ERASMUS UNIVERSITY ROTTERDAM

Erasmus School of Economics

Master Thesis Programme BA&QM

April, 2023

Fairness in Neural Networks

Student	*ID*
Saeed Rahimbaks	484633

Supervisor: M.H. Akyuz
Second assessor: P.C. Bouman

Abstract

With the growth of Machine Learning in many applications, the understanding of the fairness of such methods is important too pursue. Especially with the recent surge in popularity surrounding ML, namely the mass usage of Chat GPT, this paper serves as a collection of ideas that explore and combat unfair biased learning. Specifically, we introduce two pre-processing methods SMOTENC and LFR and an in-processing unfairness penalising method. Additionally, we use the post-hoc, model agnostic interpretability method of Shapley values to explain feature importance. All methods use a Neural Network to act as a black-box model, and are tested against a benchmark Neural Network using a 20% hold-out sample. The evaluation is based on accuracy, fairness and comparing the interpretation of features. SMOTENC performed worst of all, while LFR did not perform as expected and had to be dropped from our models. The in-processing method worked as intended, and showed promising results to decreasing unfairness in the models. Feature attributions of the Shapley value method did not significantly differ in order to conclude a general shift in learning. However, it still accomplished its goal of showing the feature importance for predictions. This can help reduce bias by understanding why a model makes its decisions.

Keywords: Machine Learning; Fairness; Neural Network; Shapley interpretation.

Contents

List of Tables

List of Figures

1 Introduction

Nowadays, Machine Learning (ML) is widely used for prediction and analysis. It is used by small companies for specific tasks, and large companies such as Apple and Google alike (Kaur, 2019; Axon, 2020). The increase in the availability of data gives further reason for the adoption of ML in many decision processes.

ML models are generally hard to interpret, which is why they are also called 'black box' models (Molnar, 2022). Often times, users of ML models have to trust the decisions of these models blindly. In cases where the accuracy of the predictions is most important, this is not as much of a problem. For example, ML models used for recommender systems are less in need of interpretability. However, when considering applications used for humans this becomes a critical issue. Interpretation of ML outcomes/decisions is extremely important when it comes to, for example, organ allocation, credit score assessment or recidivism predictions. Interpretability is thus a necessary first step to fairness in ML models. It is undesirable for ML models to base decisions on gender, ethnicity or other possible discriminatory factors. It stands that the decisions of the ML models must be fair, as they have incredible impact on the lives over which they decide. Unfortunately, ML models are not necessarily fair and must thus be applied with care. A machine can never be biased, as it simply follows the instructions of the user. Similarly, a ML model is never intrinsically unfair, but may have been taught discriminatory patterns through biased data. With the rapid adoption of ML in our lives, there is great relevance to understanding ML decisions and the inclusion of fairness in the decision making process.

As such, in this paper we will study fairness in complex black box ML models. Using two datasets with a feature specifying race, we will use several methods to compare and analyse the accuracy and fairness of ML models. This leads us to our main research question: *how does the post-hoc interpretation of a black-box ML model change when incorporating fairness?* To help answer this, we will be researching the following sub-questions:

1. How can we incorporate fairness into ML models?
2. Is there a trade-off between accuracy and fairness?
3. What methods perform best with respect to accuracy and fairness?

Specifically, we will focus on three main aspects of fairness in Machine Learning which will be further explained the Methodology section. As a benchmark model, we will use a Neural Network (NN) model. Then we will use altered NN models using two data pre-processing methods and a fairness optimisation method, as well as combinations thereof. We will be comparing these with respect to accuracy and three fairness measures. Additionally, we will use a post-hoc interpretability method to compare the interpretation of the abovementioned methods.

Fairness in ML is an important concept to consider. Disregarding fairness completely can create disadvantageous scenarios for minority groups. This is not desirable and not on par with modern standards of inclusion. Even more so, we want to avoid discrimination made in ML models due to insufficient knowledge of unfairness in ML. Fairness measures have been incorporated into ML objective functions before. The relevance of this paper lies in combining existing fairness methods for ML, as well as an extension to an existing method. Apart from scientific relevance, our research will be useful for all ML practitioners. Furthermore, insights can be gained from using our proposed methods to further develop a standard for fairness in ML. Our results show that the methods used indeed improve fairness, and post-hoc interpretation allows us to intercept unfair cases.

The structure of this paper will be as follows. We will first delve into the relevant literature about our methods, incorporating fairness in ML and post-hoc interpretability. This is followed by a closer inspection of the data and its characteristics. The models, methods and evaluation we will be using are then further detailed in the subsequent section. After this, the results will be interpreted and discussed. Lastly, we summarise our findings, limitations and directions for future research in the conclusion.

2 Literature

In this section we discuss relevant literature that was useful for our research. Many ideas and methods were considered, but they can be condensed into three categories. Namely: black-box models, fairness in

ML and post-hoc interpretability.

NNs are the base ML method we use in this paper. As such, it is important to understand its features and inner-workings in detail, so that we can manipulate it to our needs. Dasaradhs (2020) describes in simple terms the parts of a NN, how they work individually and how it all comes together to form a ML method. They show the mathematical notation behind the input-, hidden- and output layers of NNs. Furthermore, it delves slightly into the basics of the learning algorithm used for NNs, namely backpropagation. Note that backpropagation is highly dependent on the chosen cost function.

To further understand the cost function of a NN with the task of predicting a binary target, Dasaradhs (2020) describes the widely used binary cross-entropy loss function. We will be using this cost function for our NN, as well as a modified version. Ebert-Uphoff et al. (2021) dives further into cost functions and their requirements, namely differentiability and computational feasibility. NNs are powerful models, and proven to be capable of approximating many non-linear functions. The Universal Approximation Theorem confirms this notion (Hornik, Stinchcombe, and White, 1989). However, we must not forget its flaws. NNs, like many other ML models, have limited interpretation and are prone to bias or overfitting. We will be addressing some of these disadvantages later in the paper.

Fairness can be a difficult concept to translate into a quantifiable metric. Several techniques exist to improve fairness in ML. One such technique is SMOTENC, which falls under the category of pre-processing methods. Chawla et al. (2002), the authors, find that SMOTENC broadens the decision boundary for minority class prediction, which is desired over other sampling methods. As such, this method is suitable for our research and will be used as one of our pre-processing methods. Although its main advantage is adding more data synthetically, it can also be seen as its disadvantage. Creating data which does not correspond to actual samples, or in many cases persons, may give rise to the concern that this can lead to unfairly learned models. Another pre-processing method that will be used in this paper is called Learned Fair Representation (LFR), as detailed in Zemel et al. (2013). They find that LFR diminishes discrimination while retaining accuracy to relatively high values. This is precisely what this paper requires, and will thus be used as our second pre-processing method. Note, however, that this method is limited to a single fairness measure called Demographic Parity. It is not a general framework for incorporating any fairness measure, which would be preferable.

Further fairness implementations are based on metrics and definitions of fairness. The most commonly used metrics are derived from certain proportions between metrics of the confusion table, or extensions thereof. These metrics are discussed and explained in Pessach and Shmueli (2020). We will be using the measures Demographic Parity, Equality of Opportunity and Equalized Odds. These metrics are not convex and can lead to issues in optimisation (Ebert-Uphoff et al., 2021). However, Bendekgey and Sudderth (2021) propose a method that circumvents this issue. They construct a framework of Lagrangian relaxations for fairness constraints. Their work is a step forwards for fairness in ML and will be used in our paper for implementing fairness measures into a NN. Although they provide a somewhat general framework, they do not mention the inclusion of other fairness measures. We propose an extension that regards Equalized Odds in their framework. Further research to such extensions is needed to further solidify this method.

Interpretability in ML is divided into two main groups. The first is inherent to- or derived from a specific model's structure, while the second is model-agnostic. We will be focusing on the latter. Molnar (2022) provides a listing of many such methods and explains them in detail. Specifically, based on the Shapley value, we will be using a method called SHAP. Opposed to existing methods such as LIME, SHAP is derived from game theory and boasts many desirable properties. It is further explained in detail by its authors in Lundberg and Lee (2017). They also provide a programming implementation of this method, which we will be making use of for our paper. This method is a theoretically sound option for post-hoc interpretation, though it boasts some flaws as well. In general, the interpretation from this method should be carefully inspected, opposed to simply believed. SHAP could be used to give false interpretations where actual causal relationships might not occur. Furthermore, SHAP can become quite infeasible for enormous datasets. This is especially the case when dealing with wide-format datasets where the feature space is large. Our dataset is however not too large to be infeasible for SHAP. Additionally, SHAP requires access to the data for real time feature importance calculations. This may be unfit for certain ML tasks. However, these disadvantages can be dealt with, thus we will be using SHAP as our method for post-hoc interpretation.

This paper will add upon the existing literature by combining several fairness methods that are used independently in other papers. The methods used are further explained in detail in Section 4. Our work may give new insights into what direction a ML user would want to pursue for fairness in ML. Furthermore, the novel idea of post-hoc interpretability deviations due to fairness implementations may give new insights into the reasoning of ML models with respect to bias and discrimination.

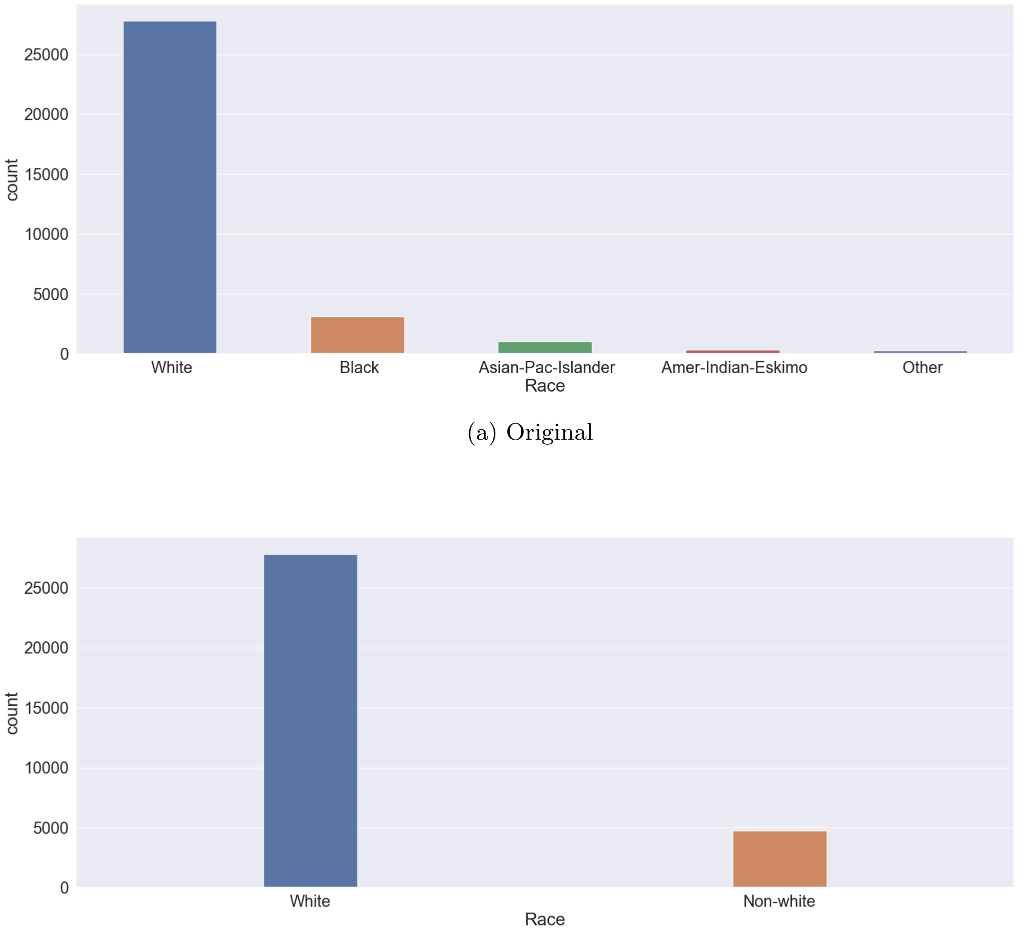
3 Data

Fairness in ML is not a new subject, and some datasets have been used by several researchers to corroborate findings. Those are the Adult dataset and Compas dataset. Both datasets have the same sensitive feature, namely *Race*, around which our methods are built and tested. Furthermore, results from our methods may be inherent to the dataset used. In the following subsections, the Adult- and Compas datasets will be explored and relevant features will be explained.

