcal{I}_g \\ 0 & \text{otherwise} \end{cases}$$

With the definition of p_{gi} and $\mathcal{J}_t \subset \mathcal{J}, t \geq 0$, the primal restricted master problem subject to equal overall mistreatment conditions becomes

$$\text{(EOM}_{RMP}\text{):} \quad \text{minimize} \quad \sum_{i\in\mathcal{I}} v_i + \sum_{j\in\mathcal{J}_t} c_j w_j \tag{54}$$

$$\text{subject to} \quad \sum_{j\in\mathcal{J}_t} \hat{a}_{ij} w_j + v_i \geq 1 \qquad \forall i \in \mathcal{I} \tag{55}$$

$$\sum_{i\in\mathcal{I}} (p_{gi} - p_{g'i}) v_i \leq \epsilon \qquad \forall g, g' \in \mathcal{G} \tag{56}$$

$$\sum_{i\in\mathcal{I}} (p_{g'i} - p_{gi}) v_i \leq \epsilon \qquad \forall g, g' \in \mathcal{G} \tag{57}$$

$$v_i \geq 0 \qquad \forall i \in \mathcal{I} \tag{58}$$

$$w_j \geq 0 \qquad \forall j \in \mathcal{J} \tag{59}$$

Dual Problem

Given the restricted master problem subject to equal overall mistreatment, we denote the dual variables corresponding to (55) by $\beta_i, i \in \mathcal{I}$. Recall that D is the set of each unique pair of groups, so we have $(g, g') \in D$. Then we denote the dual variables γ_d and δ_d that correspond to the set of constraints (42) and (43) respectively, for each $(g, g') =: d \in D$. Moreover, let $p_{di} := p_{gi} - p_{g'i}$ and thus $-p_{di} = p_{g'i} - p_{gi}$, then the *dual restricted master problem* at iteration t, subject to equal overall mistreatment conditions becomes

$$\text{(EOM-D):} \quad \text{maximize} \quad \sum_{i\in\mathcal{I}} \beta_i - \epsilon \cdot \sum_{d} (\gamma_d + \delta_d) \tag{60}$$

$$\text{subject to} \quad \sum_{i\in\mathcal{I}} \hat{a}_{ij} \beta_i \leq c_j \qquad \forall j \in \mathcal{J}_t \tag{61}$$

$$\beta_i + \sum_{d\in D} p_{di} (\gamma_d - \delta_d) \geq 1 \qquad \forall i \in \mathcal{I} \tag{62}$$

$$\beta_i \geq 0 \qquad \forall i \in \mathcal{I} \tag{63}$$

$$\gamma_d \geq 0 \qquad \forall d \in D \tag{64}$$

$$\delta_d \geq 0 \qquad \forall d \in D \tag{65}$$

Pricing Problem

Let the optimal solution of EOM-D, at iteration t be given by $\boldsymbol{\beta}_{EOM}^{(t)}, \boldsymbol{\gamma}_{EOM}^{(t)}$ and $\boldsymbol{\delta}_{EOM}^{(t)}$. Then, similar to the reduced costs for equal opportunity, we can again improve the objective function value of the dual problem if we find at least one rule $j' \in \mathcal{J} \setminus \mathcal{J}_t$ such that the reduced costs of the found column under equal overall mistreatment is negative, that is

$$\bar{c}_{j'} = c_{j'} - \sum_{i\in\mathcal{I}} \hat{a}_{ij'} \beta_{EOM}{}_i^{(t)} < 0 \tag{66}$$

Similarly as for equal opportunity, it follows that the pricing problem for the metric equal overall mistreatment is given by

$$\text{(PP-EOM):} \qquad \text{minimize}_{j \in \mathcal{J} \setminus \mathcal{J}_t} \qquad c_j - \sum_{i \in \mathcal{I}} \hat{a}_{ij} \beta_{EOM_i}^{(t)} \tag{67}$$

Solving the pricing problem for equal overall mistreatment occurs in the same manner as for equal opportunity.

In practice, we perform the FairRUX and FairRUG algorithm for both equal opportunity and equal overall mistreatment separately as they are two different algorithms.

5.4 Fairness Evaluation

To evaluate the fairness performance of the FairRUX and FairRUG algorithms, we measure how unfair our classifiers are. We define *fairness evaluation functions* based on equal opportunity and equal overall mistreatment to measure the unfairness of each prediction.

5.4.1 Metric 1: Equal Opportunity

For equal opportunity, we use two different functions to assess the fairness performance of our classifier. The reason for this is that there are works that succeed in performing fair binary classification with the metric equal opportunity, where $|\mathcal{G}| = 2$ (Lawless et al., 2021). Consequently, for benchmarking purposes, we assess the unfairness in the same manner as Lawless et al. (2021). This is not possible for multi-class classification and so we define our own fairness evaluation function in the multi-class setting.

Binary Classification, Binary Group

In the binary setting, requiring equal opportunity is equivalent to requiring an equal false negative rate between two groups, for each class. Now that we have $|\mathcal{K}| = 2$, we can consider one of the two classes to be the more advantageous outcome. Additionally, similar to Lawless et al. (2021), we assume a binary group setting, i.e. $|\mathcal{G}| = 2$. Suppose $\mathcal{G} = \{0, 1\}$ and $\mathcal{K} = \{0, 1\}$ with $k = 1$ being the more advantageous outcome of the two classes. In practice, the more advantageous outcome depends on the data set at hand. Recall that we can express the false negative rate of a classifier with samples belonging to group g, with the conditional probability definition for accuracy as

$$\text{false negative rate (FNR)} : \qquad P(\hat{y} = 0 | y = 1, G = g) \tag{68}$$

Then we define the unfairness between group one and two as the gap between false negative rates of the two distinct groups $g = 0$ and $g' = 1$, that is

$$\text{GAP}_{FNR} = \quad |P(\hat{y} = 0 | y = 1, G = 0) - P(\hat{y} = 0 | y = 1, G = 1)| \tag{69}$$

Recall that the set $\mathcal{P}_{kg}$ contains all samples that share class label k and belong to group g. Let $u_i \in \mathbb{N}$ be equal to one if the sample i is misclassified and zero otherwise. From (69), we define the fairness evaluation function α_B as follows:

$$\alpha_B(\mathbf{x}, y) := \left| \frac{1}{|\mathcal{P}_{1,0}|} \sum_{i \in \mathcal{P}_{1,0}} u_i - \frac{1}{|\mathcal{P}_{1,1}|} \sum_{i \in \mathcal{P}_{1,1}} u_i \right| \tag{70}$$

Note that the output of $\alpha_B(\mathbf{x}, y)$ is a fraction between zero and one, so we will refer to the level of unfairness as α_B-percent unfairness.

Multi-Class classification

Now we consider $|\mathcal{K}| > 2$ and $|\mathcal{G}| > 2$. We keep the indicator variable u_i to count the number of total misclassifications in a prediction as defined in the previous section. Let $\alpha_{k,d}$ be the level of unfairness between two samples that share class label k, but belong to different groups $d = (g, g') \in D$. We assume that

$$\alpha_k(g, g') := \left| \frac{1}{|\mathcal{P}_{kg}|} \sum_{i \in \mathcal{P}_{kg}} u_i - \frac{1}{|\mathcal{P}_{kg'}|} \sum_{i \in \mathcal{P}_{kg'}} u_i \right| \tag{71}$$

Then we define the level of unfairness unfairness of the total classifier $\alpha_{EO}(\mathbf{x}, y)$, as the level of unfairness between the groups g, g' that result in the largest disparity for $\alpha_k(g, g')$ over all classes $\in \mathcal{K}$. This is equivalent to

$$\alpha_{EO}(\mathbf{x}, y) = \frac{1}{|\mathcal{K}|} \sum_{k \in \mathcal{K}} \max_{g, g' \in \mathcal{G}} \alpha_k(g, g') \tag{72}$$

Again, it holds that $\alpha_{EO}(\mathbf{x}, y) \in [0, 1]$ and so we refer to the level of unfairness of the prediction as α_{EO}-percent. Another option is to compute the total unfairness in a classification. This requires changing the max-function to a summation, and removing scaling by the number of elements in $\mathcal{K}$. Note that (72) will not give the false negative rate if we were to use it in the binary class and binary group setting. The unfairness evaluation function given in (72) would also assess false positive rates and if the gap between false positive rates would be bigger than between false negative rates, then the false negative rate would not be considered in the multi-class setting.

5.4.2 Metric 2: Equal Overall Mistreatment

For the metric equal overall mistreatment, we do not distinguish between binary or multi-class classification. We assume again that $u_i \in \mathbb{N}$ is equal to one if sample i is misclassified and zero otherwise. Then we define the fairness evaluation function α_{EOM} as the gap between the misclassification rates between two groups that result in the largest disparity. This is equivalent to

$$\alpha_{EOM}(\mathbf{x}, y) := \max_{g, g' \in \mathcal{G}} \left| \frac{1}{|\mathcal{I}_g|} \sum_{i \in \mathcal{I}_g} u_i - \frac{1}{|\mathcal{I}_{g'}|} \sum_{i \in \mathcal{I}_{g'}} u_i \right| \tag{73}$$

We have again that $\alpha_{EOM}(\mathbf{x}, y) \in [0, 1]$ and so we speak of a level of unfairness of α_{EOM}-percent.

6 Computational Study

In this chapter, we perform computational experiments to test the fair classification tools that are built by the FairRUX and FairRUG algorithms. First, we present the data sets in section 6.1. Next, we perform classification on these data sets with the traditional Random Forests and Adaptive Boosting models and compare them to the results from the fair and interpretable classifiers. We also present what a sample ruleset looks like to demonstrate its simplicity and intuitive appeal. Lastly, we compare our results for binary classification to the work of Lawless et al. (2021).

6.1 Data

In this section, we describe the data sets used and any data cleaning that has been performed. In order to create fair classifications, we need data sets where 'unfairness' between sensitive attribute values is undesirable. For example, in loan approval systems, we wish to avoid unfairness between the sensitive attribute race with values Hispanic, Asian, Black, and White. Whereas, when classifying whether an individual has a certain illness or not, we wish differences between men and women to appear. Hence, the sensitive attribute gender would not require 'fair' classification in this case. The data sets used for these computational experiments are all data sets available from Kaggle or the UCI Machine Learning Repository (Dua and Graff, 2017). Encoding of the data is not needed as both RUX and RUG can work with continuous or categorical features (Akyüz and Birbil, 2021). Table 5 summarizes all data sets.

6.1.1 Binary Classification

Even though our classifier is directly usable in multi-class classification, we also consider data sets for binary classification. We choose the same data sets as Lawless et al. (2021) for benchmarking purposes.

Adult data set

The adult data set is a multivariate data set from the United States Census Bureau (Kohavi and Becker, 1996), where the prediction task is to determine whether a person will make over $50.000 a year in salary, based on attributes such as gender, education, and hours per workweek. The application of using the fair classifier in this data set can be for example in loan approval systems. A person that earns over $50.000 a year, might seem more trustworthy to loan money to. The data set contains 32.561 samples and fourteen attributes, all of which are used. Following Lawless et al. (2021), the sensitive attribute for this data set is `gender` (i.e. man or woman). It follows that $|\mathcal{K}| = 2$ and $|\mathcal{G}| = 2$. This means we add four fairness constraints to the LP to obtain a fair classification problem. We assess the fairness performance according to the fairness evaluation function defined in section 5.4 and more specifically for equal opportunity, we consider the fairness evaluation function for binary class and binary group, presented in section 5.4.1.

COMPAS data set

COMPAS (Correctional Offender Management Profiling for Alternative Sanctions) is a well-known commercial algorithm used by courts and parole authorities to assess the possibility of recidivism of a criminal defendant. A two-year follow-up research to see if crimes were actually committed after two years shows that COMPAS is skewed in favor of Caucasian defendants and against African American convicts (Angwin et al., 2016). As a result, the COMPAS data set appears to be an acceptable data set to test our fair classification algorithm on.

There is a cleaned COMPAS data set available from Kaggle that is also used by Lawless et al. (2021). Following the approach of Lawless et al. (2021), we restrict the data to only look at Caucasian and African American respondents and thus remove all other races. Naturally, the sensitive attribute of this data set is race, with only two values. This results in a binary classification with two groups, so again we have $|\mathcal{K}| = 2$ and $|\mathcal{G}| = 2$,

with four fairness constraints to add to the LP. We use the same fairness evaluation functions as for the adult data set.