3.1 Adult dataset

This dataset is obtained from an online archive (*http://archive.ics.uci.edu/ml/index.php*). The sample size is $48,842$, of which $16,281$ will be used as a test sample. This is lowered to $45,222$ and $15,060$ respectively after removing incorrect data. The target variable is whether a person's income is higher than $50,000$ dollars per year (classification) and there are 14 features in the dataset. This dataset is suitable for the problem at hand because it contains two features that are 'sensitive attributes', namely *Race* and *Gender.*

The variable *Race* will be used as our sensitive attribute in order to measure fairness. Specifically, *Race* contains the groups *White*, *Black*, *Asian-Pac-Islander*, *Amer-Indian-Eskimo* and *Other*. Below in Figure 1a we see a large discrepancy between the amount of data from white people opposed to the remaining race groups. Therefore, we decided to reduce the groups to the groups *White* and *Non-white* such that it is a binary variable. The new distribution of this transformed variable is shown in Figure 1b. The class imbalance is still present, but to a lesser extent. In Section 4.2.1 we will see how we can further combat this issue.



(a) Original

(b) Transformed

Figure 1: Histogram of *Race*

Although the variable *Sex* will not be used as our sensitive attribute, it will likely play a role in the prediction task due to the gender wage disparity (CBS, 2020). Its distribution is shown in Figure 2 below. We see a class imbalance of about 2 : 1 for $Male$ and $Female$ respectively.

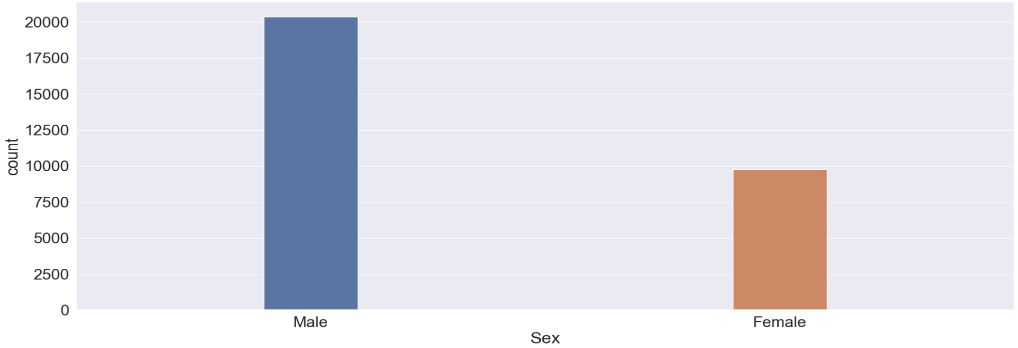


Figure 2: Histogram of *Sex*

Finally, the variable *Label* is a transformed variable that is equal to 1 when an individuals income is larger than 50, 000 per year, or equal to 0 otherwise. This will serve as our dependent variable in the models and our prediction task. Its distribution is displayed in Figure 3. We find a class imbalance of about 1 : 3. This could pose problems for our models but will be dealt with accordingly, as explained in Section 4.2.1.

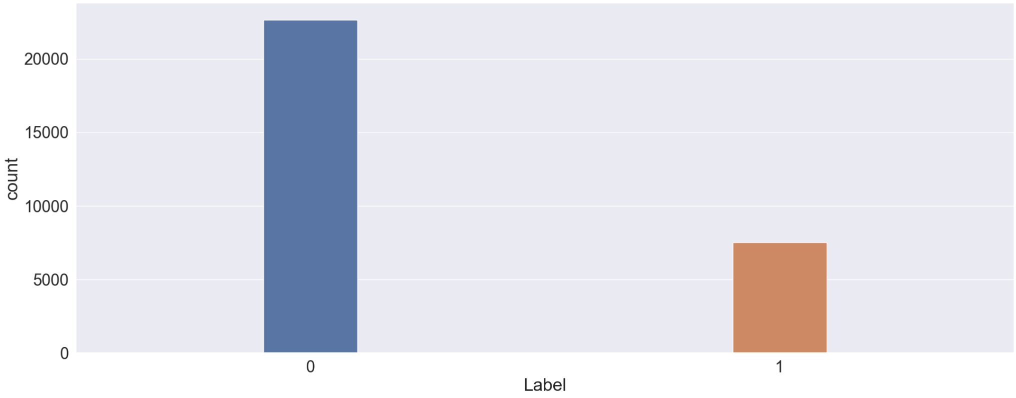


Figure 3: Histogram of *Label*

3.2 Compas dataset

The Compas dataset (Angwin et al., 2016) is accessed using a SQL database file. There are many variables available in the dataset, but most are irrelevant for our purpose. Some examples of irrelevant features are first- and last names, date of sample gathering and case number. The relevant features are chosen and thus subjective. We use the variables *Sex*, *Age*, three types of juvenile felony counts, *Priors count*, *Charge degree* and *Decile score*. Notable is that all of these variables are either categorical- or count data. This poses some issues for a method called SMOTENC which is explained further below. One could alleviate this by including several continuous variables if those are available.

Furthermore, the sensitive attribute we will use is *Race*, similar to the Adult dataset. The distribution of this variable is shown in the figure below:

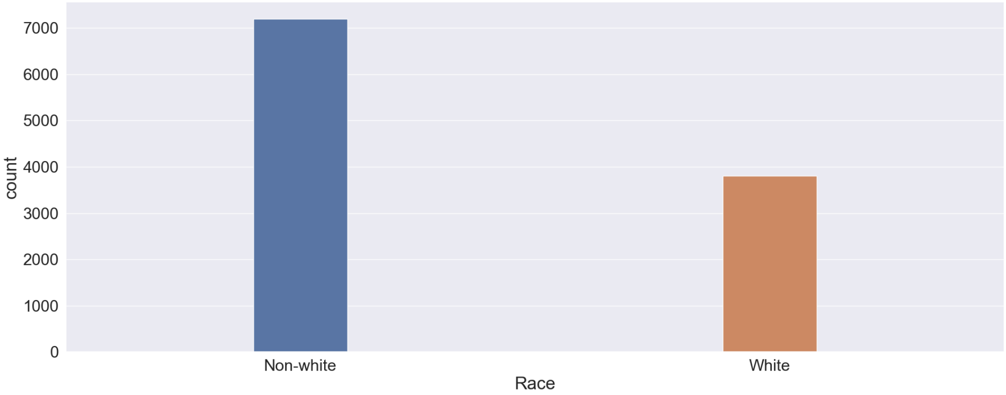


Figure 4: Histogram of *Race*

We see that the distribution of *Race* is more equalised, and class *Non-white* is actually more prevalent than *White*. The class imbalance is roughly 1 : 2. This will not pose a problem for our models, as the imbalance is not severe.

Furthermore, the variable *Sex* could also be used as the sensitive attribute. Its distribution is shown in the figure below:

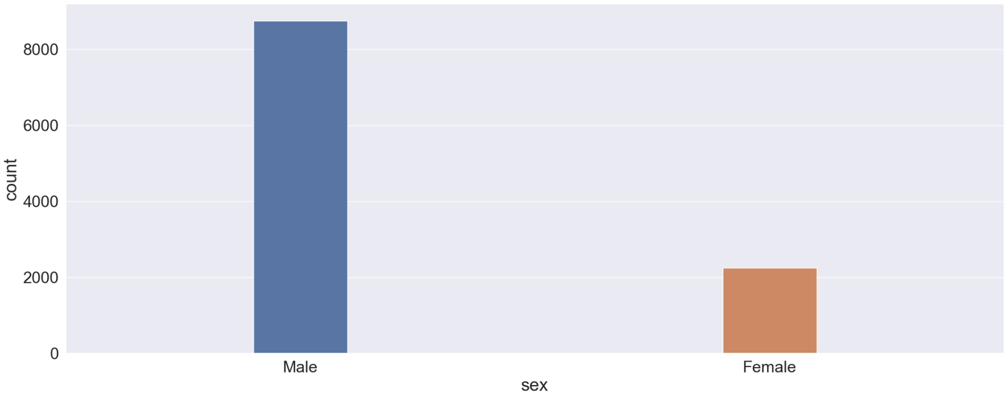


Figure 5: Histogram of *Sex*

We see that the class imbalance is about 1 : 3. This class imbalance is a large, but irrelevant as it is not used as our sensitive feature. If one would like to use this as a sensitive feature, certain methods explained in Section 4.2 can help alleviate this issue.

Lastly, the dependent variable is *Label* and will be the prediction task for our models. It represents whether a sample (person) is a recidivist. That is, a value of 1 means the person in question is a recidivist, and a value of 0 means he or she is not. The histogram of this variable is shown below:

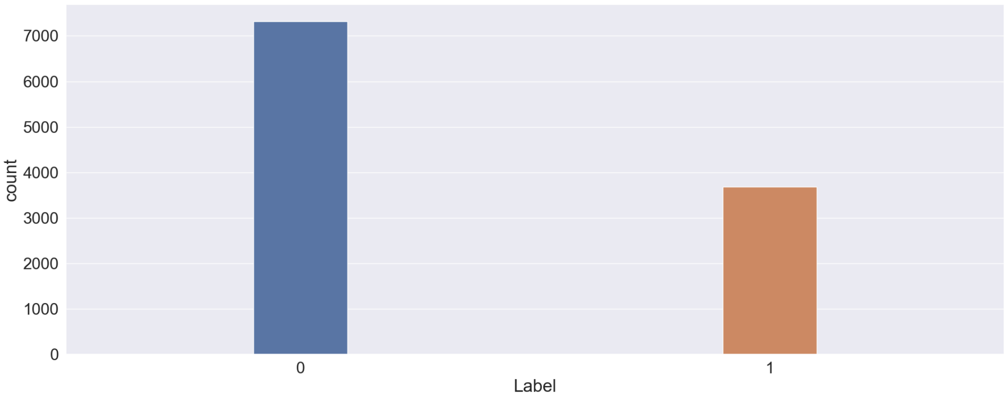


Figure 6: Histogram of *Label*

We see again that the class imbalance is less compared to the Adult dataset. Similarly to *Race*, the class imbalance is roughly 1 : 2. This is beneficial for our model, as a large class imbalance can pose problems for prediction as explained above.

4 Methodology

This section explains our proposed methods for combating fairness. The base model is explained upon which we will build extensions. Then, a post-hoc interpretability method is presented and explained, after which we will explain the evaluation of all models.

4.1 Neural Network

A Neural Network (NN) falls in the category of supervised learning. It consists of an input layer containing the data, the output layer with predictions, and several so-called hidden layers. These hidden layers recursively make use of linear combinations (similar to regression), and non-linear activation functions. This layered structure allows the NN to approximate any nonlinear function between the labels and the data, $y = f(x_1, ..., x_N)$. The approximation is noted as $\hat{f}(x_1, ..., x_N)$. We denote our data in the matrix X with elements $x_{ij}, i \in \{1, ..., N\}, j \in \{1, ..., M\}$, where N is the sample size and M is the number of features. Additionally, we will split the sensitive attribute *Race*, a_i, from the original data and denote the remaining data as x^*_{ij}, where the following relation will hold:

$$x_{ij} = [x_{i0}, x^*_{ij}, a_i], \tag{1}$$

where x_{i0} is equal to $1, \forall i \in 1, ..., N$, to include a bias term. This is done for clarity when needing to differentiate between the sensitive attribute and the remaining data. Lastly, y will contain the labels and has elements $y_i, i \in \{1, ..., N\}$.

Neural Networks optimise their weights by means of back propagation and gradient descent. The cost function of the most popular classification NN is as follows:

$$\begin{aligned} C(\beta) &= \frac{1}{N}\sum_{i=1}^{N} C_i(\beta) & (2)\\ &= \frac{1}{N}\sum_{i=1}^{N} -y_i \log(\hat{y}_i) & (3)\\ &= \frac{1}{N}\sum_{i=1}^{N}\sum_{k=1}^{K} -y_{ik} \log \hat{y}_{ik}, & (4)\end{aligned}$$

which is known as the binary cross-entropy cost function (Godoy, 2019). Furthermore, k is the subscript describing the node of the output layer. Note that the terms cost function and loss function will be used interchangeably in this paper, and the term objective function will refer to the same function but in a mathematical context. A standard NN is a feed forward network, in which the variables x_{ij} are propagated through linear combinations and non-linear activation functions as specified below:

$$\begin{aligned} a^{(1)}_{mi} &= \sum_{j=0}^{M_1} x_{ij}\beta^{(1)}_{mj} & (5)\\ z_{mi} &= h(a^{(1)}_{mi}) & (6)\\ a^{(2)}_{ki} &= \sum_{m=0}^{M_2} z_{mi}\beta^{(2)}_{km} & (7)\\ \hat{y}_{ki} &= \sigma(a^{(2)}_{ki}), & (8)\end{aligned}$$

where M_1 and M_2 are the amount of hidden layer nodes for layers 1 and 2 respectively. Furthermore, $h(\cdot)$ is usually the Rectified Linear Unit (ReLU) function, $\max\{0, x\}$, but can be another non-linear function. $\sigma(\cdot)$ is the well known sigmoid function, $\frac{1}{1+e^{-x}}$. The sigmoid function is widely used because it converts the unbounded range $[-\infty, +\infty]$ to the bounded range $[0, 1]$, which is needed for the classification task. The sigmoid function is however not the only function that achieves this, and thus can be changed to our needs.

Backpropagation moves in the opposite direction of the network in order to optimise the weights $\beta^{(1)}$ and $\beta^{(2)}$ by recursively applying the chain rule of differentiation, as shown below:

$$\begin{aligned} \frac{\partial C_i}{\partial \beta^{(2)}_{km}} &= \frac{\partial C_i}{\partial \hat{y}_{ki}} \cdot \frac{\partial \hat{y}_{ki}}{\partial a^{(2)}_{ki}} \cdot \frac{\partial a^{(2)}_{ki}}{\partial \beta^{(2)}_{km}} & (9)\\ \frac{\partial C_i}{\partial \beta^{(1)}_{mj}} &= \sum_{k=1}^{K} \frac{\partial C_i}{\partial \hat{y}_{ki}} \cdot \frac{\partial \hat{y}_{ki}}{\partial a^{(2)}_{ki}} \cdot \frac{\partial a^{(2)}_{ki}}{\partial z_{mi}} \cdot \frac{\partial z_{mi}}{\partial a^{(1)}_{mi}} \cdot \frac{\partial a^{(1)}_{mi}}{\partial \beta^{(1)}_{mj}}. & (10)\end{aligned}$$

The weights $\beta^{(1)}$ and $\beta^{(2)}$ are then optimised by means of a gradient descent algorithm.

The optimisation of the cost function with respect to the weights is done by gradient descent. Changes in the objective function can consequently hinder the convergence of the chosen gradient descent algorithm. It is of importance to thus carefully design this new objective function, such that the gradient descent algorithm can still converge. The new objective function must be differentiable, and in practice the function must be fast to compute due to the amount of times it will be called in a program. To ensure convergence, both of these requirements must hold. However, we will explain how we can work around this to some extent in the subsequent subsection.

4.2 Data pre-processing

As stated in the previous sections, ML models cannot inherently be biased or unfair due to their nature of following logical rules set by the designer. Unfairness is created through the design of the model or, more prominently, the unfair nature a dataset may have. As such, we will use data pre-processing methods to combat this.

There is a difference between data manipulation and the manipulation of models, but they serve a common purpose in this case. We will apply the data-preprocessing methods described below to our benchmark NN. They will be considered as models in the comparison and evaluation with our custom models that are further detailed in the following subsections

4.2.1 SMOTENC

Specifically, we will use SMOTENC (Chawla et al., 2002) to augment the data such that there is no group imbalance in the sensitive attribute *Race*. This can prevent our model from learning unfair relationships in regards to *Race* due to the scarcity of data points of the minority group *Non-white*. SMOTENC is a synthetic oversampling technique. Where the most basic oversampling technique is simply copying data points, SMOTENC augments the data by creating unique data points close to the original data points. SMOTENC takes the minority class data points and calculates their K-Nearest-Neighbours (KNN). It then takes the difference between the original data points and its nearest neighbour, and multiplies this by a randomly generated value in the range $[0, 1]$ to create its synthetic data points. The amount of minority class data points used and the value of K for the KNN algorithm are decided based on the amount of oversampling needed. Although these new data points are synthetic and not based on the data of an actual person, they are closely related to the actual data. In concise terms, SMOTENC creates these new data points from two existing data points by averaging with random weights. This method will alleviate the class imbalance and may prove to be valuable in increasing fairness in our model.

4.2.2 Learned Fair Representation

Next we will use a method called Learned Fair Representation (LFR), as detailed in Zemel et al. (2013). This method learns a new representation of the data such that it is more fair, while attempting to retain as much information as the original dataset has. It does this by creating a representation through so called 'prototypes', a multinomial variable Z, where the probability of a data points being a certain prototype is denoted as follows:

$$\begin{aligned} M_{ik} &:= \Pr[Z = k|x_i] && (11) \\ &= \frac{\exp\left(-d(x_i, v_k)\right)}{\sum_{l=1}^{K} \exp\left(-d(x_i, v_l)\right)}, && (12) \end{aligned}$$

where $d(x_i, v_k)$ is a distance measure and v_k denotes the learned prototype vector for prototype k. Zemel et al. (2013) propose a weighted $L2$ (squared-Euclidian) distance: $d(x_i, v_k; \alpha) = \sum_{j=1}^{M} \alpha_j (x_{ij} - v_{kj})^2$. For notation, let $X^{(1)}$ denote the subset of X where $a_i = 1$, and let $X^{(0)}$ denote the subset of X where $a_i = 0$. Incorporating this into the model leads to a modified loss function of three terms, where each term has its own minimisation aim. The loss function is specified as follows:

$$L(\beta, A) = A_x L_x + A_y L_y + A_z L_z, \qquad (13)$$

where A_x, A_y and A_z are hyperparameters that determine the weight for each term of the loss functions, not to be confused with the variable weights in Equation (11). The terms in the loss function are defined as follows:

$$L_x = \sum_{i=1}^{N} (x_i - \hat{x}_i)^2 \qquad (14)$$

$$L_y = \sum_{i=1}^{N} -y_i \log(\hat{y}_i) \qquad (15)$$

$$L_z = \sum_{k=1}^{K} \left| M_k^{(1)} - M_k^{(0)} \right|, \qquad (16)$$

where $\hat{x}_i = \sum_{k=1}^{K} M_{ik} v_k$, $\hat{y}_i = \sum_{k=1}^{K} M_{ik} w_k$, $M_k^{(1)} = \frac{1}{|X^{(1)}|} \sum_{i \in X^{(1)}} M_{ik}$ and $M_k^{(0)} = \frac{1}{|X^{(0)}|} \sum_{i \in X^{(0)}} M_{ik}$. The interpretation of the terms is as follows:

- L_x concerns the quality of the mapping from X to Z. The term measures the discrepancy between the representations $\hat{x}_i$ and the original data x_i by means of a squared error metric.
- L_y is the accuracy term similar to the original loss function explained in Section 4.1.
- L_z is the term for Demographic Parity, which is the fairness measure that will be minimised in this method.

The hyperparameters will be tuned by means of a grid search of a size that is computationally feasible. However, note that alpha may also be set specifically to ensure certain importance weight to the terms explained above. We optimise Equation (13) with gradient descent to obtain the estimates of α, v and w. Next, the benchmark NN model using this custom loss function with new data representations will be solved by means of a gradient descent algorithm. Note that the objective function is no longer convex and may converge to a local optimum.

4.3 Fairness measures

4.3.1 Demographic Parity

There are three fairness measures we will be using in this paper. First is Demographic Parity (DP), which requires the people in both groups of the sensitive attribute to have equal probability to be classified positive (Pratik and Mykola, 2018). Mathematically, that would mean:

$$\Pr[\hat{y} = 1 | A = 1] = \Pr[\hat{y} = 1 | A = 0], \tag{17}$$

where A is the sensitive attribute *Race*. DP can be relaxed to allow our models to incorporate it into their decision making process. A general relaxation is the 'p% rule', which requires $\frac{\Pr[\hat{y}=1|A=1]}{\Pr[\hat{y}=1|A=0]} < p$. p is often chosen to equal 80% due to the 80% rule in American regulation (Commission, 1978). We can then minimise the Demographic Parity Difference (DPD):

$$DPD = \Pr[\hat{y} = 1 | A = 1] - \Pr[\hat{y} = 1 | A = 0], \tag{18}$$

$$DPD_p = \Pr[\hat{y} = 1 | A = 1] - p * \Pr[\hat{y} = 1 | A = 0]. \tag{19}$$

$$\tag{20}$$

However, in Section 4.3.4 we will see how a general relaxation is not sufficient for solving this constrained optimisation.