Default data set

The Default data set is a data set available from the UCI Machine Learning Repository that contains information on default payments of credit card clients in Taiwan (Yeh, 2016). The prediction task is to determine whether a client is likely to default payment, based on attributes such as gender, education, age, and history of past payments. The social relevance of using this data set is for departments such as risk management in any kind of area, to assess whether a client is credible or not. This data set contains 30.000 samples and we again assume the sensitive attribute to be gender as in Lawless et al. (2021). It follows that $|\mathcal{K}| = 2$ and $|\mathcal{G}| = 2$ and so we add four fairness constraints to add to the LP and assess the fairness performance with the same fairness evaluation functions as for the adult and COMPAS data set.

6.1.2 Multi-Class Classification

The following data sets are all applicable to multi-class classification. In contrast to other works in most of the literature, and our own experiments in binary classification, in the multi-class setting we also consider there to be more than two social groups, i.e. $|\mathcal{G}| > 2$.

Student data set

The Student Performance data set contains student achievements in Portuguese secondary education (Cortez, 2014). The purpose of classification in this data set is to predict a students' performance in secondary school, based on attributes such as student grades, demographic and social features. As mentioned in (Cortez and Silva, 2008), the application of this data set was to improve the Portuguese school system and to find out what factors affect a students' performance.

This data set contains 649 samples and a numerical output ranging from zero to twenty to describe a students' performance. We categorized the labels into five categories to limit the number of fairness constraints we need to add to the LP. The categorization of the performance labels is based on the academic equivalence of Portuguese grades to American grades that range from A to E. We consider the sensitive attribute to be gender. This results in $|\mathcal{K}| = 5$ and $|\mathcal{G}| = 2$ and so we add ten fairness constraints to the LP.

Table 4: Grade system conversion to class labels (de Ensino Superior, 2018)

Portugal	United States	Class Label
18-20.00	A	0
14.00-17.99	B	1
10.00-13.99	C	2
7.00-9.99	D	3
0.00-6.99	E	4

Nursery data set

The Nursery data set was derived from a decision model that was originally developed in 1989 to rank applications for nursery schools (Snellen and de Donk, 1989). Nursery schools in Ljubljana, Slovenia experienced an excessive amount of enrollments and thus required such a ranking system. Many of the applications that were turned down required an objective explanation. Like today, negative events must be explained.

The data set contains 12.960 samples and five classes, ranging from not recommending a child to recommending a child with priority. The attributes include the family structure, employment of the parents, and housing conditions. We consider the sensitive attribute to be the social health and picture of the family. This sensitive attribute has three values: *non-problematic, slightly problematic* and *problematic*. With $|\mathcal{K}| = 5$ and $|\mathcal{G}| = 3$, we add thirty fairness constraints to the LP.

Law data set

The Law School Admission Council National Longitudinal Study (LSAC) was undertaken in response to rumors that suggested that bar passage rates among examinees of color were unfairly lower compared to their Caucasian peers (Wightman, 1998). The Law data set presents the national longitudinal bar passage data, with the prediction task to determine a students' GPA in law school, based on attributes such as LSAT score, undergraduate GPA, gender, and race. The outcome is divided into four equidistant quantiles and so we have $|\mathcal{K}| = 4$. We follow the methodology of Denis et al. (2021) by choosing `race` to be the sensitive attribute and cleaning the data such that we have only three attributes and two social groups (white and non-white students). With $|\mathcal{K}| = 4$ and $|\mathcal{G}| = 2$ we add eight fairness constraints to the LP.

Table 5: Data sets used for fair classification

Data set	Type	# observations	# attributes	# classes	# groups	Sensitive attribute	Area
Adult	Binary	32561	14	2	2	Gender	Economics
COMPAS	Binary	5278	7	2	2	Race	Criminal Justice
Default	Binary	30000	24	2	2	Gender	Finance
Student	Multi-class	649	33	5	2	Gender	Education
Nursery	Multi-class	12960	8	5	3	Social Health	Education
Law	Multi-class	20649	3	4	2	Race	Education

6.2 Numerical Experiments

We present the results of the FairRUX and FairRUG algorithm in section 6.2.1. Next, we compare the results to other fairness methods in the binary setting, in section 6.2.4. The FairRUX and FairRUG algorithms are implemented in the programming language Python, version 3.7. We used Gurobi 9.0 as the commercial LP solver. All runs are taken on a 3.2 GHz 8-core Apple M1 processor with 16 GB RAM.

6.2.1 Results

First, we train Random Forests (RF) and Adaptive Boosting (ADA) models. We compared the fairness of both models with the fairness evaluation functions, and used 10-fold cross-validation to select the hyperparameters that resulted in the *least amount of unfairness*. We chose a value for the maximum depth of each tree from the set $\{1, 2, 3\}$ and the number of trees in the forests from the set $\{100, 200\}$. Then, we apply the FairRUX algorithm to the set of rules that are created by the trained RF and ADA models. We refer to these results as FairRUXRF for the FairRUX algorithm based on RF, and FairRUXADA for the FairRUX algorithm based on ADA. To find the optimal allowed level of unfairness ϵ^*, that is, the ϵ for which the unfairness of a method is lowest, we varied the strictness of fairness ϵ between $[0, 1]$. See Table 6 for the parameters used. Ranging ϵ between zero and one also shows us the effect of relaxing the fairness constraints on the predictive performance and the fairness level of the methods. When we impose strict fairness conditions on the LP, we consider $\epsilon = 0$. This means that we do not allow for any gap of unfairness to exist between two groups. Also, $\epsilon = 1$ implies that we relax the fairness constraints completely. Computing the level of unfairness for the traditional methods and the fair methods with $\epsilon = 1$ gives us a good idea of whether there was any unfairness present at all in the data sets.

Table 6: Overview of hyperparameters used

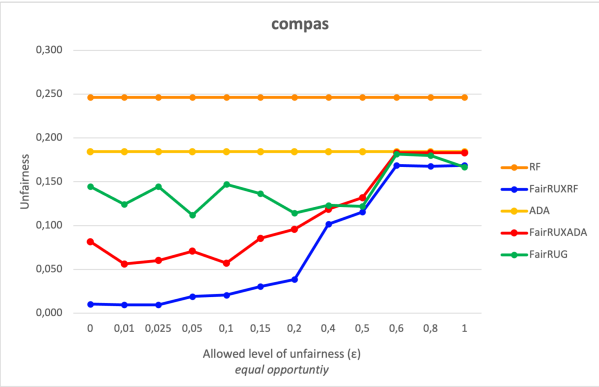
Parameter	Value
`max_depth` of tree	3
`n_estimators`	100
ϵ	$\{0,\ 0.01,\ 0.025,\ 0.05,\ 0.1,\ 0.15,\ 0.2,\ 0.4,\ 0.5,\ 0.6,\ 0.8,\ 1\}$

Next, we apply our FairRUG algorithm in 10-fold with the same values of epsilon as in Table 6. In Figure 4, we present the unfairness under different strictness of fairness for the binary COMPAS data set. Figure 5 shows the unfairness under different strictness of fairness for the multi-class student data set. The results for the other data sets can be found in the appendix B.

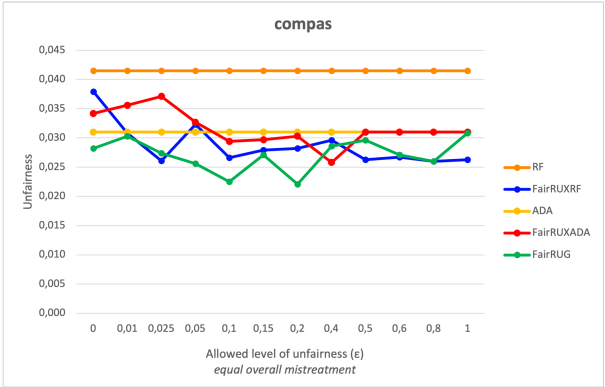
Equal Opportunity

Overall, we observe that FairRUX or FairRUG based on equal opportunity almost always results in a more fair classification compared to the normal RF or ADA. In the adult data set in Figure 9a of Appendix B, FairRUG only appears to be exceeding the level of unfairness of RF for $\epsilon \geq 0.2$. However, under stricter levels of unfairness, FairRUG still outperforms normal RF in terms of fairness. In one other case, the nursery data set, normal RF does perform better in terms of fairness than its fair version FairRUXRF and FairRUG, see Figure 11a of Appendix A. However, the minimum of our fair methods in the nursery data set is still on par with normal RF. Even so, in all other cases, FairRUX or FairRUG always proved to be fairer than normal RF or ADA.

Figure 4a, 5a and all figures of Appendix B with equal opportunity, generally show that increasing ϵ results in a higher measured unfairness and that our methods work best for low levels of ϵ. For example, Figure 4a demonstrates that the lowest amount of unfairness is reached for $\epsilon = 0.01$ for both RUXRF and RUXADA, resulting in unfairness of 1% and 5.6% respectively, but starts increasing when $\epsilon > 0.01$. Nevertheless, they still

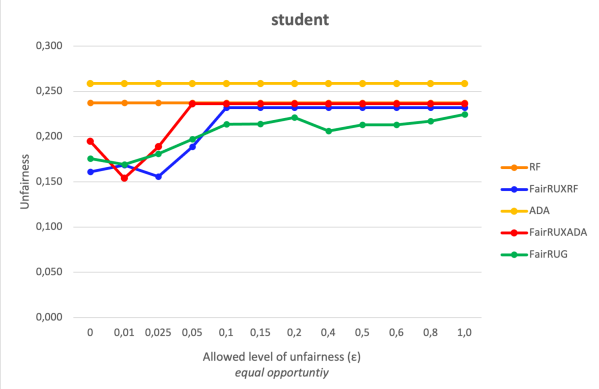


(a) COMPAS with equal opportunity

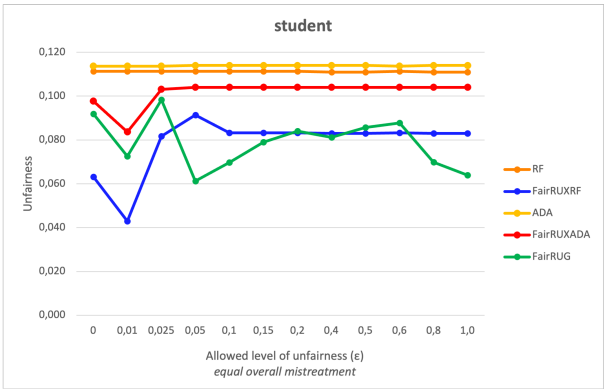


(b) COMPAS with equal overall mistreatment

Figure 4: Unfairness of all methods based on equal opportunity (left column) and equal overall mistreatment (right column), applied on the COMPAS data set.



(a) Student with equal opportunity



(b) Student with equal overall mistreatment

Figure 5: Unfairness of all methods based on equal opportunity (left column) and equal overall mistreatment (right column), applied on the student data set.

reach lower amounts of unfairness compared to the normal RF and ADA, which obtain a level of 24.6% and 18.4% respectively. It is also good to note that the FairRUX and FairRUG generally do not measure a higher level of unfairness when we relax the fairness constraints with $\epsilon = 1$ completely. In the default data set, it shows that the unfairness of the FairRUX and FairRUG does not improve when we increase ϵ, see Figure 10a of Appendix B. This implies that there is little to no unfairness in the default data set as the fairness constraints are easily met by the LP. Lawless et al. (2021) also confirms this in their findings. It is no surprise that the same can be concluded for the law data set, see Figure 12a of Appendix A, as a student's LSAT score and undergraduate GPA seem like trustworthy attributes for a student's expected GPA in law school. The absence of unfairness in the law data set is also supported by the findings of Denis et al. (2021). As our FairRUX and FairRUG achieve a lower amount of

fairness than RF and ADA for data sets where no unfairness is present, it is good to note that our methods do not seem to increase the unfairness in classification when there is no unfairness present in the data set. The actual numerical results of the complete experiment can be found in the tables of Appendix A.

Equal Overall Mistreatment

We observe that FairRUX and FairRUG achieve a lower level of unfairness than the normal RF and ADA in five of the six data sets, the exception being the nursery data set. In the nursery data set, FairRUXRF and FairRUXADA classify with a higher level of unfairness compared to RF and ADA respectively, see Figure 11. However, FairRUG seems to be on par with RF and ADA. It seems that it is easier to meet the fairness constraints based on equal overall mistreatment, as the accuracy and unfairness remain constant from lower levels of epsilon. This could be explained by the possibility that there is less unfairness present in the data sets when we measure by equal overall mistreatment.