4.3.2 Equality of Opportunity

The second fairness measure is Equality of Opportunity (EO), which requires the model to classify both groups of the sensitive attribute with equal false positive rates. Note that, similar to the approach of Bendekgey and Sudderth (2021), we can translate the equation below to be for false negative rate if that suits the problem better. It is calculated as follows:

$$\Pr[\hat{y} = 1 | Y = 0, A = 1] = \Pr[\hat{y} = 1 | Y = 0, A = 0], \tag{21}$$

where again A is the sensitive attribute *Race*. Similarly to the previously described relaxation, we aim to minimise the Equality of Opportunity Difference (EOD):

$$EOD = \Pr[\hat{y} = 1 | Y = 0, A = 1] - \Pr[\hat{y} = 1 | Y = 0, A = 0]. \tag{22}$$

4.3.3 Extension: Equalized Odds

Equalized Odds (EOdds) is similar to Equality of Opportunity. It is satisfied if both the false positive rate and the false negative rate is equal for both groups of the sensitive attribute. In mathematical notation, that is:

$$\Pr[\hat{y} = 1 | Y = y, A = 1] = \Pr[\hat{y} = 1 | Y = y, A = 0], \forall y \in \{0, 1\} \tag{23}$$

where we again want to minimise its difference:

$$\Pr[\hat{y} = 1|Y = y, A = 1] - \Pr[\hat{y} = 1|Y = y, A = 0]. \tag{24}$$

Note that this formulation can be split up into two equation regarding equal false positive rate (such as in Equation (22)) and false negative rate.

4.3.4 Fairness in loss function

The fairness measures described above will be incorporated into the loss function of our NN model. We adhere to the implementation of Bendekgey and Sudderth (2021), as the goal of this paper is not to find a novel implementation of fairness measures into the objective function of our model. Incorporating the fairness measures described above would lead to a non-convex function that is non-differentiable, which violates the requirements listed in Section 4.1. A NN cannot backpropagate through such a network and will fail to converge. Where preceding common relaxations (e.g. linear relaxation) fail to ensure fairness or provide any guarantee to their fairness measure, Bendekgey and Sudderth (2021) propose to use a Log-Sigmoid Sums (LSS) relaxation to guarantee a certain threshold or upper-bound to the fairness measure. The upper bound can be set by determining λ, which can be considered a hyper parameters. It will be selected using a grid search. The augmented loss function will be generally noted as:

$$C(\beta) = \frac{1}{N} \sum_{i=1}^{N} \left[-y_i \log(\hat{y}_i) + \lambda R_g \right], \tag{25}$$

where $g(r) = -\frac{1}{\log(2)} \log(\sigma(-r))$ is the scaled log-sigmoid function and R_g is defined as follows:

$$R_g^{DP} = \begin{cases} \frac{g(\hat{f}(x_1,\ldots,x_N))}{p_{1\cdot}} & \text{if } a_i = 1, \\ \frac{g(-\hat{f}(x_1,\ldots,x_N))}{p_{0\cdot}} & \text{if } a_i = 0. \end{cases} \tag{26}$$

$$R_g^{EO} = \begin{cases} \frac{g(\hat{f}(x_1,\ldots,x_N))}{p_{10}} & \text{if } a_i = 1, y = 0, \\ \frac{g(-\hat{f}(x_1,\ldots,x_N))}{p_{00}} & \text{if } a_i = 0, y = 0, \\ 0 & \text{else.} \end{cases} \tag{27}$$

where p_{ay} describes the proportions of the data with group $a \in \{0, 1\}$ and label $y \in \{0, 1\}$. Similarly, $p_{a\cdot}$ describes the proportions of the data with group $a \in \{0, 1\}$ but with any label.

The loss function in Equation (25) is now convex and will be optimised for DP and EO with their respective relaxed constraint by means of a gradient descent algorithm, specifically *ADAM* which is available in most ML and optimisation packages.

For EOdds, we extend the framework of Bendekgey and Sudderth (2021) as they only used DP and EO. As stated before, we can split the EOdds difference derived from Equation (23) into two parts. This results in the following surrogate fairness function:

$$\begin{aligned} R_g^{EOdds} = & \begin{cases} \frac{g(\hat{f}(x_1,\ldots,x_N))}{p_{10}} & \text{if } a_i = 1, y = 0, \\ \frac{g(-\hat{f}(x_1,\ldots,x_N))}{p_{00}} & \text{if } a_i = 0, y = 0, \\ 0 & \text{else.} \end{cases} \\ + & \begin{cases} \frac{g(\hat{f}(x_1,\ldots,x_N))}{p_{11}} & \text{if } a_i = 1, y = 1, \\ \frac{g(-\hat{f}(x_1,\ldots,x_N))}{p_{01}} & \text{if } a_i = 0, y = 1, \\ 0 & \text{else.} \end{cases} \end{aligned} \tag{28}$$

Note that we did not construct as similar proof as Bendekgey and Sudderth (2021) did for DP and EO, as that is out of the scope of this paper. Further research into proving bounds for other fairness measures should be done.

4.4 Interpretability method

As stated before, all models will be compared and evaluated on their interpretation. Although a somewhat intangible metric, we have established before that interpretability is a key component to fair ML. The challenge here is obtaining good interpretation from our black-box models. The method we will use is SHAP, a global post-hoc model-agnostic interpretation method that is becoming popular in the ML field. It is based on the Shapley value concept in the field of study of game theory.

4.4.1 Shapley

To understand SHAP, we must thus first understand Shapley and its properties. Adhering closely to the notation used in Lundberg and Lee (2017), Shapley values in our ML context will be defined as follows:

$$\phi_{ij} := \phi_j(x_i) = \sum_{S \subseteq \{1,...,M\} \setminus \{j\}} \frac{|S|!(M-|S|-1)!}{M!} (\hat{f}_{S\cup\{j\}}(x_{i,S\cup\{j\}}) - \hat{f}_S(x_{i,S})), \tag{29}$$

where $\hat{f}_S$ is the prediction of model f trained on the arbitrary subset S, and x_S pertains the data containing only the subset S of features. For example, $x_{1,\{1,2,3\}}$ would describe data points x_{11}, x_{12} and x_{13}. As such, the interpretation for $\hat{f}_{S\cup\{j\}}(x_{i,S\cup\{j\}}) - \hat{f}_S(x_{i,S})$ is the difference between the prediction of the model trained including feature j and excluding feature j, for row $i \in \{1, ..., N\}$. From these individual data points (local) explanations, we define the feature importance (global) as simply the average Shapley values of that feature. That is:

$$\phi_j^{Shapley} := \frac{1}{N}\sum_{i=1}^{N} \phi_{ij} \tag{30}$$

$$= \frac{1}{N}\sum_{i=1}^{N} \sum_{S \subseteq \{1,...,M\} \setminus \{j\}} \frac{|S|!(M-|S|-1)!}{M!} (\hat{f}_{S\cup\{j\}}(x_{i,S\cup\{j\}}) - \hat{f}_S(x_{i,S})). \tag{31}$$

Shapley values have the followings desirable properties:

1. **Efficiency** states that all Shapley value attributions must sum up to the difference between the predicted value and its average. That is, $\sum_{j=1}^{M} \phi_{ij} = \hat{f}(x_{i1}, ..., x_{iM}) - E[\hat{f}(x_{i1}, ..., x_{iM})], \forall i \in 1, ..., N$.

2. **Symmetry** holds if $\hat{f}_{S\cup\{j\}}(x_{i,S\cup\{j\}}) = \hat{f}_{S\cup\{k\}}(x_{i,S\cup\{k\}}) \implies \phi_{ij} = \phi_{ik}, \forall S \cup \{1, ..., M\} \setminus \{j, k\}, \forall j, k \in \{1, ..., M\}, \forall i \in \{1, ..., N\}$. The interpretation of this is that equal predictions for features j and k imply equal Shapley values.

3. **Dummy** asserts that a feature that has no influence on the prediction must imply a Shapley value of 0. Mathematically, this holds when $\hat{f}_{S\cup\{j\}}(x_{i,S\cup\{j\}}) = \hat{f}_S(x_{i,S}) \implies \phi_{ij} = 0, \forall S \cup \{1, ..., M\} \setminus \{j\}, \forall j \in \{1, ..., M\}, \forall i \in \{1, ..., N\}$.

4. **Additivity** is interpreted when considering games (in our case models) where several pay-outs (predictions) are added. Suppose we have a summation function $g(\hat{f}_1, ..., \hat{f}_B) = \sum_{b=1}^{B} \hat{f}_b$ where the predictions of the individuals models are simply added together. This property states that $\sum_{b=1}^{B} \phi_i(x_i; \hat{f}_b) = \phi_i(x_i; g)$, where $\phi_i(x_i; \hat{f})$ follows the definition of Equation (30) with an additional parameters specifying the model f. This means that the summation of the Shapley values of the B individual games equals the Shapley value of the total game.

These desirable properties make the Shapley value a grounded method to interpretability in ML. Other post-hoc interpretability methods such as LIME (Ribeiro, Singh, and Guestrin, 2016) make assumptions that are not necessarily appropriate and ungrounded in theory. Furthermore, the Shapley value is the only attribution method that boasts the abovementioned properties (Hervé, 2003). This gives it a clear advantage over other such methods in regards to fairness.

However, its main disadvantage is its severely large time complexity and model comparisons. As seen before, the Shapley method must calculate many different permutations of models to calculate even a single Shapley value. This complexity increases linearly with the amount of data points and exponentially with the amount of features. In our case, for the Adult dataset this would equal $48842 * 2^{14}$ model evaluations, which is infeasible for our purposes.

4.4.2 SHAP

This brings us to our proposed method for interpretation, namely SHAP. Lundberg and Lee (2017) created this method and implemented it into the *shap* Python package, which we will be using. SHAP is a feasible alternative to Shapley, and is founded on the same theory. This alone is an advantage over other post-hoc interpretability methods like LIME, where the assumptions are not necessarily appropriate. Further detailed in Lundberg and Lee (2017), the SHAP method boasts the following desirable properties:

1. **Local accuracy** is similar to the Shapley value efficiency property. It states that $\hat{f}(x_{i1}, ..., x_{iM}) = \phi_{i0}^{SHAP} + \sum_{j=1}^{M} \phi_{ij}^{SHAP} x_{ij} = E[\hat{f}(x_{i1}, ..., x_{iM})] + \sum_{j=1}^{M} \phi_{ij}^{SHAP} x_{ij}, \forall i \in \{1, ..., N\}$.
2. **Missingness** is similar to the dummy property of Shapley values. Mathematically, it states that $x_{ij} = 0 \implies \phi_{ij}^{SHAP} = 0$.
3. **Consistency** states that the attribution of a feature should stay equal or increase whenever a change in the original model $f_X^{'}(X)$ results in a non-negative change in the marginal contribution of that feature. That is: $f_X^{'}(X) - f_X^{'}(X \setminus X_j) \geq f_X(X) - f_X(X \setminus X_j) \implies \phi_j^{SHAP}(x_i; f^{'}) \geq \phi_j^{SHAP}(x_i; f)$.