Furthermore, for each method and each data set, we report the accuracy and unfairness of the method, the objective function value, the number of rules used in classification, and the average rule length, under their best parameter ϵ^*, see Tables 7 and 8. Note that the RF and ADA models are not dependent on ϵ, thus the unfairness and accuracy be constant for every ϵ. The number of rules and average rule length demonstrate the interpretability of the models. The length of a rule denotes the number of conditions in a rule. Hence, the fewer the rules and the shorter the rule, the more interpretable our prediction. Following Akyüz and Birbil (2021), for FairRUXRF and FairRUG, the length of a rule $j \in \mathcal{J}$ serves as the rule costs coefficient c_j of that rule. Hence, the shorter the rule, the lower the objective function value of the fair LP. For FairRUXADA, the rule costs coefficients are assigned as the inverse of the estimator weights of the trees in the trained ADA model (Akyüz and Birbil, 2021). Table 7 and 8 summarizes these results of the FairRUX and FairRUG algorithm with the optimal level of unfairness ϵ^* for each data set. For the results under all levels of ϵ, see the numerical values of the complete experiment Appendix A. All values computed are 10-fold means.

6.2.2 Fairness-Accuracy Trade-offs

With the implementation of fairness in the interpretable framework, we wish to establish whether our predictions keep a good accuracy. Figure 6 shows the accuracy under varying ϵ of FairRUX and FairRUG based on both metrics for the COMPAS data set, and Figure 7 for the student data set. The same figures can be found for all other data sets in Appendix C. Varying ϵ and performing 10-fold cross-validation allows us to generate the fairness-accuracy trade-offs.

We mainly observe that increasing ϵ, which produces a higher level of unfairness of our classifier, results in higher accuracy and lower objective function value. If there is unfairness in the data set present, this is a result we expect as we will decrease the unfairness in predictions, but our classifier will less often accurately predict the target value. The figures in Appendix C and Figure 6 confirm this is the case for the adult and COMPAS data set, with both metrics. For the other data sets, this also occurs, however, we also observe that the accuracy does not increase for the values of ϵ for which the unfairness also stops improving. For example, Figure 15 demonstrates that FairRUXRF and FairRUXADA keep a constant accuracy from $\epsilon = 0.1$ and $\epsilon = 0.05$ respectively. We can

Table 7: Results under optimal allowed level of unfairness ϵ^* subject to *equal opportunity*

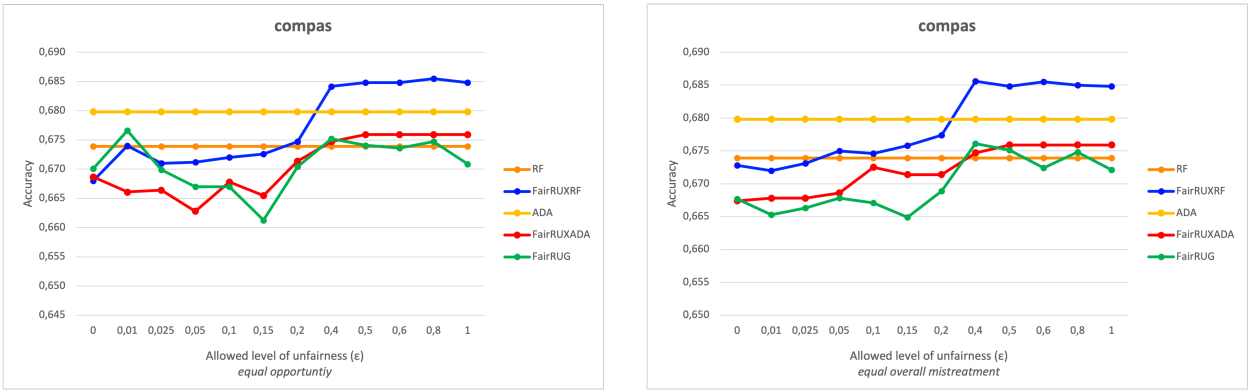
	$\epsilon*$		RF		FairRUXRF		ADA		FairRUXADA		FairRUG	
ADULT	0	Accuracy	0.837	(0.01)	0.848	(0.01)	0.858	(0.01)	0.854	(0.01)	0.853	(0.01)
		Unfairness α_B	0.048	(0.01)	**0.012**	(0.01)	0.080	(0.01)	0.036	(0.02)	0.025	(0.02)
		z	8085	(139)	8455	(61)	7858	(53)	8480	(49)	8299	(56)
		# rules	796	(1)	37	(10)	762	(19)	44	(7)	49	(4)
		avg. rule length	-	-	2.56	(0.10)	-	-	2.30	(0.09)	2.25	(0.16)
		CPU time (s)	0.747	(0.11)	50.70	(3.21)	2.117	(0.18)	40.718	(6.98)	108.9	(31.1)
COMPAS	0.01	Accuracy	0.674	(0.02)	0.674	(0.01)	0.680	(0.02)	0.666	(0.02)	0.677	(0.01)
		Unfairness α_B	0.246	(0.09)	**0.01**	(0.01)	0.184	(0.06)	0.056	(0.05)	0.124	(0.07)
		z	2921	(19.1)	3032	(5.3)	2994	(17.2)	3274	(56.3)	3529	(385)
		# rules	800	(0)	48	(12)	533	(5)	16	(8)	9	(7)
		avg. rule length	-	-	2.90	(0.01)	-	-	2.48	(0.28)	2.69	(0.15)
		CPU time (s)	0.132	(0)	4.165	(1.07)	0.177	(0.01)	2.671	(0.24)	0.539	(0)
DEFAULT	0.01	accuracy	0.813	(0.01)	0.821	(0.01)	0.820	(0.01)	0.819	(0.01)	0.820	(0.01)
		Unfairness α_B	0.015	(0.01)	0.012	(0.01)	0.015	(0.01)	**0.011**	(0.01)	**0.011**	(0.01)
		z	9483	(399.7)	9644	(46.94)	9547	(58.47)	9638	(41.86)	9592	(41.10)
		# rules	800	(0)	105	(95)	747	(27)	141	(59)	30	(4)
		avg. rule length	-	-	2.94	(0.03)	2.60	(0.17)	-	-	2.85	(0.15)
		CPU time (s)	0.976	(0.07)	157.0	(65.1)	6.920	(1.48)	87.08	(25.7)	154.9	(17.9)
STUDENT	0.01	Accuracy	0.672	(0.05)	0.595	(0.09)	0.635	(0.06)	0.567	(0.05)	0.681	(0.08)
		Unfairness α_{EO}	0.237	(0.05)	0.169	(0.06)	0.259	(0.12)	**0.154**	(0.07)	0.169	(0.09)
		z	26.37	(1.59)	27.71	(1.75)	19.13	(1.94)	19.18	1.91	130.32	27.52
		# rules	793	(3)	81	(7)	799	(2)	69	(7)	19	(7)
		avg. rule length	-	-	2.74	(0.03)	-	-	2.70	(0.05)	2.27	(0.12)
		CPU time (s)	0.092	(0)	0.986	(0.01)	0.125	(0)	0.873	(0)	0.451	(0.10)
NURSERY	0.01	Accuracy	0.765	(0.01)	0.687	(0.03)	0.761	(0.01)	0.670	(0.02)	0.758	(0.01)
		Unfairness α_{EO}	0.027	(0.01)	0.052	(0.02)	0.079	(0.03)	0.047	(0.03)	**0.027**	(0.01)
		z	467	(30.2)	475	(32.6)	427	(19.2)	428	(19.5)	3322	(255)
		# rules	701	(1)	64	(8)	740	(11)	78	(8)	11	(1)
		avg. rule length	-	-	2.69	(0.02)	-	-	2.69	(0.03)	2.25	(0.23)
		CPU time (s)	0.417	(0.13)	137.4	(4.02)	0.772	(0.15)	22.52	(3.15)	10.96	(2.62)
LAW	0.01	Accuracy	0.988	(0)	0.997	(0)	0.997	(0)	0.997	(0)	0.997	(0)
		Unfairness α_{EO}	0.135	(0.07)	**0**	(0)	0.046	(0.06)	**0**	(0)	**0**	(0)
		z	61.9	(6.19)	61.9	(6.19)	53.6	(3.83)	53.9	(4.29)	53.14	(3.82)
		# rules	614	(3)	4	(0)	680	(34)	7	(1)	6	(1)
		avg. rule length	-	-	1	(1)	-	-	1	(1)	1	(1)
		CPU time (s)	0.453	(0.16)	11.35	(2.91)	0.867	(0.29)	11.88	(3.31)	7.374	(2.32)

Table 8: Results under optimal allowed level of unfairness ϵ^* subject to *equal overall mistreatment*

	$\epsilon*$		RF		FairRUXRF		ADA		FairRUXADA		FairRUG	
ADULT	0	Accuracy	0.837	(0.01)	0.848	(0.01)	0.858	(0.01)	0.854	(0.01)	0.854	(0.01)
		Unfairness α_B	0.111	(0.01)	**0.100**	(0.02)	0.108	(0.01)	0.102	(0.01)	0.104	(0.01)
		z	8141.9	(292.6)	8454.9	(61.4)	7857.7	(53.2)	8480.4	(0.01)	8321.4	(61.2)
		# rules	796	(1)	37	(10)	762	(19)	44	(7)	41	(6)
		avg. rule length	-	-	2.712	(0.08)	-	-	2.140	(0.12)	2.27	(0.03)
		CPU time(s)	0.539	(0.04)	56.51	(5.24)	1.903	(0.07)	38.74	(2.99)	138.9	(46.8)
COMPAS	0.1	Accuracy	0.674	(0.02)	0.675	(0.02)	0.680	(0.02)	0.673	(0.02)	0.667	(0.02)
		Unfairness α_B	0.042	(0.02)	0.027	(0.02)	0.031	(0.02)	0.029	(0.03)	**0.023**	(0.02)
		z	2939.8	(18.2)	3051.9	(16.8)	3016.6	(26.1)	3252.8	(53.4)	3618.7	(338.2)
		# rules	800	(0)	32	(8)	535	(8)	16	(3)	12	(12.0)
		avg. rule length	-	-	2.897	(0.02)	-	-	2.295	(0.18)	2.588	(0.28)
		CPU time(s)	0.126	(0)	4.308	(0.87)	0.167	(0)	2.858	(0.28)	3.427	(4.87)
DEFAULT	0.15	Accuracy	0.809	(0.01)	0.821	(0.01)	0.820	(0.01)	0.820	(0.01)	0.820	(0.01)
		Unfairness α_B	0.026	(0.01)	0.026	(0.01)	0.025	(0.01)	0.025	(0.01)	**0.023**	(0.01)
		z	9555.8	(50.0)	9555.8	(50.0)	9545.2	(52.1)	9545.2	(52.1)	9557.8	(46.3)
		# rules	800	(0)	33	(14)	743	(27)	26	(4)	21	(4.0)
		avg. rule length	-	-	2.555	(0.09)	-	-	2.304	(0.09)	2.253	(0.15)
		CPU time(s)	0.962	(0)	35.48	(5.32)	6.092	(0.12)	30.17	(6.68)	119.18	(19.5)
STUDENT	0.01	Accuracy	0.661	(0.04)	0.592	(0.07)	0.664	(0.05)	0.590	(0.08)	0.676	(0.07)
		Unfairness α_{EOM}	0.111	(0.09)	**0.082**	(0.05)	0.114	(0.07)	0.103	(0.07)	0.098	(0.04)
		z	27.44	(2.51)	28.34	(2.58)	20.71	(4.33)	20.75	(4.44)	103.5	(29.8)
		# rules	794	(2)	85	(6)	798	(2)	76	(8)	21	(6.0)
		avg. rule length	-	-	2.748	(0.02)	-	-	2.579	(0.12)	2.319	(0.25)
		CPU time(s)	0.093	(0)	0.993	(0.06)	0.128	(0.01)	0.922	(0.16)	0.247	(0.04)
NURSERY	0.025	Accuracy	0.765	(0.01)	0.668	(0.03)	0.761	(0.01)	0.666	(0.02)	0.755	(0.02)
		Unfairness α_{EOM}	0.033	(0.02)	0.057	(0.03)	0.034	(0.02)	0.045	(0.02)	**0.027**	(0.01)
		z	466.9	(30.2)	466.9	(30.2)	426.9	(19.2)	426.9	(19.2)	3154.9	(200.4)
		# rules	701	(1)	61	(11)	740	(11)	78	(8)	15	(5.0)
		avg. rule length	-	-	2.698	(0.03)	-	-	2.689	(0.03)	2.433	(0.13)
		CPU time(s)	0.267	(0.03)	14.00	(3.41)	0.506	(0.05)	16.12	(3.11)	7.769	(2.78)
LAW	0.025	Accuracy	0.988	(0)	0.998	(0)	0.998	(0)	0.998	(0)	0.998	(0)
		Unfairness α_{EOM}	0.012	(0.01)	0.006	(0)	0.005	(0)	**0.004**	(0)	**0.004**	(0.)
		z	62.52	(4.5)	62.52	(4.5)	52.60	(4.49)	53.95	(3.95)	53.94	(3.62)
		# rules	613	(3)	4	(0)	659	(49)	7	(1)	7	(3)
		avg. rule length	-	-	1	(0)	-	-	1	(0)	1.22	(0)
		CPU time(s)	0.254	(0.02)	7.926	(0.27)	0.476	(0.01)	7.773	(19)	5.322	(2.23)

see from its numerical results in Table 14 of Appendix A that the level of unfairness, based on both metrics, is constant for $\epsilon \geq 0.1$ and $\epsilon \geq 0.05$ as well.

Further, we see that in most cases FairRUXRF or FairRUG have a higher predictive performance than normal RF. In comparison to ADA, FairRUG seems to be on par and in fact, even outperforms ADA on several data sets. This is the case for the adult data set with equal opportunity and the student data set, with both metrics. Generally, ADA achieves the highest accuracy but also produces the highest amount of unfairness in classification, whereas FairRUX and FairRUG maintain a low level of unfairness. Even so, the cases where FairRUX and FairRUG show lower predictive performance, still maintain good accuracy. When we look at the predictive performance under ϵ^*, we generally consider the settings where our fair methods perform their best in terms of fairness, but their worst in terms of accuracy. However, we see that the average gap between the predictive performances of FairRUXADA and ADA, over all data sets, is around 3% for equal opportunity and 2.9% for equal overall mistreatment. For FairRUG and ADA these gaps are 2.7% and 0.18% for equal opportunity and equal overall mistreatment respectively. Hence, we can conclude that FairRUX and FairRUG maintain good accuracy.



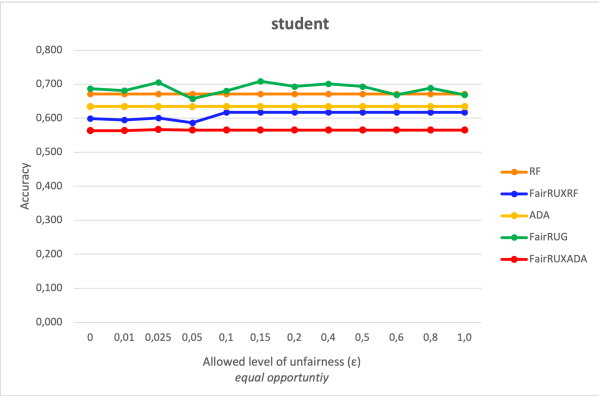
(a) COMPAS with equal opportunity

(b) COMPAS with equal overall mistreatment

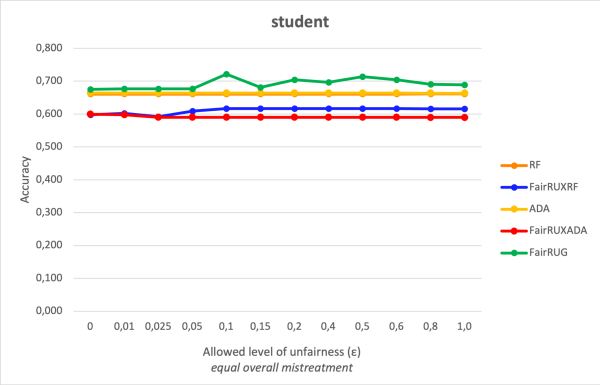
Figure 6: Accuracy of all methods based on equal opportunity (left column) and equal overall mistreatment (right column), applied on the COMPAS data set.

6.2.3 Fairness-Interpretability Trade-off

We have built our fair and interpretable classifier on the interpretable classification framework of Akyüz and Birbil (2021). With our emphasis on fairness, it is important to keep in mind that our classifier has to stay interpretable. Fortunately, Table 7 and Table 8 show us that this is still the case as implementing the fairness constraints significantly reduces the number of rules used in classification. Overall, FairRUG produced the least number of rules with the rules also having the shortest average length. Example 6.1 shows the complexity reduction of the FairRUG algorithm and demonstrates the simplicity of its outcome. Example 6.1 displays all the generated rules by FairRUG with $\epsilon = 0.01$. Each rule is assigned a weight and selects a class that is considered to be the predicted class label. This demonstrates that we are able to reduce a large number of rules, namely approximately 800 when



(a) Student with equal opportunity



(b) Student with equal overall mistreatment

Figure 7: Accuracy of all methods based on equal opportunity (left column) and equal overall mistreatment (right column), applied on the student data set.

generated with RF under the used parameters, to just five rules. From Table 7, we see that measured unfairness by normal RF is equal to 13.5%. However, with FairRUG, the measured unfairness is equal to 0%, and it is now far easier to interpret what rules were used to predict the chances of a student passing the bar. We see that rule 0, is a strict rule with weight 1, where students with the lowest GPA's are classified as having the highest probability to fail the bar exams. It makes sense that this prediction is mainly motivated by his or her undergraduate GPA and it is to be expected that ones gender or race has less influence on that outcome. If we were to regard `race` as the sensitive attribute, then an expected result is that there should not exist any unfairness in the data set. Accordingly, the absence of unfairness in the law data set can now also be explained by the generated rules that show classification is only based on the attribute undergraduate GPA.

Example 6.1. *Generated rules for classification on law data set, by FairRUG based on equal opportunity.*

Rule 0:	**if** $16.50 <$ `undergrad GPA` ≤ 18.50	**then** `class` $= 1$, `weight` $= 1.00$
Rule 1:	**if** `undergrad GPA` < 19.50	**then** `class` $= 3$, `weight` $= 0.75$
Rule 2:	**if** $19.50 <$ `undergrad GPA` ≤ 20.50	**then** `class` $= 4$, `weight` $= 0.75$
Rule 3:	**if** $20.50 <$ `undergrad GPA` ≤ 23.00	**then** `class` $= 3$, `weight` $= 0.75$
Rule 4:	**if** `undergrad GPA` ≤ 23	**then** `class` $= 4$, `weight` $= 0.75$

6.2.4 Comparison to Other Works

The work of Lawless et al. (2021) is able to achieve good results for fair classification in the binary setting. As mentioned earlier, they conducted their experiment on the data sets adult, COMPAS and default. This allows us to benchmark the performance of FairRUX and FairRUG to their fair classifier. Lawless et al. (2021) create a fair classifier that also utilizes column generation and so we refer to their classifier as FairCG.

According to Lawless et al. (2021), their classifier performs best for $\epsilon = 0.025$, and so Table 9 shows the results of FairCG and our fair methods for each data set for $\epsilon = 0.025$. We also include the values of ϵ^* for which our methods worked best, for each data set. Note that these results are also given in Table 7, but are added here as well for the comparison to FairCG. Other values of ϵ reported by Lawless et al. (2021) are the values $\epsilon = 0$ and $\epsilon = 1$. Hence, we used $\epsilon = 0$ to compare the results under strict fairness conditions and $\epsilon = 1$ to compare the results when we relax the fairness constraints completely. All predictive performance measures are based on the classifier's test accuracy.

Table 9: Comparison of FairCG, FairRUX and FairRUG[1]

	Unfairness				Accuracy (%)			
ADULT	FairCG	FairRUXRF	FairRUXADA	FairRUG	FairCG	FairRUXRF	FairRUXADA	FairRUG
$\epsilon = 0$	0.01	0.01	0.04	0.03	76.0	84.8	85.4	85.3
$\epsilon = 0.025$	0.01	0.01	0.04	0.03	81.7	85.0	85.4	85.5
$\epsilon = 1$	0.08	0.05	0.05	0.06	82.5	85.1	85.5	85.7
$\epsilon^* = 0$	0.01	0.01	0.04	0.03	76.0	84.8	85.4	85.3
COMPAS	FairCG	FairRUXRF	FairRUXADA	FairRUG	FairCG	FairRUXRF	FairRUXADA	FairRUG
$\epsilon = 0$	0.01	0.01	0.08	0.15	53.8	66.8	66.9	67.0
$\epsilon = 0.025$	0.01	0.01	0.06	0.14	64.5	67.1	66.6	67.7
$\epsilon = 1$	0.24	0.17	0.18	0.17	67.6	68.5	67.6	67.1
$\epsilon^* = 0.01$	-	0.01	0.06	0.12	-	67.4	66.6	67.7
DEFAULT	FairCG	FairRUXRF	FairRUXADA	FairRUG	FairCG	FairRUXRF	FairRUXADA	FairRUG
$\epsilon = 0$	0.00	0.02	0.01	0.01	77.8	82.0	81.9	81.7
$\epsilon = 0.025$	0.00	0.02	0.01	0.01	77.7	82.0	81.9	81.8
$\epsilon = 1$	0.01	0.02	0.01	0.01	82.0	82.0	81.9	82.0
$\epsilon^* = 0.01$	-	0.01	0.01	0.01	-	82.1	81.9	82.0

In terms of fairness, we observe that FairCG and our fair methods are quite evenly matched. Sometimes FairCG outperforms FairRUX or FairRUG, and other times it is the other way round. However, in terms of accuracy, all of our three methods, FairRUXRF, FairRUXADA, and FairRUG, outperform FairCG of Lawless et al. (2021). For the adult and COMPAS data set, we see that FairRUXRF is the best performing method in terms of fairness. For $\epsilon = 0$ and $\epsilon = 0.025$, FairCG achieves a lower amount of unfairness than FairRUXADA and FairRUG. However, when we relax the fairness constraints with $\epsilon = 1$, not only do FairRUX and FairRUG achieve lower unfairness, we also see that FairCG performs classifications with a higher amount of unfairness compared to normal RF and ADA. For the default data set, with an unfairness gap between FairCG and our fair methods of 1%, FairCG performs slightly better than FairRUX and FairRUG. A possible explanation could be that our fair classifiers are a bit more sensitive to the absence of unfairness in the default data set.

[1]The dashes in Table 9 signify that we were unable to obtain the results for these values of ϵ as Lawless et al. (2021) did not explicitly report them.

When we compare the predictive performance of the methods, we see that FairRUX and FairRUG outperform FairCG altogether. There are two occurrences where FairCG achieves higher accuracy, but taking into account the standard deviation, FairRUX or FairRUG might still perform better. One case where this occurs is in the COMPAS data set for $\epsilon = 1$, where the accuracy gap between FairCG and FairRUG is 0.5%, but the standard deviation of FairRUG and FairCG is 9% and 1.1% respectively. The other case occurs in the default data set for $\epsilon = 1$. The accuracy gap between FairCG and FairRUXADA is 1% with standard deviations 1% and 0.5% respectively.

Further, Lawless et al. (2021) compare the results of their FairCG to other methods, of which the fair classifier of Hardt et al. (2016) and the exponential gradient method included in the FairLearn Python package (Agarwal et al., 2018) were worthy competitors. We refer to these fair methods as Hardt and FairLearn respectively. Hardt and FairLearn outperform FairCG in terms of fairness and accuracy for $\epsilon = 1$, in the data sets adult and COMPAS. For this reason, we include Hardt and FairLearn in this comparison to other works, see Table 10. In all other cases, Lawless et al. (2021) show that FairCG outperforms Hardt and FairLearn.

Table 10: Comparison between FairCG, Hardt and FairLearn for $\epsilon = 1$

		FairCG	Hardt	FairLearn	FairRUXRF	FairRUXADA	FairRUG
ADULT	Accuracy $(\%)$	82.5	83.0	82.4	85.1	85.5	85.3
	Unfairness	0.08	0.18	0.12	0.05	0.05	0.06
COMPAS	Accuracy $(\%)$	67.6	65.9	65.8	68.5	67.6	67.1
	Unfairness	0.24	0.24	0.22	0.17	0.18	0.17

From Table 9 we see that FairRUX and FairRUG show better predictive performance compared to FairCG, but as Table 10 shows cases where FairCG is outperformed by other methods as well, we compare FairRUX and FairRUG to Hardt and FairLearn. We see that FairRUX and FairRUG is able to perform classifications with a smaller amount of unfairness and higher predictive performance compared to Hardt and FairLearn. Note that these were the only cases where FairCG was outperformed by Hardt or Fairlearn, thus there is no need to include fairness and accuracy results of these methods for other values of ϵ.