SHAP is a unique solution that complies with these properties. We will use SHAP as our interpretation method, and compare the interpretations for our distinct models.

4.5 Evaluation

In order to evaluate the methods proposed above fairly, we must first ensure they are consistent and model choices are equal. All models will use the same optimisation method, namely *ADAM*, provided by the Tensorflow Python package (Martín Abadi et al., 2015). Furthermore, all models are ran using several random seeds to ensure the models are stable. This means their results are consistent and not prone to small differences in initialisation. The base NN, and all variations thereof, have the same initial values, number of layers and nodes, and seed to ensure that the differences in their results are solely due to the variations and restrictions we pose on the models. This allows us to attribute the resulting differences to our proposed methods.

All models, data pre-processing techniques and their combinations will be evaluated with several methods. These methods are subdivided as:

- Prediction accuracy: how well does the model predict the target variable?
- Fairness measures: how do the models compare with respect to the fairness measures explained in Section 4.3?
- Interpretation: how important are the variables in predicting the target variable? This will highlight the changes in the models when using our proposed variations.

The prediction and fairness performance measures are computed using a 20% hold-out sample. The Shapley values for interpretation are computed with the remaining 80% training sample.

5 Results

The models will be further referred to as follows:

- Model **b**: the base NN with no additional techniques applied. It will be initialised with grid-searched hyper-parameters.
- Model **s**: NN with SMOTENC data-preprocessing method applied.
- Model **d**: NN with custom loss function that includes the surrogate relaxation function for Demographic Parity (DP).
- Model **e**: NN with custom loss function that includes the surrogate relaxation function for Equality of Opportunity (EO).
- Model **eo**: NN with custom loss function that includes the surrogate relaxation function for Equalized Odds (EOdds).

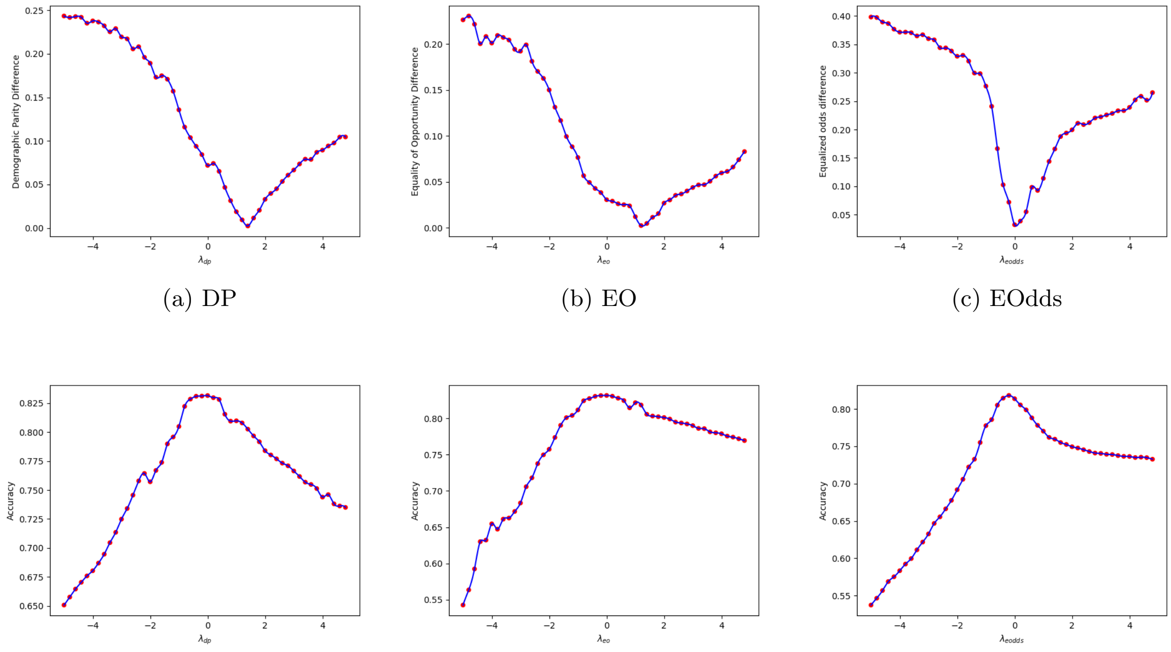
- Model **s+d**: NN where we combine SMOTENC and the surrogate relaxation function for DP.
- Model **s+e**: NN where we combine SMOTENC and the surrogate relaxation function for EO.
- Model **s+eo**: NN where we combine SMOTENC and the surrogate relaxation function for EOdds.

The models that make use of the LFR method explained in Section 4.2.2 did not have meaningful results. The accuracy of these models were simply equal to the proportion of largest class in the outcome. This means the NN did not learn from the data, which in turn means the data holds no meaningful information. The LFR models cannot be learned from in regards to fairness in our case, as they have a DP of 0. However, this is solely because the NN indiscriminately predicts the majority class for all samples. Similarly, the values of EO and EOdds for the LFR models contain no information as well. Furthermore, the disadvantage of LFR in our case is that interpretability of the prototypes is also hard to find. Thus, Shapley values were not meaningful either. In the following parts of this Section, we will not mention the LFR models again.

5.1 Lambda selection

In Figures 7 and 8 below we see the grid search for the λ parameter of the surrogate relaxation models (**d** and **e**). Note that the functions are interpolated using natural cubic splines. Contrary to expectation, plotting DP and EO against their respective λ shows a non-linear function that is neither strictly decreasing nor strictly increasing. One would expect a higher (or lower) λ to drive its respective fairness measure arbitrarily low. However, we see reflection points around 1.4 and 0 for DP and EO respectively in the Adult dataset, as well as for EO in the Compas dataset, where both sides are also non-symmetric. DP in the Compas dataset seems to follow an upward trend with a large drop around $\lambda = -2$. These results are remarkable, and without further research and inspection an explanation is hard to find.

Regardless of this, we can still select some λ to achieve our goals. As Bendekgey and Sudderth (2021) suggest, there simply exists some λ that achieves a certain desired fairness measure, which does not necessarily contradict our findings.

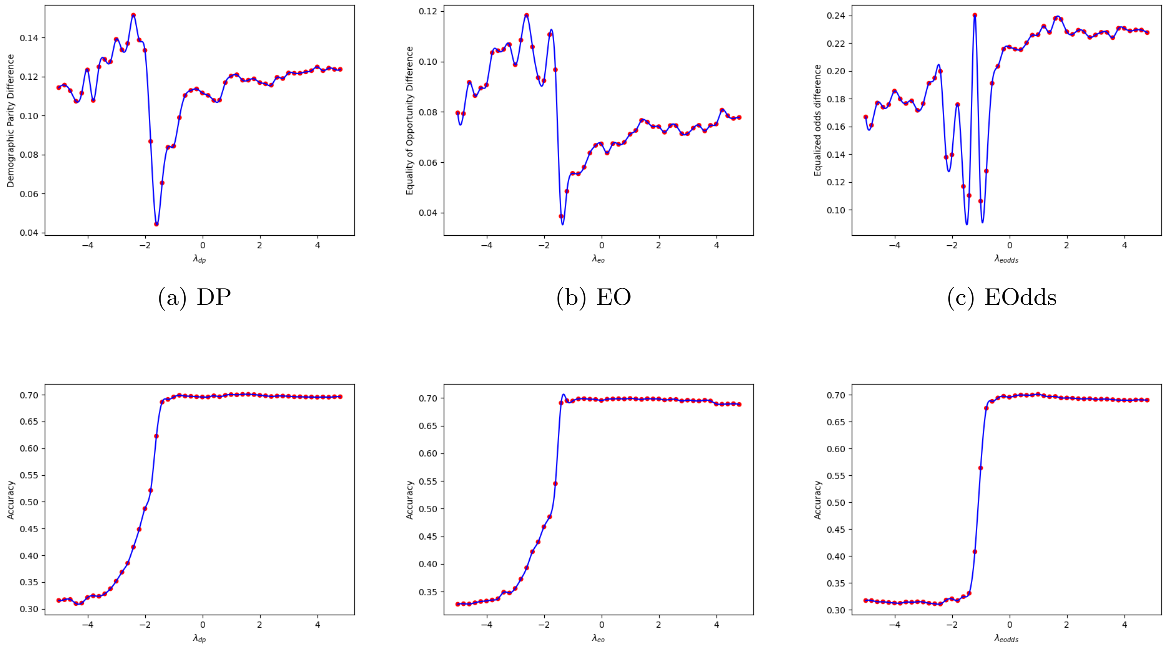


(a) DP (b) EO (c) EOdds

(d) DP accuracy (e) EO accuracy (f) EOdds accuracy

Figure 7: Grid search and spline interpolation of λ (Adult dataset)

For the Adult dataset, we select a λ somewhat close to 0 as we see that the absolute deviation of this number also drives the accuracy of the model down. Note that $\lambda = 0$ corresponds to the base model (**b**). Specifically, we chose $\lambda = 1.4$ for DP and $\lambda = -0.20$ for EO. Interesting is that for EOdds the optimal λ is 0, which corresponds to the base model (**b**).



(a) DP

(b) EO

(c) EOdds

(d) DP accuracy

(e) EO accuracy

(f) EOdds

Figure 8: Grid search and spline interpolation of λ (Compas dataset)

For the Compas dataset, we must first note that the selection of λ can be seen as a trade-off between accuracy and its fairness measure. The optimal λ would be -1.6 for both DP and EO. However, we see that the accuracy is low for those λ's, and increases (but converges) as we increase λ. Therefore, we choose λ equal to -1.4 for both DP and EO. For EOdds, the same notion holds. The optimal value would be $\lambda = -1$, but $\lambda = -0.8$ achieves a better accuracy. We also note that for Figure 8c the graph shows unstable behaviour in the region $(-2.4, -0.2)$.

Further research to the behaviour of λ on a larger range and smaller intervals between points might create some insight to the trade-off between accuracy and fairness. This grid search has a computationally heaven burden, and it therefore out of the scope of this paper. For the other models where a λ is used, it will be chosen in the same manner.

5.2 Prediction accuracy

In this subsection we compare the models on their predictive accuracy. The values are arrived from a Confusion Matrix (CM), as described by Kohavi and Provost (1998). This means the column *Total* describes the proportion of correctly classified samples. The columns *Class 0* and *Class 1* then describe respectively the proportion of true negatives to all negative (0) labels and the proportion of true positives to all positive (1) labels. The results are summarised in the table below:

Table 1: Accuracy (Adult dataset)

Model	*Class 0*	*Class 1*	*Total*
b	0.92	0.55	0.83
s	0.93	0.56	0.84
d	0.95	0.32	0.80
e	0.94	0.44	0.82
eo	0.92	0.55	0.83
s+d	0.97	0.26	0.80
s+e	0.95	0.35	0.81
s+eo	0.93	0.48	0.82

Values are rounded.