With our binary classification on par or even outperforming FairCG in two of the three data sets, and with higher predictive performance in general, we can conclude that FairRUX and FairRUG are worthy competitors to other works out there.

7 Conclusion

As the use of machine learning models in decision-making continues to grow, it is important that these models are trustworthy. With many accurate machine learning models available today, we cannot forget the importance of an interpretable and fair model. Sometimes, data that is used for training these models, contain discrimination towards a group of individuals, and this discrimination is then perpetuated by the machine learning models today to an unfair decision. This thesis aimed to develop a fair classification tool based on the interpretable classification framework of Akyüz and Birbil (2021), that remained applicable to multi-class classification problems, all the while maintaining a good accuracy.

We have presented a linear programming formulation that is subject to fairness constraints. The objective function of this linear program minimized the classification error of each data sample to maintain a good accuracy, and simultaneously minimized the number of decision rules used in classification, to achieve interpretability. To incorporate fairness, we imposed fairness as optimization constraints. We derived these fairness constraints from two fairness metrics, based on the separation definition of fairness: requiring *equal opportunity* and requiring *equal overall mistreatment*. As the fairness conditions hold the possibility of being unrealistic, a level of unfairness of $\epsilon \geq 0$ was allowed. The resulting linear program was used in the construction of two fair classification algorithms, Fair Rule Extraction (FairRUX) and Fair Rule Generation (FairRUG). FairRUX extracts the most important classification rules from trained Random Forests and Adaptive Boosting models, by solving the fair linear programs. On average, FairRUX was able to reduce the number of used rules in prediction by approximately 90%, making the predictions interpretable compared to the normal Random Forests and AdaBoost. In the FairRUG algorithm, rules are not extracted from existing methods, but the rules are generated via a column generation procedure. Hence, the FairRUG is a classifier itself. The column generation procedure utilizes the same fair linear program as FairRUX and iteratively added rules to its generated rule set by solving the pricing subproblem. FairRUG was able to generate rule sets that are approximately 97% less than the size of the decision rule sets of Random Forests and AdaBoost. Naturally, these rule sets have proven to be interpretable as well.

The fair algorithms were tested on six data sets, available from the UCI Machine Learning Repository or Kaggle. These data sets, adult, COMPAS, default, student, nursery, and law, contained binary and multi-class target values. Because the FairRUX and FairRUG algorithms are based on linear programming models, they are scalable for large data sets and they also work on continuous and categorical features. Hence, there was not much pre-processing or cleaning of the data needed. The only cleaning of data performed was to create experiments under the same circumstances as other works for benchmarking purposes.

The computational experiments have proven that a fair and interpretable classifier, with good test accuracy, is attainable in the multi-class setting. For each data set, there was one or more $\epsilon \geq 0$ that resulted in a classification of which the unfairness in ensemble methods was reduced by FairRUX or FairRUG. Imposing strict fairness, or relaxing fairness, resulted in effects that were to be expected. The stricter we are on the linear program in terms of fairness, the lower the level of unfairness in prediction. If there is any unfairness present in the data set, then this resulted in a lower accuracy. The accuracy and objective function values of the linear program share an inverse relation, and so the objective function value also increased, as expected.

The contribution of this thesis is two-fold. First, fairness and interpretability in the multi-class setting are attainable, while maintaining good accuracy. This means that groups of individuals that are subject to discrimination by machine learning models are closer to receiving a fair judgment, under our definitions of fairness. Second, we bench-marked our fair classification tool to other works in the binary setting and proved that FairRUX and FairRUG are on par with, or sometimes even outperform the other works in terms of fairness. We also compared the predictive performance of our tools to other works, and FairRUX and FairRUG always showed to achieve higher accuracy on the test data.

Finally, even though the emphasis of this thesis is placed on fairness, it is important to note that our classifier remains interpretable as well.

8 Discussion

Section 8.1 discusses any practical limitations encountered during this research. This research was also subject to theoretical limitations that are suggested as opportunities for future research in section 8.2.

8.1 Practical limitations

From the computational experiments, we have seen that the model works best for different values of ϵ. In practice, this means that we would have to test a range of ϵ before choosing the best one, and thus we need to perform hyperparameter tuning for every different data set. However, applying the 10-fold cross-validation on large data sets may take a long time. The 10-fold cross-validation of data sets which contained more than ten thousand samples, took about an hour, to an hour and a half. Hence, before this fair and interpretable classifier would be used by decision-makers, hyper-parameter tuning has to be performed for different data sets.

8.2 Further research

In this section, we describe opportunities for further research. They are either ideas we have not started on, or something that was attempted during this research, but leave for further research directions.

8.2.1 Comparison of Metrics

Figure 8 shows the percentages of unfairness in our three fair methods, subject to both metrics, for the student data set. We see a clear difference in the level of unfairness measured by both metrics, however, the image does not necessarily imply that the equal overall mistreatment constraints result in a fairer classification than predictions based on equal opportunity. As both unfairness levels α_{EO} and α_{EOM} are defined in different manners, see section 5.4, we cannot simply compare these values. Next to achieving fair classification in the multi-class setting, it would be insightful to know if there is one metric better than the other. Further research could contain the definition of a function that scales α_{EO} and α_{EOM} such that they are comparable.

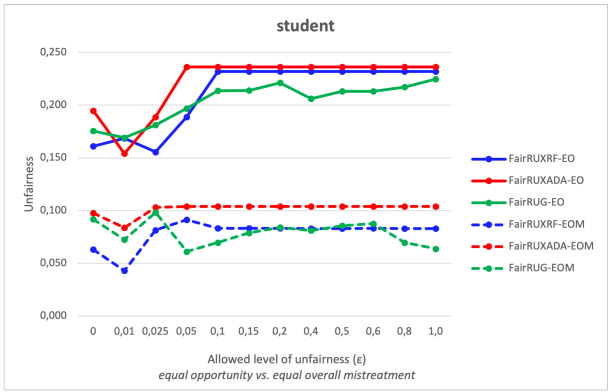


Figure 8: Unfairness for *equal opportunity* and *equal overall mistreatment*, applied on the student data set

8.2.2 Additional Metrics: Equalized Odds

During this research, ambition was to define the metric equalized odds in the multi-class setting. Equalized odds is a stricter version of equal opportunity, and in the binary setting, requires the true positive rate and false negative rate to be equal among groups. Translating this metric to a fairness condition posed no problem, however, we found difficulty in defining the fair linear programming constraints based on equalized odds. With the same assumption as for equal opportunity, it follows that equalized odds is equivalent to a one-vs-all binary classification where we require an equal misclassification rate and equal *correct* classification rate, for each class k, among all groups $g, g' \in \mathcal{G}$.
Recall that requiring an equal misclassification rate for each class, among all group, is the same as equal opportunity. For this condition, we refer to condition (17) and (18). For an equal correct classification rate for each class among all groups, we wish that $\forall k \in \mathcal{K}, \forall g, g' \in \mathcal{G}$:

$$P(\hat{y} = y | y = k, G = g) = P(\hat{y} = y | y = k, G = g') \tag{74}$$

Or with a with an allowed level of unfairness of $\epsilon \geq 0$:

$$|P(\hat{y} = y | y = k, G = g) - P(\hat{y} = y | y = k, G = g')| \leq \epsilon \tag{75}$$

Where we introduced variable v_i for each i to describe the *misclassification* error of sample i to define LP-constraints for conditions (17) and (18), an idea would be to define a variable u_i for each i to describe the extent of *correct* classification of a sample i. However, we found that defining such a u_i proved to be difficult. We ran into some problems such as one definition of $u_i \geq 0$ resulted in the non-convex optimization problem. Another approach defined $u_i \in \{0, 1\}$, however this would change the fair LP to a Mixed Integer Linear Program, which is a lot harder to solve. Further research could entail the definition of this u_i and perhaps to continue with a heuristic.

8.2.3 Extension to Regression Problems

This research is conducted for multi-class classification. However, we can utilize ensemble methods in regression problems as well. This requires a reformulation of the LP, with, for example, the mean absolute deviation as its loss function in the objective. The resulting dual problem and pricing subproblem will also differ. Extension to regression problems would increase the applicability of our fair classifier.

References

Adnan, N. and Islam, Z. (2017). ForEx++: A New Framework for Knowledge Discovery from Decision Forests. *Australasian Journal of Information Systems, https://doi.org/10.3127/ajis.v21i0.1539*, 21.

Agarwal, A., Beygelzimer, A., Dudik, M., Langford, J., and Wallach, H. (2018). A Reductions Approach to Fair Classification. *arXiv preprint arXiv:1803.02453*.

Akyüz, H. and Birbil, I. (2021). Discovering Classification Rules for Interpretable Learning with Linear Programming. *arXiv preprint arXiv:2104.10751*.

Angwin, J., Larson, J., Mattu, S., and Kirchner, L. (2016). Machine Bias: There's Software Used Across The Country To Predict The Future Criminals, And I'ts Biased Against Blacks. *ProPublica, https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing*.

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

Bertsimas, D. and Dunn, J. (2017). Optimal Classification Trees. *Machine Learning*, 106(7):1039–1082.

Bertsimas, D., O'Hair, A., Relyea, S., and Silberholz, J. (2016). An Analytics Approach to Designing Combination Chemotherapy Regimens for Cancer. *Management Science*, 62(5):1511–1531.

Biau, G. and Scornet, E. (2016). A Random Forest Guided Tour. *TEST*, 25:197–227.

Bien, J. and Tibshirani, R. (2009). Classification by Set Cover: The Prototype Vector Machine. *arXiv preprint arXiv:0908.2284*.

Birbil, S. I., Edali, M., and Yuceoglu, B. (2020). Rule Covering for Interpretation and Boosting. *arXiv preprint arXiv:2007.06379*.

Blanquero, R., Carrizosa, E., and abd Dolores Romero Morales, C. M.-R. (2016). Optimal Randomized Classification Trees. *Computers and Operations Research*, 132.

Bringmann, B. and Zimmermann, A. (2009). One in a Million: Picking the Right Patterns. *Knowledge and Information Systems*, 18:61–81.

Carrizosa, E., Molero-Río, C., and Morales, D. R. (2021). Mathematical Optimization in Classification and Regression Trees. *TOP*, 29:5–33.

Chen, T. and Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 785–794.

Chouldechova, A. (2016). Fair Prediction with Disparate Impact: A Study of Bias in Recidivism Prediction Instruments. *Proc. 3rd FATML*.

Corbett-Davies, S. and Goel, S. (2018). The Measure and Mismeasure of Fairness: A Critical Review of Fair Machine Learning. *arXiv preprint arXiv:1808.00023*.

Cortez, P. (2014). Student Performance Data Set. *Available from the UCI Machine Learning Repository: https://archive.ics.uci.edu/ml/datasets/student+performance*.

Cortez, P. and Silva, A. (2008). Using Data Mining to Predict Secondary School Student Performance. *Proceedings of 5th FUture BUsiness TECHnology Conference*, pages 5–12.

de Ensino Superior, D.-G. (2018). Conversion of the Final Grade to the Portuguese Scale. *Available from the DGES: https://www.dges.gov.pt/en/pagina/conversion-final-classification-portuguese-scale*.

Denis, C., Elie, R., Hebiri, M., and Hu, F. (2021). Fairness Guarantee in Multi-Class Classification. *arXiv preprint arXiv: 2109.13642*.

Desaulniers, G., Desrosiers, J., and Solomon, M. M. (2005). *Column Generation*. Springer. https://link.springer.com/book/10.1007/b135457.

Du, M., Yang, F., Zhou, N., and Hu, X. (2021). Fairness in Deep Learning: A Computational Perspective. *IEEE Intelligent Systems*, 36:25–34.

Dua, D. and Graff, C. (2017). UCI machine learning repository. http://archive.ics.uci.edu/ml.

Dwork, C., Hardt, M., Pitassi, T., Reingold, O., and Zemel, R. (2012). Fairness Through Awareness. *Proc. ACM ITCS*, pages 214–226.

Feldman, M., Friedler, S. A., Moeller, J., Scheidegger, C., and Venkatasubramanian, S. (2015). Certifying and Removing Disparate Impact. *Proc. ACM SIGKDD*, pages 259–268.