We see that model **s** performs best in regards to accuracy. This is due to SMOTENC altering the data such that there is more data to be trained on, as well as less class imbalance which can lead to lower accuracy through only predicting one class. As expected, the base model (**b**) boasts good accuracy as a standard prediction method. We note that in all model using fairness restriction, models **e** outperform models **d**. This may be due to the formulation of model **e**, as its loss function is equal to the loss function only focused on predictive accuracy (as in model **b**) whenever $y = 1$.

Table 2: Accuracy (Compas dataset)

Model	*Class 0*	*Class 1*	*Total*
b	0.89	0.31	0.70
s	0.90	0.27	0.69
d	0.84	0.37	0.69
e	0.91	0.26	0.69
eo	0.90	0.30	0.70
s+d	0.85	0.36	0.69
s+e	0.86	0.27	0.66
s+eo	0.91	0.26	0.69

Values are rounded.

Opposed to the Adult dataset results, we find that model **s** actually achieves a smaller accuracy. It seems SMOTENC learned more about class 0, as its accuracy is increased opposed to model **b**. This is most likely due to the limitations of SMOTENC, as it does not handle datasets with mainly categorical variables well. In our case, the Compas dataset consists of mostly categorical variables and a few counting variables. SMOTENC works better when there is at least one, and preferably more, continuous variable.

Furthermore, we see that the accuracy is relatively stable around 69% for all models. Although we must note that for models involving a λ selection, it is selected slightly away from its optimum to not suffer a large loss in accuracy.

5.3 Fairness measures

In this subsection we compare the models on the fairness measures described in Section 4.3. Specifically, the values are calculated using Equations (18), (22) and (24).

5.3.1 Adult dataset

The results for the Adult dataset are summarised in the table below:

Table 3: Fairness measures (Adult dataset)

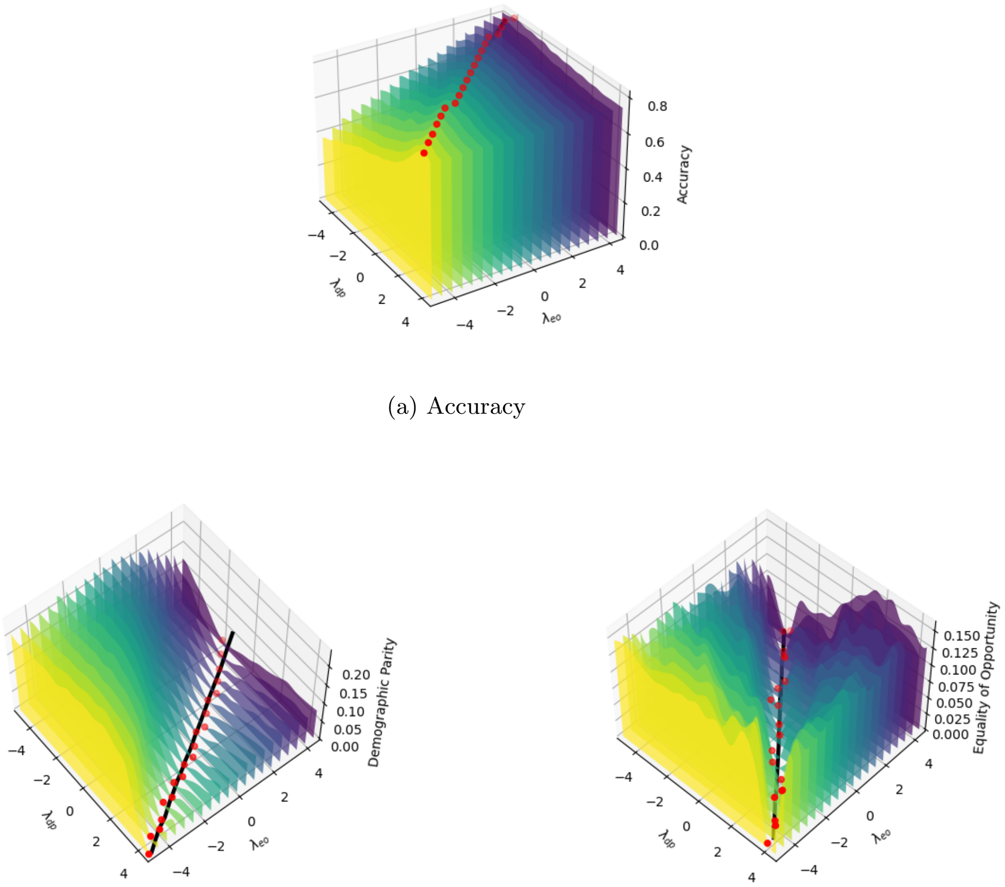
Model	*DP*	*EO*	*EOdds*
b	0.0720	0.0306	0.0465
s	0.0742	0.0256	0.0402
d	0.0023	0.0205	0.0786
e	0.0344	0.0025	0.0462
eo	0.0719	0.0306	0.0465
s+d	0.0008	0.0183	0.0783
s+e	0.0207	0.0024	0.0797
s+eo	0.0746	0.0363	0.0492

Values are rounded.

We see that the models **d** and **e** drive their respective fairness measures, DP and EO, down considerably compared to the base model. Combining SMOTENC and DP in the objective function, model **s+d**, shows even further significant reduction in DP. However, contrary to expectations, this is not the case for model **s+e**, where its EO is even larger than the base model. A reason for this may be that the λ corresponding to this model is chosen over a relatively small range. An optimisation over a considerably larger range may provide a λ that drives EO as low as we would expect from the results of model **s+d**.

Model **s** performs best with respect to EOdds. We did not expect models **eo** and **s+eo** to outperform the other models, as the λ selection showed that a λ of 0 is optimal in this case.

Furthermore, we note that optimising the loss function with a particular fairness measure included does not drive the other fairness measure down as a consequence. One can thus optimise for several fairness measures if it is required for the problem at hand. This result is interesting to pursue. A novel model, which we will refer to as **d+e**, describes a model where both fairness measures are included in the objective function of Equation (25) by means of relaxed surrogates. This model now has two separate lambdas regulating DP and EO, which we denote λ_{dp} and λ_{eo} respectively. The λ's are again found by using a grid search, this time over two axes. This means the computational burden for this grid search is more expensive. The accuracy, DP and EO are shown in the figures below.



(a) Accuracy

(b) DP

(c) EO

Figure 9: Grid search and line fitting of λ's (Adult dataset)

These plots show some interesting characteristics. In Figure 9a we see that the general shape of the curve describing the relation between λ_{dp} seems to be shifting for each λ_{eo}. The red points on the graph show the maximum value of the accuracy for each λ_{eo}, corresponding to a particular λ_{dp}. An unexpected but remarkable result is that the points seem to align in a linear fashion. A line is fit through the points, minimising a squared distance. Note, however, that the line is offset by the z-axis in order to make the line more visible, so any discrepancies in the 'straightness' of the line is due to the 3-dimensional nature of the graph. This line shows a relation between λ_{eo} (y-axis) and λ_{dp} (x-axis) that attains the maximum accuracy. The line is estimated by the equation $y = -1.14x - 0.56$.

Furthermore, both Figures 9b and 9c show a similar line through their respective minima. The former showing the same property of shifting the general shape of the curve for each λ_{eo}. The line for DP can be described by the equation $y = -1.53x + 2.05$.

Figure 9c seems to behave less steady. Even though the graph does have a line that fits nicely through its minima ($y = -1.16x + 0.01$), the values around the minima do not follow a behaviour that looks predictable. This is somewhat expected, as we saw in Figure 7 the difference in stability of the functions for λ_{dp} and λ_{eo}.

The lines seem to generalise a relation between the λ's, but the relations are not the same for the three different metrics. This means we cannot pick a single point to optimise all metrics. However, since the lines are not parallel, there must exist a point where the lines meet. Note that the lines are fit so they do not necessarily touch the optimal points. However, the coordinates of the intersection points between two lines can serve as a starting point for the region in which we can grid search for the optimal λ's. The usefulness of this approach for grid searching may depend on the problem at hand, because the optimising for two points may result in a unsatisfactory value of the third metric. If the third metric is far from its optimum near the crossing of two lines, one could also choose to simply optimise only one λ as we did in models **d** and **e**.

5.3.2 Compas dataset

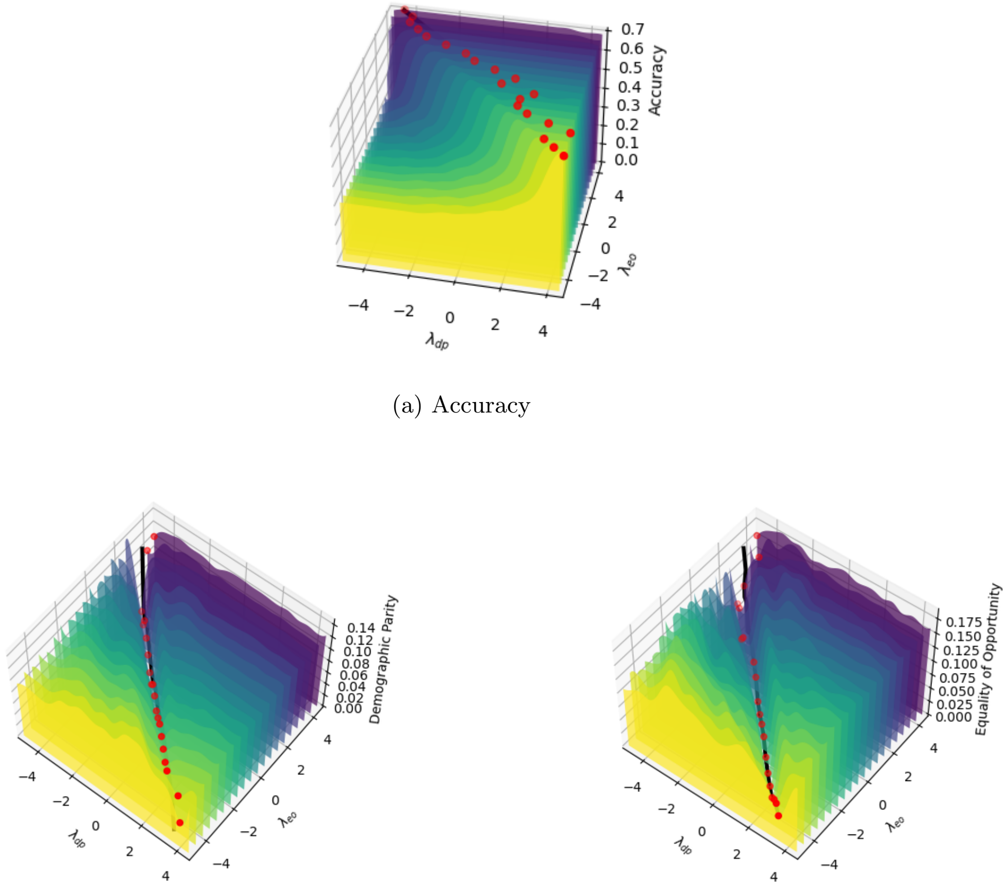
In the table below we find the results for the Compas dataset.

Table 4: Fairness measures (Compas dataset)

Model	*DP*	*EO*	*EOdds*
b	0.11	0.07	0.22
s	0.10	0.06	0.21
d	0.07	0.03	0.11
e	0.07	0.04	0.13
eo	0.10	0.06	0.18
s+d	0.08	0.04	0.14
s+e	0.08	0.05	0.17
s+eo	0.10	0.05	0.20

Values are rounded.

We find that the model **s** does not sufficiently drive DP, EO and EOdds down. We do see that all models involving a λ boast lower fairness measures. Models **eo** and **s+eo** only slightly drive EOdds down compared to model **b**. Interesting is that model **d** and **e** correspond with a lower EOdds than models **eo** and **s+eo**. This is contrary to expectation, and further research should be done as to why this is the case. Opposed to the Adult dataset results, we find that in general incorporating one fairness metric in the objective function actually does drive the other fairness measure down as a result. However, this is not a proof or a guarantee. Similar to what we did with the Adult dataset, we can again construct a novel model that uses both fairness metrics in the objective function. The results are shown in the graph below:



(a) Accuracy

(b) DP

(c) EO

Figure 10: Grid search and line fitting of λ's (Compas dataset)

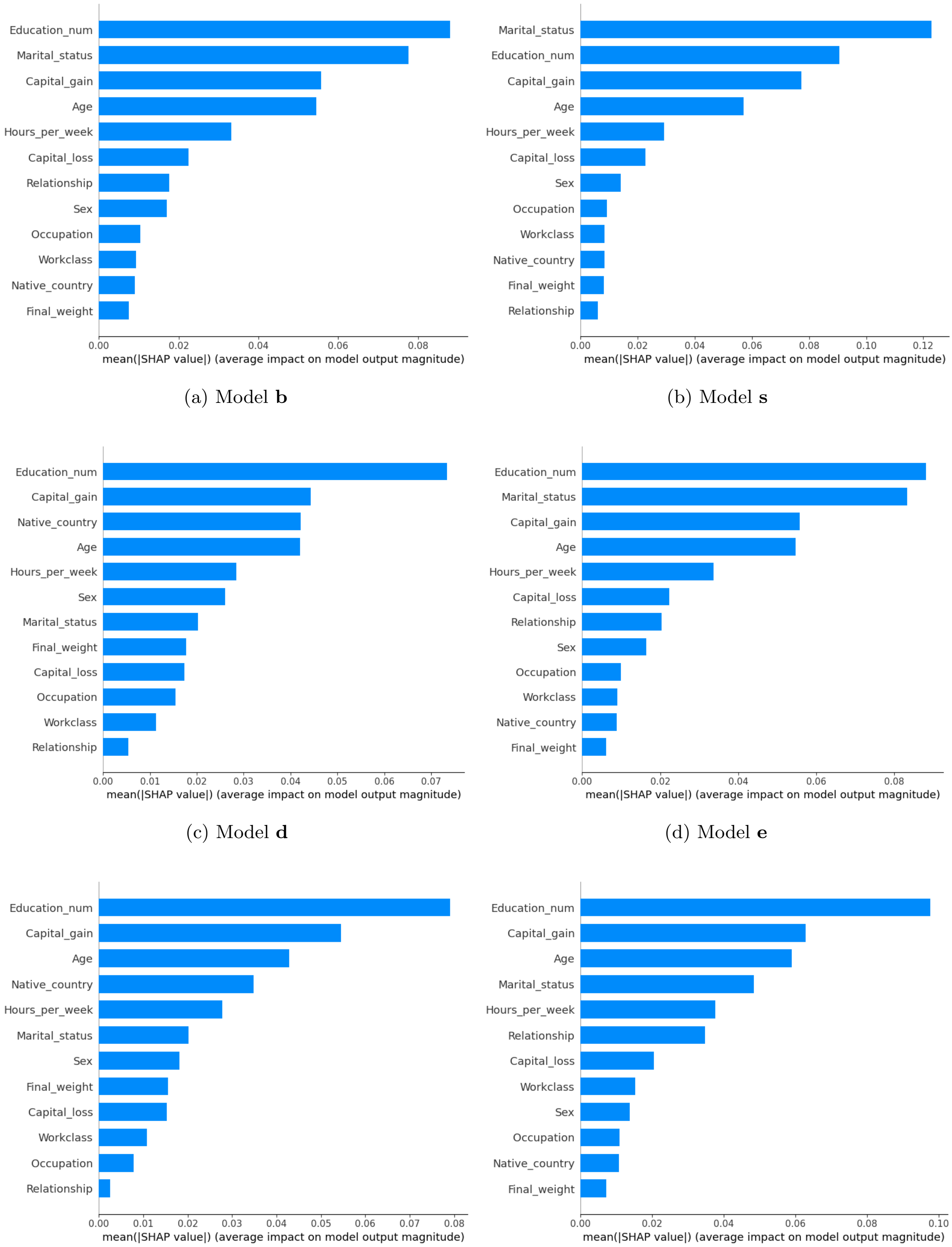
In the figures above we again see that the accuracy follows a similar shape for each λ_{eo}. The fitted line $(-0.94x + 0.20)$, however, seems to have more error.

This may be due to the results we have seen in Section 5.1, where the accuracy has a rapid increase close to the optimal point of the fairness measures. Smaller intervals in the grid search may reduce the error substantially.

We find similar 'valleys' to the results of the Adult dataset in Figures 10b and 10c that follow the fitted line corresponding to the respective minima of the fairness measures. The lines are given by the equations $y = -1.01x - 1.38$, $y = -1.01x - 1.51$ for DP and EO respectively. Remarkably, the slopes of the lines are close to equal, only differing by about 0.0005%. However, the starting points are not the same nor sufficiently close. This means these almost parallel lines will not meet at λ values near 0, corresponding to the base model **b**. Nonetheless, similar to the Adult dataset results, we can find a decent starting point for a grid search by finding the point where the lines of accuracy and DP or EO meet.

5.4 Interpretation

In this subsection we compare the models on their post-hoc interpretation using SHAP, as described in Section 4.4.2. The results are summarised in the figures below:



(a) Model **b**

(b) Model **s**

(c) Model **d**

(d) Model **e**

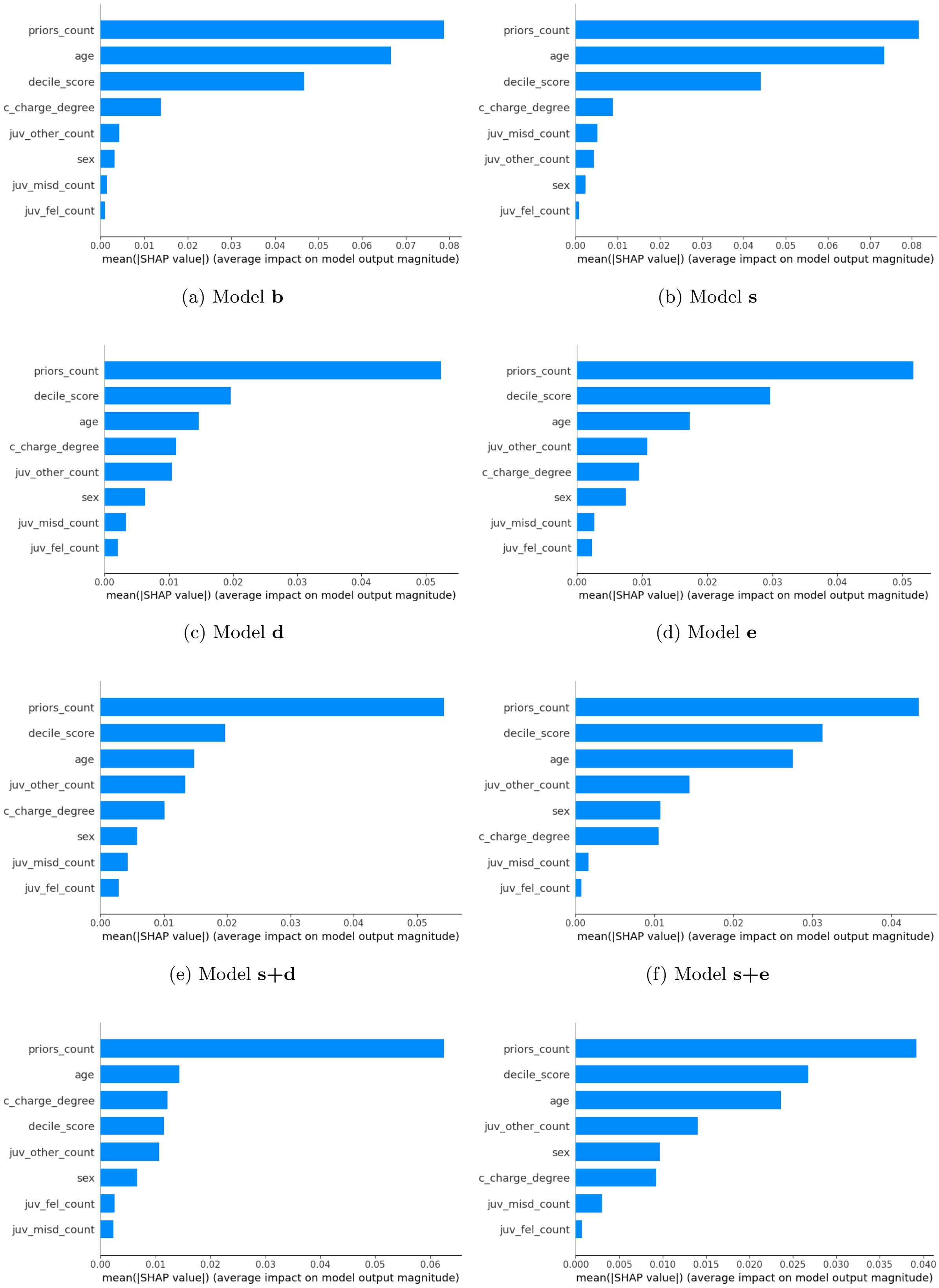
(e) Model **s+d**

(f) Model **s+e**

Figure 11: Shapley feature importances (Adult dataset)

For the Adult dataset, we find that the largest impacts on model output does not change considerably across all different models. The few differences we see are hard to attribute solely to the fairness method used. For example, model **s** switches its primary contributor from *Education number* to *Marital status*. However, both variables were already ranked highest among the variables. A larger difference would be models **d**, **s+d** and **s+e** attributing *Marital status* significantly lower than in the base model **b**. This may indicate that using *Marital status* as a predictor may lead to unfair bias in the NN. Similarly, we find that models **d** and **s+d** rank the variable *Final weight* higher than model **b**. This may suggest that *Final weight* is a more fair predictor. Furthermore, an interesting finding is that the variable *Native country* impacts models **d** and **s+d** more. This variable could be seen as a surrogate for the omitted variables *Race*. However, the models do still score better with respect to DP than model **b**.

As stated before in Section 5.1, we find an optimal λ of 0 for models **eo** and **s+eo**. This corresponds to the base model, which in turn means that the Shapley feature attributions are equal for these models.