Freund, Y. and Schapire, R. E. (1997). A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting. *Journal of Computer and System Sciences*, 55(1):119–139.

Friedman, J. H. (2001). Greedy Function Approximation: A Gradient Boosting Machine. *The Annals of Statistics*, 29(5):1189–1232.

Gerlings, J., Shollo, A., and Constantiou, I. (2021). Reviewing the need for Explainable Artifical Intelligence (xAI). *Proceedings of the 54th Hawaii International Conference on System Sciences*.

González, S., García, S., Ser, J. D., Rokach, L., and Herrera, F. (2020). A Practical Tutorial on Bagging and Boosting Based Ensembles for Machine Learning: Algorithms, Software Tools, Performance Study, Practical Perspectives and Opportunities. *Information Fusion*, 64:205–237.

Hardt, M., Price, E., and Srebro, N. (2016). Equality of Opportunity in Supervised Learning. *NIPS*.

Kleinberg, J., Mullainathan, S., and Raghavan, M. (2017). Inherent Trade-Offs in the Fair Determination of Risk Scores. *Proceedings of Innovations in Theoretical Computer Science (ITCS)*.

Kohavi, R. and Becker, B. (1996). Adult Data Set. Available from the UCI Machine Learning Repository: https://archive.ics.uci.edu/ml/datasets/adult.

Lakkaraju, H., Bach, S., and Leskovec, J. (2016). Interpretable Decision Sets: A Joint Framework for Description and Prediction. *Proc. ACM SIGKDD*.

Larson, J., Mattu, S., Kirchner, L., and Angwin, J. (2016). How We Analyzed the COMPAS Recidivism Algorithm. *Journal of Computer and System Sciences*.

Lawless, C., Dash, S., Günlük, O., and Wei, D. (2021). Interpretable and Fair Boolean Rule Sets via Column Generation. *arXiv preprint arXiv: 2111.08466*.

Lee, J. K., Bu, Y., Raja, D., Sattigeri, P., Das, R. P. S., and Wornell, G. W. (2021). Fair Selective Classification Via Sufficiency. *Proceedings of the 38th International Conference on Machine Learning*, 139:6076–6086.

Liu, S., Patel, R. Y., Daga, P. R., Liu, H., Fu, G., Doerksen, R. J., Chen, Y., and Wilkins, D. E. (2012). Combined Rule Extraction and Feature Elimination in Supervised Classification. *IEEE Transactions on Nanobioscience*, 11(3):228–236.

Madhavan, R. and Wadhwa, M. (2020). Fairness-Aware Learning with Prejudice Free Representations. *Proc. ACM ICIKM*, pages 2137–2140.

Molnar, C., Casalicchio, G., and Bischl, B. (2020). Interpretable Machine Learning - A Brief History, State-of-the-Art and Challenges. *arXiv preprint arXiv:2010.09337*.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Ravichandran, S., Venkatesh, B., Khurana, D., and Edakunni, N. U. (2020). FairXGBoost: Fairness-aware Classification in XGBoost. *Proc. IEEEE/ACM ASE*, pages 883–894.

Rivest, R. L. (1987). Learning Decision Lists. *Machine Learning*, 2:229 246.

Roman, Y., Bates, S., and Candès, E. J. (2020). Achieving Equalized Odds by Resampling Sensitive Attributes. *Advances in Neural Information Processing Systems: Annual Conference on Neural Information Processing Systems*, 33.

Rosasco, L., Vito, E. D., Coaponnetto, A., Piana, M., and Verri, A. (2003). Are Loss Functions All the Same. *Neural Computation*, 16(5):1063–1076.

Schielzeth, H. (2010). Simple Means to Improve the Interpretability of Regression Coefficients. *Methods in Ecology and Evolution*, 1:103–113.

Shickel, B. and Rashidi, P. (2020). Sequential Interpretability: Methods, Applications, and Future Direction for Understanding Deep Learning Models in the Context of Sequential Data. *arXiv preprint arXiv:2004.12524*.

Simonyan, K., Vedaldi, A., and Zisserman, A. (2014). Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps. *Computing Research Repository*, abs/1213.6034.

Snellen, I. T. M. and de Donk, W. B. H. J. V. (1989). *Expert Systems in Public Administration: Evolving Practices and Norms - An Application for Admission in Publiv School Systems*. Elsevier Science Ltd. https://archive.ics.uci.edu/ml/datasets/nursery.

Tavakol, M. (2020). Fair Classification with Counterfactual Learning. *Proceedings of the 43rd International ACM SIGIR Conference on Research andDevelopment in Information Retrieval*, available from: https://doi.org/10.1145/3397271.3401291.

Tibshirani, R. (1996). Regression Shrinkage and Selection via the Lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, 58:267–288.

Torres, J. M., comesaña, C. I., and García-Nieto, P. J. (2019). Machine Learning Techniques Applied to Cybersecurity. *International Journal of Machine Learning and Cybernetics*, 10:2823–2836.

Wanga, S., Wanga, Y., Wang, D., Yin, Y., Wanga, Y., and Jin, Y. (2020). An Improved Random Forest-Based Rule Extraction Method for Breast Cancer Diagnosis. *Applied Soft Computing Journal*, 86:105941.

Wightman, L. F. (1998). LSAC National Longitudinal Bar Passage Study. *Law School Admission Council - Reports - Research(143). Data set available from: http://www.seaphe.org/databases.php*.

Woodworth, B., Suriya Gunasekar, M. I. O., and Srebro, N. (2017). Learning Non-Discriminatory Predictors. *Proc. of Machine Learning Research*, 65:1–34.

Yeh, I.-C. (2016). Default Data Set. *Available from the UCI Machine Learning Repository: https://archive.ics.uci.edu/ml/datasets/default+of+credit+card+clients*.

Zafar, M. B., Valera, I., Gomez-Rodriguez, M., and Gummadi, K. P. (2019). Fairness Constraints: A Flexible Approach for Fair Classification. *Journal of Machine Learning Research*, 20:1–42.

Zeng, J., Ustun, B., and Rudin, C. (2017). Interpretable Classification Models for Recidivism Prediction. *Journal of Royal Statistics*, Series A 180(3):689–722.

9 Appendices

A Results: Numerical Experiments for all ϵ

A.1 Data set: Adult

Table 11: Accuracy and measured unfairness based on both metrics for the adult data set

	ϵ	**0**	**0.01**	**0.025**	**0.05**	**0.1**	**0.15**	**0.2**	**0.4**	**0.5**	**0.6**	**0.8**	**1.0**
	Accuracy	0.837	0.837	0.837	0.837	0.837	0.837	0.837	0.837	0.837	0.837	0.837	0.837
RF	Equal Opportunity	0.048	0.048	0.048	0.048	0.048	0.048	0.048	0.048	0.048	0.048	0.048	0.048
	Equal Overall Mistr.	0.111	0.111	0.111	0.111	0.111	0.111	0.111	0.111	0.111	0.111	0.111	0.111
	Accuracy	0.858	0.858	0.858	0.858	0.858	0.858	0.858	0.858	0.858	0.858	0.858	0.858
ADA	Equal Opportunity	0.080	0.080	0.080	0.080	0.080	0.080	0.080	0.080	0.080	0.080	0.080	0.080
	Equal Overall Mistr.	0.108	0.108	0.108	0.108	0.108	0.108	0.108	0.108	0.108	0.108	0.108	0.108
	Accuracy	0.848	0.848	0.850	0.851	0.850	0.851	0.851	0.851	0.851	0.851	0.851	0.851
Fair	Equal Opportunity	0.012	0.013	0.014	0.029	0.040	0.043	0.044	0.045	0.045	0.045	0.045	0.045
RUXRF	Accuracy	0.848	0.849	0.850	0.851	0.851	0.851	0.851	0.851	0.851	0.851	0.851	0.851
	Equal Overall Mistr.	0.100	0.102	0.107	0.107	0.107	0.108	0.108	0.108	0.108	0.108	0.108	0.108
	Accuracy	0.854	0.854	0.854	0.854	0.855	0.855	0.855	0.855	0.855	0.855	0.855	0.855
Fair	Equal Opportunity	0.036	0.031	0.031	0.035	0.051	0.053	0.053	0.053	0.053	0.053	0.053	0.053
RUXADA	Accuracy	0.854	0.854	0.854	0.854	0.855	0.854	0.855	0.855	0.855	0.855	0.855	0.855
	Equal Overall Mistr.	0.102	0.102	0.102	0.104	0.107	0.108	0.108	0.108	0.108	0.108	0.108	0.108
	Accuracy	0,853	0,855	0,855	0,855	0,854	0,855	0,855	0,855	0,856	0,855	0,857	0,857
Fair	Equal Opportunity	0,025	0,031	0,026	0,036	0,043	0,046	0,048	0,049	0,053	0,051	0,054	0,055
RUG	Accuracy	0,854	0,854	0,855	0,855	0,855	0,854	0,855	0,856	0,855	0,856	0,855	0,856
	Equal Overall Mistr.	0,104	0,103	0,105	0,104	0,108	0,108	0,108	0,108	0,108	0,108	0,109	0,107

A.2 Data set: COMPAS

Table 12: Accuracy and measured unfairness based on both metrics for the COMPAS data set

	ϵ	**0**	**0.01**	**0.025**	**0.05**	**0.1**	**0.15**	**0.2**	**0.4**	**0.5**	**0.6**	**0.8**	**1.0**
	Accuracy	0.674	0.674	0.674	0.674	0.674	0.674	0.674	0.674	0.674	0.674	0.674	0.674
RF	Equal Opportunity	0.246	0.246	0.246	0.246	0.246	0.246	0.246	0.246	0.246	0.246	0.246	0.246
	Equal Overall Mistr.	0.042	0.042	0.042	0.042	0.042	0.042	0.042	0.042	0.042	0.042	0.042	0.042
	Accuracy	0.680	0.680	0.680	0.680	0.680	0.680	0.680	0.680	0.680	0.680	0.680	0.680
ADA	Equal Opportunity	0.184	0.184	0.184	0.184	0.184	0.184	0.184	0.184	0.184	0.184	0.184	0.184
	Equal Overall Mistr.	0.031	0.031	0.031	0.031	0.031	0.031	0.031	0.031	0.031	0.031	0.031	0.031
	Accuracy	0.668	0.674	0.671	0.671	0.672	0.673	0.675	0.684	0.685	0.685	0.686	0.685
Fair	Equal Opportunity	0.010	0.010	0.010	0.019	0.021	0.031	0.039	0.102	0.115	0.169	0.168	0.169
RUXRF	Accuracy	0.673	0.672	0.673	0.675	0.675	0.676	0.677	0.686	0.685	0.686	0.685	0.685
	Equal Overall Mistr.	0.038	0.031	0.026	0.032	0.027	0.028	0.028	0.030	0.026	0.027	0.026	0.026
	Accuracy	0.669	0.666	0.666	0.663	0.668	0.666	0.671	0.675	0.676	0.676	0.676	0.676
Fair	Equal Opportunity	0.082	0.056	0.060	0.071	0.057	0.086	0.096	0.119	0.132	0.183	0.183	0.183
RUXADA	Accuracy	0.667	0.668	0.668	0.669	0.673	0.671	0.671	0.675	0.676	0.676	0.676	0.676
	Equal Overall Mistr.	0.034	0.036	0.037	0.033	0.029	0.030	0.030	0.026	0.031	0.031	0.031	0.031
	Accuracy	0.670	0.677	0.670	0.667	0.667	0.661	0.670	0.675	0.674	0.674	0.675	0.671
Fair	Equal Opportunity	0.145	0.124	0.144	0.112	0.147	0.137	0.114	0.123	0.122	0.182	0.180	0.167
RUG	Accuracy	0.668	0.665	0.666	0.668	0.667	0.665	0.669	0.676	0.675	0.672	0.675	0.672
	Equal Overall Mistr.	0.028	0.030	0.027	0.026	0.023	0.027	0.022	0.029	0.030	0.027	0.026	0.031

A.3 Data set: Default

Table 13: Accuracy and measured unfairness based on both metrics for the default data set