(a) Model **b**

(b) Model **s**

(c) Model **d**

(d) Model **e**

(e) Model **s+d**

(f) Model **s+e**

(g) Model **eo**

(h) Model **s+eo**

Figure 12: Shapley feature importances (Compas dataset)

In the figures above, we find the Shapley interpretation results for the Compas dataset. We see that

Priors count contributes most to all models' output. The general ranking of output contributions is not changed much across the different models. However, we note that the magnitude of attributions of *Priors count*, *Decile score* and *Age* seem to be distributed more to the other variables in models **d**, **e**, **s+d**, **s+e** and **s+eo**. Those variables are also the most correlated with the sensitive feature *Race*. As such, they may function as a proxy for *Race*, and the models involving a λ seem to decrease their attribution.

Furthermore, we note that *Sex* is more prominent in attribution in all models except **b** and **s**. This variable is a sensitive feature which we have not taken into account in the models. Therefore this result may not be optimal for those that wish to reduce bias with respect to all sensitive features. A future research area might be the inclusion of two or many more sensitive features into our proposed models. The models are structured such that they are capable of accomplishing this.

6 Conclusion

In this paper we have applied several techniques to take fairness into account in ML. Specifically, we used two pre-processing techniques named SMOTENC and LFR on a benchmark Neural Network. Furthermore, we have applied surrogate relaxation criteria in the loss function of a NN to create models that penalise unfairness during the learning phase. Afterwards, certain evaluation metrics have been used to compare the models with respect to the classical accuracy, as well as two fairness metrics.

The first pre-processing method, SMOTENC, is found to have no significant impact on the fairness measures used in evaluation. This is mainly due to the limitations of the technique itself, as it does not handle datasets with mainly categorical variables well. These datasets are not uncommon though, so SMOTENC may not be a fitting technique. Additionally, even though the bias to a sensitive feature is grounded in the data itself, SMOTENC does not target this bias. SMOTENC only reduces class imbalance by creating artificial data points. Therefore, specific pre-processing techniques to combat unfairness will likely perform better.

LFR, the second pre-processing method, did not work as expected for our purposes. The models trained with learned prototypes did not learn any meaningful relations between the predictors and target variables. It seems the LFR algorithm implemented by us lost all relevant information of the datasets. Furthermore, another limitation of LFR is its inability to use standard post-hoc interpretation methods such as Shapley. It misses a key component to fair ML, as the interpretation of prediction are invaluable to understanding the decisions of ML models. However, LFR is an interesting subject to pursue for future research. It tackles the aforementioned issue of SMOTENC, which does not specifically target bias in the dataset. LFR learns a fair representation of the data, which can then be used for any downstream task. Although the prototypes lack meaningful interpretation, the dataset is transformed to be intrinsically fair. Thus, combats bias as a pre-processing method which does not need to alter existing models, which is one of its main advantages.

All models were evaluated to a benchmark NN, which performed poorly in regards to fairness measures. Both DP and EO were introduced as surrogate relaxation constraints to implement a penalty for unfair learning in our NN. The two datasets showed different results for the models. Whereas the Adult dataset suggested that one surrogate fairness constraint does not guarantee the other, the Compas dataset showed that both fairness metrics decreased with either surrogate fairness constraint. However, the common thread is that the fairness metrics did in fact decrease using this method. We must note, however, that the upper-bound set by choosing a λ as proposed in Bendekgey and Sudderth (2021) did not follow our expectations. In the results we see a highly fluctuating function describing the relation between λ and the fairness measures. Optimal λ's can be found, but an exhaustive grid search must be done which does not even guarantee global minima. Nonetheless, the methods did accomplish their goals of reducing unfairness. We also found that we can select λ to make a trade-off between accuracy and its respective fairness measure. This allows us to make the method feasible in real life applications, as accuracy does not have to suffer too much while as the same time decreasing bias.

Lastly, the post-hoc interpretation method SHAP, using the Shapley value, did not show very significant differences in the impact of variables on the model output. This means that incorporating fairness techniques does not necessarily change the underlying valuation of features in a dataset. The small changes in attribution we did find may be attributed to the methods we used, but are not satisfactory in decreasing unfairness in ML by means of interpretation. However, the Shapley method for interpretation itself proves to be a good way to gain intuition for how a model makes its prediction. The figures in Section 5.4 show the average impact on model output for the features, but individual prediction can also be expressed in its feature attributions. This would for example allow us to take edge cases and closer inspect the model's prediction for it. One could inspect whether a sensitive feature, in our case *Race*, played too much of a role in the prediction. A manual check could be made following the inspection of the model prediction of these cases to ensure fair decision making.

6.1 Limitations and Future Research

As explained above, both pre-processing methods did not work as expected. SMOTENC performed poorly due to its nature, while LFR needs more research to be implemented correctly. Another limitation of LFR is its large computation time. Global optima are not guaranteed, and the time complexity is largely dependent on the many matrix multiplications needed to find the optima. However, the idea of LFR is closely connected to the understanding that fairness arises from data. Being able to alter the data in such a way that fairness can be guaranteed is a good step in the right direction. Additionally, once the method is well behaved, the optimisation needs only be done once.

Furthermore, the other models suffered from the problem of hyper-parameter selection. This is done by an exhaustive grid search that does not guarantee global minima. However, as explained in Section 5.3, the trade-off between accuracy and several fairness metrics in the objective function showed interesting behaviour. Minima follow a linear pattern that should meet at certain points. It may be interesting to research how these functions behave on larger grids and intersections, as well as incorporating more fairness metrics in the objective functions to search on a larger-dimension grid.

Another point to discuss is the implementation of other fairness measures and different models. Bendekgey and Sudderth (2021) proposed a general method of incorporating fairness measures as surrogate relaxation constraints into an objective functions. Other surrogate relaxation constraints, R_g as in Equation (25), can be constructed in similar ways to decrease other fairness measures. Depending on the application, these other fairness measure may prove to be more valuable. Additionally, many models use objective functions to learn. These different models may pose to be an interesting subject of research as to see which of the popular (or unpopular) ML methods perform well with the implementation of Bendekgey and Sudderth (2021).

To conclude, even though our research had limitations, researching these methods have potential to make ML more interpretable and fair for its user.

7 References

[1] Julia Angwin, Surya Mattu, Lauren Kirchner, and ProPublica. *Machine Bias.* 2016.

[2] Samuel Axon. *Here's why Apple believes it's an AI leader- and why it says critics have it all wrong.* 2020.

[3] Henry C Bendekgey and Erik Sudderth. "Scalable and Stable Surrogates for Flexible Classifiers with Fairness Constraints". In: *Advances in Neural Information Processing Systems.* Ed. by M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan. Vol. 34. Curran Associates, Inc., 2021, pp. 30023–30036. URL: `https://proceedings.neurips.cc/paper/2021/file/fc2e6a440b94f64831840137698021e1-Paper.pdf`.

[4] CBS. *Gender pay gap still narrowing.* May 2020. URL: `https://www.cbs.nl/en-gb/news/2020/18/gender-pay-gap-still-narrowing`.

[5] Nitesh Chawla, Kevin Bowyer, Lawrence Hall, and W. Kegelmeyer. "SMOTE: Synthetic Minority Over-sampling Technique". In: *J. Artif. Intell. Res. (JAIR)* 16 (June 2002), pp. 321–357. DOI: `10.1613/jair.953`.

[6] Equal Employment Opportunity Commission. *Uniform Guidelines on Employee Selection Procedures (1978).* 1978. URL: `https://www.govinfo.gov/content/pkg/CFR-2017-title29-vol4/xml/CFR-2017-title29-vol4-part1607.xml`.

[7] K. Dasaradhs. *A gentle introduction to math behind Neural Networks.* Oct. 2020. URL: `https://towardsdatascience.com/introduction-to-math-behind-neural-networks-e8b60dbbdeba`.

[8] Dheeru Dua and Casey Graff. *UCI Machine Learning Repository.* 2017. URL: `http://archive.ics.uci.edu/ml`.

[9] Imme Ebert-Uphoff, Ryan Lagerquist, Kyle Hilburn, Yoonjin Lee, Katherine Haynes, Jason Stock, Christina Kumler, and Jebb Q. Stewart. *CIRA Guide to Custom Loss Functions for Neural Networks in Environmental Sciences – Version 1.* 2021. DOI: `10.48550/ARXIV.2106.09757`. URL: `https://arxiv.org/abs/2106.09757`.

[10] Daniel Godoy. *Understanding binary cross-entropy / log loss: a visual explanation.* Feb. 2019. URL: `https://towardsdatascience.com/understanding-binary-cross-entropy-log-loss-a-visual-explanation-a3ac6025181a`.

[11] Moulin Hervé. *Fair division and collective welfare.* 2003.

[12] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. "Multilayer feedforward networks are universal approximators". In: *Neural Networks* 2.5 (1989), pp. 359–366. ISSN: 0893-6080. DOI: `https://doi.org/10.1016/0893-6080(89)90020-8`. URL: `https://www.sciencedirect.com/science/article/pii/0893608089900208`.

[13] Harkiran Kaur. *How Does Google Use Machine Learning?* 2019.

[14] R. Kohavi and F. Provost. "Glossary of terms. Machine Learning—Special Issue on Applications of Machine Learning and the Knowledge Discovery Process." In: (1998).

[15] Scott M Lundberg and Su-In Lee. "A Unified Approach to Interpreting Model Predictions". In: *Advances in Neural Information Processing Systems 30.* Ed. by I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett. Curran Associates, Inc., 2017, pp. 4765–4774. URL: `http://papers.nips.cc/paper/7062-a-unified-approach-to-interpreting-model-predictions.pdf`.

[16] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems.* Software available from tensorflow.org. 2015. URL: `https://www.tensorflow.org/`.

[17] Christoph Molnar. *Interpretable Machine Learning. A Guide for Making Black Box Models Explainable.* 2nd ed. 2022. URL: `https://christophm.github.io/interpretable-ml-book`.

[18] Dana Pessach and Erez Shmueli. *Algorithmic Fairness*. 2020. DOI: 10.48550/ARXIV.2001.09784. URL: https://arxiv.org/abs/2001.09784.

[19] Gajane Pratik and Pechenizkiy Mykola. *On Formalizing Fairness in Prediction with Machine Learning*. 2018. arXiv: 1710.03184 [cs.LG].

[20] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. *"Why Should I Trust You?": Explaining the Predictions of Any Classifier*. 2016. arXiv: 1602.04938 [cs.LG].

[21] Linda F. Wightman. *Law School Admission Council National Longitudinal Study*. 1998.

[22] Rich Zemel, Yu Wu, Kevin Swersky, Toni Pitassi, and Cynthia Dwork. "Learning Fair Representations". In: *Proceedings of the 30th International Conference on Machine Learning*. Ed. by Sanjoy Dasgupta and David McAllester. Vol. 28. Proceedings of Machine Learning Research 3. Atlanta, Georgia, USA: PMLR, 17–19 Jun 2013, pp. 325–333. URL: https://proceedings.mlr.press/v28/zemel13.html.