	ϵ	0	0.01	0.025	0.05	0.1	0.15	0.2	0.4	0.5	0.6	0.8	1.0
RF	Accuracy	0.813	0.813	0.813	0.813	0.813	0.813	0.813	0.813	0.813	0.813	0.813	0.813
	Equal Opportunity	0.015	0.015	0.015	0.015	0.015	0.015	0.015	0.015	0.015	0.015	0.015	0.015
	Equal Overall Mistr.	0.026	0.026	0.026	0.026	0.026	0.026	0.026	0.026	0.026	0.026	0.026	0.026
ADA	Accuracy	0.819	0.821	0.821	0.821	0.821	0.821	0.821	0.821	0.821	0.821	0.821	0.821
	Equal Opportunity	0.015	0.012	0.015	0.014	0.014	0.014	0.014	0.014	0.014	0.014	0.014	0.015
	Equal Overall Mistr.	0.025	0.025	0.025	0.025	0.025	0.025	0.025	0.025	0.025	0.025	0.025	0.025
Fair RUXRF	Accuracy	0.820	0.820	0.820	0.820	0.820	0.820	0.820	0.820	0.820	0.820	0.820	0.820
	Equal Opportunity	0.015	0.015	0.015	0.015	0.015	0.015	0.015	0.015	0.015	0.015	0.015	0.015
	Accuracy	0.819	0.819	0.820	0.820	0.820	0.820	0.820	0.820	0.820	0.820	0.820	0.820
	Equal Overall Mistr.	0.026	0.025	0.025	0.025	0.025	0.025	0.025	0.025	0.025	0.025	0.025	0.025
Fair RUXADA	Accuracy	0.819	0.819	0.820	0.820	0.820	0.820	0.820	0.820	0.820	0.820	0.820	0.819
	Equal Opportunity	0.011	0.011	0.011	0.012	0.012	0.012	0.012	0.012	0.012	0.012	0.012	0.012
	Accuracy	0.819	0.819	0.820	0.820	0.820	0.820	0.820	0.820	0.820	0.820	0.820	0.820
	Equal Overall Mistr.	0.026	0.025	0.025	0.025	0.025	0.025	0.025	0.025	0.025	0.025	0.025	0.025
Fair RUG	Accuracy	0.817	0.818	0.820	0.820	0.820	0.820	0.820	0.820	0.820	0.819	0.820	0.820
	Equal Opportunity	0.012	0.011	0.013	0.013	0.012	0.013	0.011	0.012	0.013	0.012	0.012	0.013
	Accuracy	0.816	0.817	0.820	0.820	0.820	0.820	0.819	0.819	0.820	0.820	0.820	0.820
	Equal Overall Mistr.	0.026	0.025	0.026	0.026	0.025	0.023	0.026	0.025	0.026	0.025	0.025	0.026

A.4 Data set: Student

Table 14: Accuracy and measured unfairness based on both metrics for the student data set

	ϵ	0	0.01	0.025	0.05	0.1	0.15	0.2	0.4	0.5	0.6	0.8	1.0
RF	Accuracy	0.672	0.672	0.672	0.672	0.672	0.672	0.672	0.672	0.672	0.672	0.672	0.672
	Equal Opportunity	0.237	0.237	0.237	0.237	0.237	0.237	0.237	0.237	0.237	0.237	0.237	0.237
	Equal Overall Mistr.	0.111	0.111	0.111	0.111	0.111	0.111	0.111	0.111	0.111	0.111	0.111	0.111
ADA	Accuracy	0.635	0.635	0.635	0.635	0.635	0.635	0.635	0.635	0.635	0.635	0.635	0.635
	Equal Opportunity	0.259	0.259	0.259	0.259	0.259	0.259	0.259	0.259	0.259	0.259	0.259	0.259
	Equal Overall Mistr.	0.114	0.114	0.114	0.114	0.114	0.114	0.114	0.114	0.114	0.114	0.114	0.114
Fair RUXRF	Accuracy	0.599	0.595	0.601	0.587	0.618	0.618	0.618	0.618	0.618	0.618	0.618	0.618
	Equal Opportunity	0.161	0.169	0.156	0.189	0.232	0.232	0.232	0.232	0.232	0.232	0.232	0.232
	Accuracy	0.598	0.602	0.592	0.609	0.616	0.616	0.616	0.616	0.616	0.616	0.616	0.616
	Equal Overall Mistr.	0.063	0.043	0.082	0.091	0.083	0.083	0.083	0.083	0.083	0.083	0.083	0.083
Fair RUXADA	Accuracy	0.564	0.564	0.567	0.566	0.566	0.566	0.566	0.566	0.566	0.566	0.566	0.566
	Equal Opportunity	0.195	0.154	0.189	0.236	0.236	0.236	0.236	0.236	0.236	0.236	0.236	0.236
	Accuracy	0.600	0.598	0.590	0.590	0.590	0.590	0.590	0.590	0.590	0.590	0.590	0.590
	Equal Overall Mistr.	0.098	0.084	0.103	0.104	0.104	0.104	0.104	0.104	0.104	0.104	0.104	0.104
Fair RUG	Accuracy	0.687	0.681	0.706	0.658	0.681	0.709	0.693	0.701	0.693	0.669	0.689	0.669
	Equal Opportunity	0.176	0.169	0.181	0.197	0.214	0.214	0.221	0.206	0.213	0.213	0.217	0.225
	Accuracy	0.675	0.676	0.676	0.677	0.721	0.681	0.704	0.697	0.714	0.704	0.690	0.689
	Equal Overall Mistr.	0.092	0.073	0.098	0.061	0.070	0.079	0.084	0.081	0.086	0.088	0.070	0.064

A.5 Data set: Nursery

Table 15: Accuracy and measured unfairness based on both metrics for the nursery data set

	ϵ	0	0.01	0.025	0.05	0.1	0.15	0.2	0.4	0.5	0.6	0.8	1.0
	Accuracy	0.765	0.765	0.765	0.765	0.765	0.765	0.765	0.765	0.765	0.765	0.765	0.765
RF	Equal Opportunity	0.027	0.027	0.027	0.027	0.027	0.027	0.027	0.027	0.027	0.027	0.027	0.027
	Equal Overall Mistr.	0.033	0.033	0.033	0.033	0.033	0.033	0.033	0.033	0.033	0.033	0.033	0.033
	Accuracy	0.761	0.761	0.761	0.761	0.761	0.761	0.761	0.761	0.761	0.761	0.761	0.761
ADA	Equal Opportunity	0.079	0.079	0.079	0.079	0.079	0.079	0.079	0.079	0.079	0.079	0.079	0.079
	Equal Overall Mistr.	0.034	0.034	0.034	0.034	0.034	0.034	0.034	0.034	0.034	0.034	0.034	0.034
	Accuracy	0.687	0.692	0.687	0.688	0.693	0.685	0.688	0.688	0.688	0.688	0.688	0.688
Fair	Equal Opportunity	0.045	0.052	0.049	0.066	0.063	0.067	0.066	0.067	0.067	0.067	0.067	0.067
RUXRF	Accuracy	0.687	0.688	0.688	0.688	0.688	0.688	0.688	0.688	0.688	0.688	0.688	0.688
	Equal Overall Mistr.	0.065	0.057	0.057	0.057	0.057	0.057	0.057	0.057	0.057	0.057	0.057	0.057
	Accuracy	0.670	0.668	0.667	0.666	0.671	0.667	0.666	0.666	0.666	0.666	0.666	0.666
Fair	Equal Opportunity	0.062	0.047	0.052	0.060	0.051	0.060	0.060	0.060	0.060	0.060	0.060	0.060
RUXADA	Accuracy	0.670	0.666	0.666	0.666	0.666	0.666	0.666	0.666	0.666	0.666	0.666	0.666
	Equal Overall Mistr.	0.047	0.045	0.045	0.045	0.045	0.045	0.045	0.045	0.045	0.045	0.045	0.045
	Accuracy	0.758	0.765	0.761	0.757	0.760	0.759	0.763	0.762	0.762	0.760	0.762	0.762
Fair	Equal Opportunity	0.030	0.027	0.035	0.037	0.038	0.035	0.038	0.036	0.040	0.044	0.042	0.046
RUG	Accuracy	0.757	0.754	0.755	0.757	0.754	0.761	0.762	0.762	0.762	0.762	0.761	0.760
	Equal Overall Mistr.	0.032	0.035	0.027	0.033	0.033	0.032	0.030	0.034	0.033	0.036	0.035	0.038

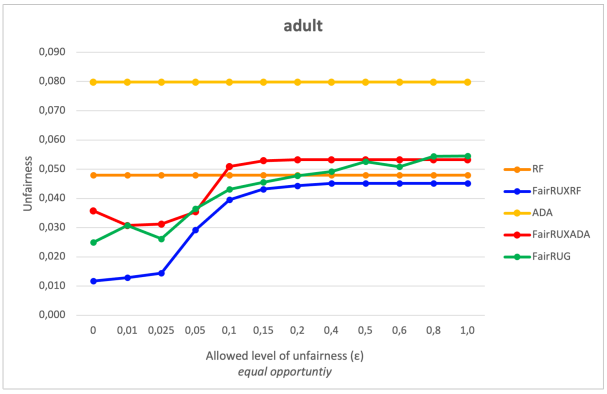
A.6 Data set: Law

Table 16: Accuracy and measured unfairness based on both metrics for the law data set

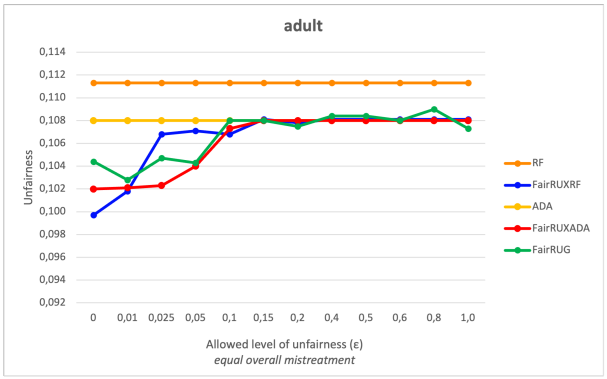
	ϵ	0	0,01	0,025	0,05	0,1	0,15	0,2	0,4	0,5	0,6	0,8	1,0
	Accuracy	0,988	0,988	0,988	0,988	0,988	0,988	0,988	0,988	0,988	0,988	0,988	0,988
RF	Equal Opportunity	0,135	0,135	0,135	0,135	0,135	0,135	0,135	0,135	0,135	0,135	0,135	0,135
	Equal Overall Mistr.	0,012	0,012	0,012	0,012	0,012	0,012	0,012	0,012	0,012	0,012	0,012	0,012
	Accuracy	0,997	0,997	0,997	0,997	0,997	0,997	0,997	0,997	0,997	0,997	0,997	0,997
ADA	Equal Opportunity	0,046	0,046	0,046	0,046	0,046	0,046	0,046	0,046	0,046	0,046	0,046	0,046
	Equal Overall Mistr.	0,005	0,005	0,005	0,005	0,005	0,005	0,005	0,005	0,005	0,005	0,005	0,005
	Accuracy	0,997	0,997	0,997	0,997	0,997	0,997	0,997	0,997	0,997	0,997	0,997	0,997
Fair	Equal Opportunity	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000
RUXRF	Accuracy	0,998	0,998	0,998	0,998	0,998	0,998	0,998	0,998	0,998	0,998	0,998	0,998
	Equal Overall Mistr.	0,006	0,006	0,006	0,006	0,006	0,006	0,006	0,006	0,006	0,006	0,006	0,006
	Accuracy	0,997	0,997	0,997	0,997	0,997	0,997	0,997	0,997	0,997	0,997	0,997	0,997
Fair	Equal Opportunity	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000
RUXADA	Accuracy	0,998	0,998	0,998	0,998	0,998	0,998	0,998	0,998	0,998	0,998	0,998	0,998
	Equal Overall Mistr.	0,004	0,004	0,004	0,004	0,004	0,004	0,004	0,004	0,004	0,004	0,004	0,004
	Accuracy	0,997	0,997	0,997	0,997	0,997	0,997	0,997	0,997	0,997	0,997	0,997	0,997
Fair	Equal Opportunity	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000
RUG	Accuracy	0,998	0,998	0,998	0,998	0,998	0,998	0,998	0,998	0,998	0,998	0,998	0,998
	Equal Overall Mistr.	0,004	0,004	0,004	0,004	0,004	0,004	0,004	0,004	0,004	0,004	0,004	0,004

B Results: Unfairness Under Varying ϵ

B.1 Data set: Adult



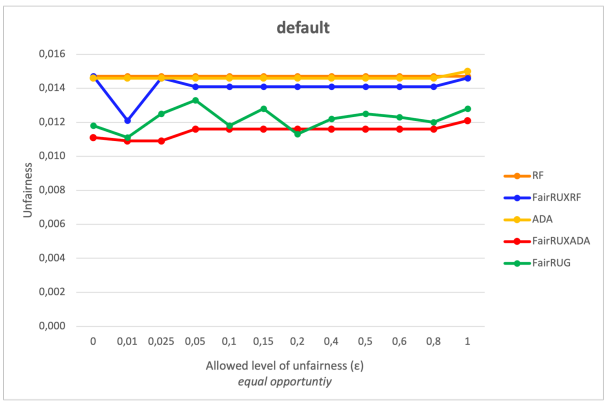
(a) Adult with equal opportunity



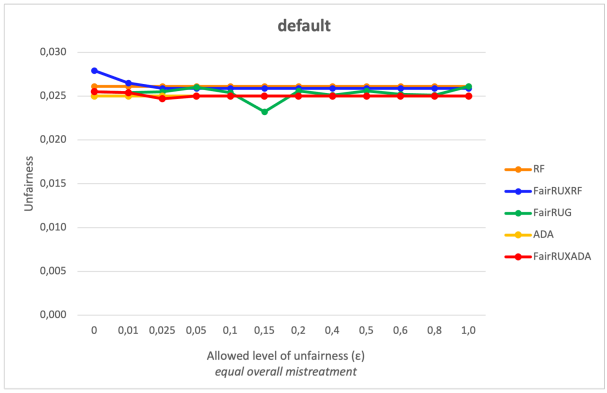
(b) Adult with equal overall mistreatment

Figure 9: Unfairness of all methods based on equal opportunity (left column) and equal overall mistreatment (right column), applied on the adult data set.

B.2 Data set: Default



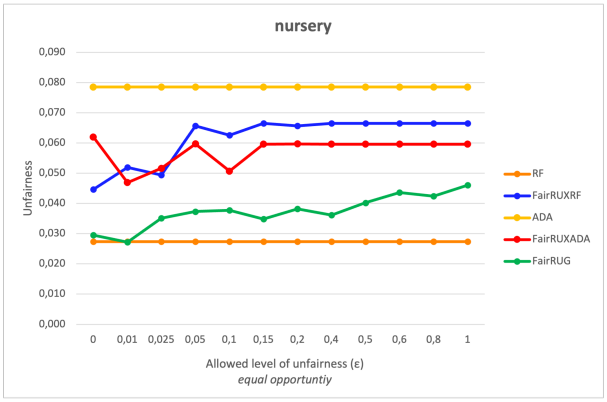
(a) Default with equal opportunity



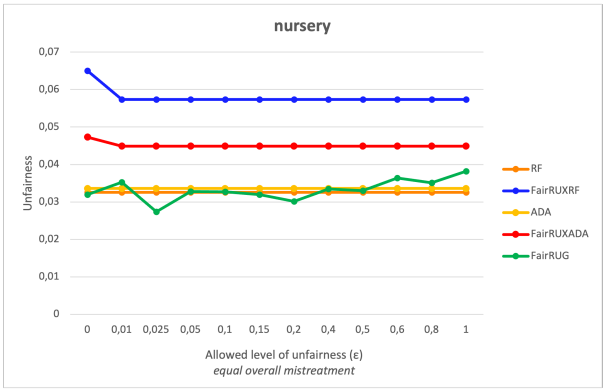
(b) Default with equal overall mistreatment

Figure 10: Unfairness of all methods based on equal opportunity (left column) and equal overall mistreatment (right column), applied on the default data set.

B.3 Data set: Nursery



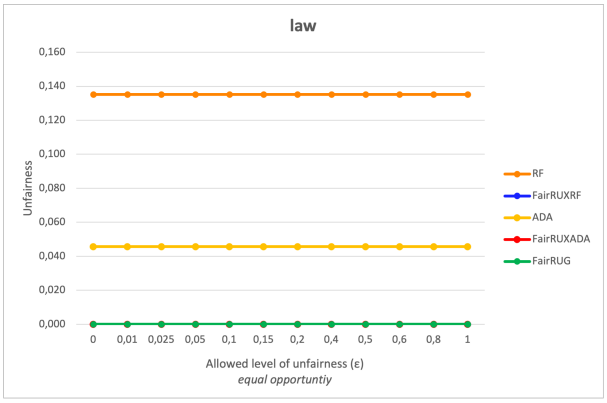
(a) Nursery with equal opportunity



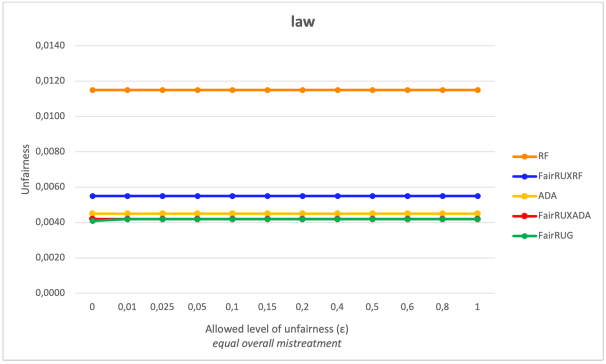
(b) Nursery with equal overall mistreatment

Figure 11: Unfairness of all methods based on equal opportunity (left column) and equal overall mistreatment (right column), applied on the nursery data set.

B.4 Data set: Law



(a) Law with equal opportunity

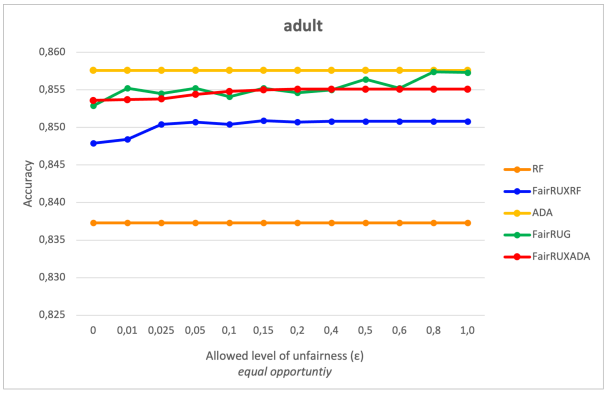


(b) Law with equal overall mistreatment

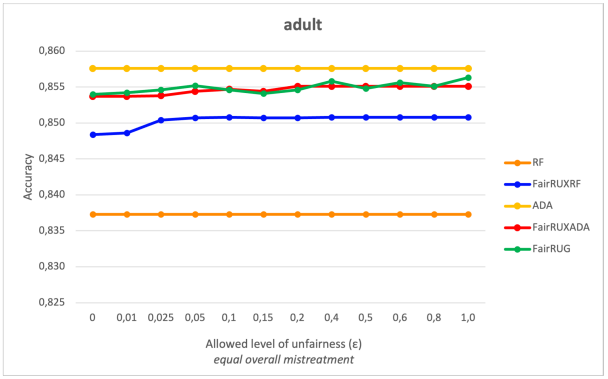
Figure 12: Unfairness of all methods based on equal opportunity (left column) and equal overall mistreatment (right column), applied on the law data set.

C Results: Accuracy Under Varying ϵ

C.1 Data set: Adult



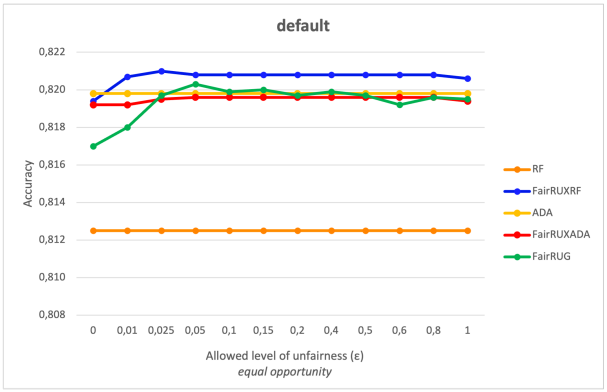
(a) Adult with equal opportunity



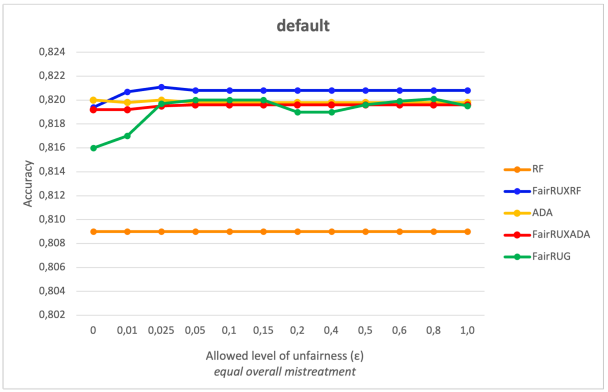
(b) Adult with equal overall mistreatment

Figure 13: Accuracy of all methods based on equal opportunity (left column) and equal overall mistreatment (right column), applied on the adult data set.

C.2 Data set: Default



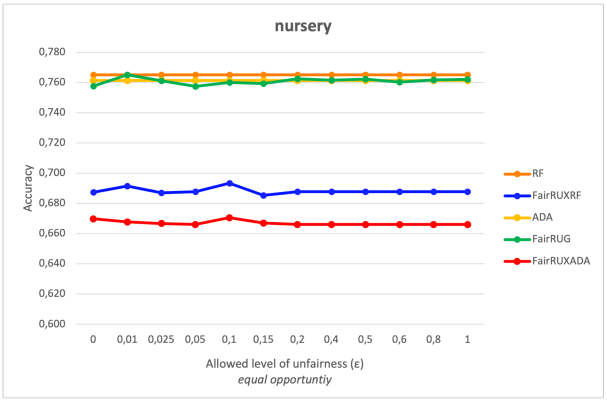
(a) Default with equal opportunity



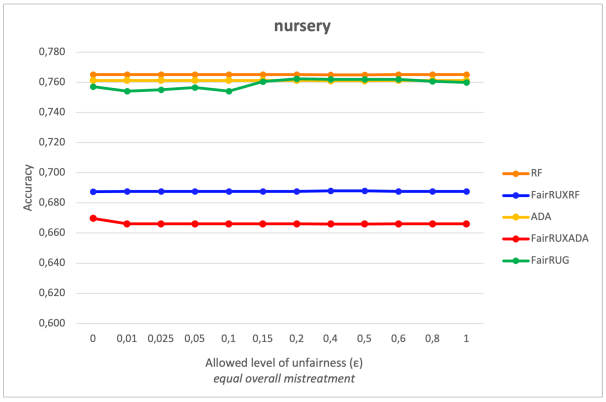
(b) Default with equal overall mistreatment

Figure 14: Accuracy of all methods based on equal opportunity (left column) and equal overall mistreatment (right column), applied on the default data set.

C.3 Data set: Nursery



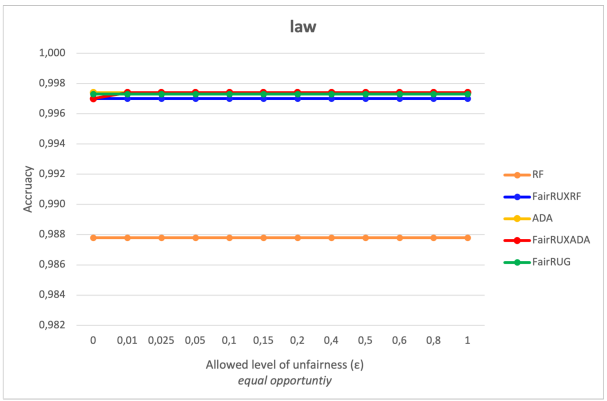
(a) Nursery with equal opportunity



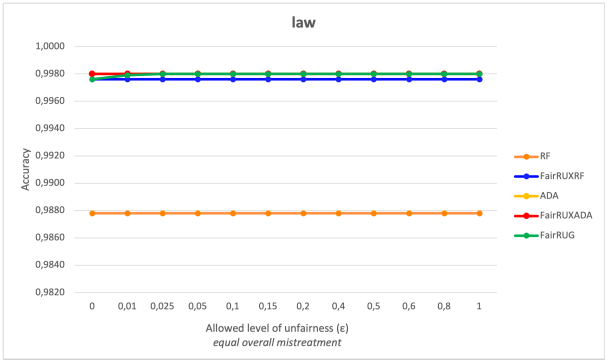
(b) Nursery with equal overall mistreatment

Figure 15: Accuracy of all methods based on equal opportunity (left column) and equal overall mistreatment (right column), applied on the nursery data set.

C.4 Data set: Law



(a) Law with equal opportunity



(b) Law with equal overall mistreatment

Figure 16: Accuracy of all methods based on equal opportunity (left column) and equal overall mistreatment (right column), applied on the law data set.