Enhancing Fairness in Supervised Machine Learning

Bita Omidi

Thesis submitted to the University of Ottawa
in partial Fulfillment of the requirements for the
Master of Science Systems Science

School of Electrical Engineering and Computer Science
Faculty of Engineering
University of Ottawa

Enhancing Fairness in Supervised Machine Learning

Bita Omidi

Master of Science
Systems Science
School of Electrical Engineering and Computer Science
University of Ottawa

Work is done under the supervision of Dr. Marina Sokolova and Dr. Tet Yeap

Abstract

The increasing influence of machine learning algorithms and artificial intelligence on the high-impact domains of decision making has led to an increasing concern for the ethical and legal challenges posed by the sensitive data-driven systems. Machine learning can identify the statistical patterns in the historically collected big data generated by a huge number of instances that might be affected by human and structural biases. ML algorithms have the potential to amplify these inequities. Lately, there have been several attempts to reduce bias in artificial intelligence in order to maintain fairness in machine learning projects. These methods fall under three categories of pre-processing, in-processing and post-processing techniques. There are at least 21 notations of fairness in the recent literature, which not only provide different measurement methods of fairness but also lead to completely different concepts. It is worth mentioning that, it is impossible to satisfy all of the definitions of fairness at the same time and some of them are incompatible with each other. As a result, it is important to choose a fairness definition that need to be satisfied according to the context that we are working on.

The current study investigates some of the most common definitions and metrics for fairness introduced by researchers to compare three of the proposed de-biasing techniques regarding their effects on the performance and fairness measures through empirical experiments on four different datasets. The de-biasing methods include the “Reweighing Algorithm”, “Adversarial De-biasing Method”, the “Reject Option Classification Method” performed on the classification tasks of “Survival of patients with heart failure”(Heart Failure Dataset), “Prediction of hospital readmission among diabetes patients” (Diabetes Dataset), “Credit classification of bank account holders” (German Credit Dataset), and “The COVID19 related anxiety level classification of Canadians” (CAMH Dataset).

Findings show that the adversarial de-biasing in-processing method can be the best technique for mitigating bias working with the deep learning classifiers when we are capable of changing the classification process. This method has not led to a considerable reduction of accuracy except for the CAMH dataset. The “Reject Option Classification” that is a post-processing method, causes the most deterioration of prediction accuracy in all datasets. On the other hand, this method has the best performance in alleviating the bias generated through the classification process. The “Reweighing Algorithm” as a pre-processing technique does not cause a considerable reduction in the accuracy and is capable of reducing bias in classification tasks, although its performance is not as strong as the Reject Option classifier.

Acknowledgments

I would like to express my gratitude toward my professors, Dr. Marina Sokolova – my supervisor – and Dr. Tet Yeap – my co-supervisor – for their kind supports patience and excellent guidance through the path of conducting this thesis and for all the time and consideration, they put into directing my work. Last but not least, my love goes to my family, who, despite the distance, have supported me emotionally through this feat.

Table of Contents

List of Tables

List of Figures

1 Introduction

1.1 Motivation

The accelerating power of data mining and machine learning in the analysis and classification of big data and pattern recognition have led to a growing trend towards algorithmic decision making in many facets of our lives. Artificial intelligence (AI) can positively influence the quality of education that we receive (Bosch et al., 2016; Holstein, McLaren, & Aleven, 2018). AI plays a subtle role in the users' experience on social media and the posts and advertisements they observe every day (Alvarado & Waern, 2018; Bucher, 2017). Machine Learning (ML) systems have penetrated the recruitment process of companies, deciding which candidate is a suitable fit for a position using behavioral science and AI technology ("HireVue: Video Interview Software & Recruitment Platform," 2020; "Talent Matching Platform,"). Algorithmic decision-making has even affected the judicial systems in the recent years with software applications trying to predict future criminals or recidivism potential (Angwin, Larson, Mattu, & Kirchner, 2016; Chouldechova, 2017; Lum & Isaac, 2016). Also, the widespread adoption of Electronic Health Records (EHR) and technological advances in digital medical imaging in large health systems provide the possibility of rapid access to longitudinal clinical care data. It has provided a great opportunity to empower diagnosis, treatment selection, and health system efficiency (Chowriappa, Dua, & Todorov, 2014; Esteva et al., 2017).

This significant development in ML has recently led to an increasing concern for the ethical and legal challenges posed by the sensitive data-driven systems. Machine learning can identify the statistical patterns in the historically collected big data generated by a vast number of instances from different units worldwide. Since the collected data might be affected by human and structural biases, these inequities might be reflected by the trained models and may, consequently, result in a discriminatory decision-making. To the best of our knowledge, there is no comprehensive study to compare the different metrics and de-biasing methods that have been introduced to come across a benchmark for the suitable metrics and practices in each application. Therefore, this existing gap within the ML studies surged the

concern for conducting this research on implementing different metrics and de-biasing methods on various datasets and comparing them.

1.2 Objectives

Our purpose is to make the methods compatible with different datasets, which have different types of inputs, sensitive attributes and classification objectives. We will implement them to demonstrate if the models which were made through using the de-biasing techniques have acceptable accuracy and performance. At the same time, they follow the definitions of fairness and show less demographic bias against the protected group in comparison with the models trained without the de-biasing process. Regarding acceptable accuracy, we want the difference between the accuracy of training with and without de-biasing to be statistically insignificant or in favor of the de-biasing method.

This study investigates some of the most common definitions and metrics for fairness introduced by researchers and discusses what each of these definitions stipulates and represents. We compare three of the proposed de-biasing techniques regarding their effects on the performance and fairness measures through empirical experiments on four different datasets. We illustrate how the raw datasets might be biased against some demographic groups and how applying machine learning algorithms can magnify this bias.

A range of common fairness metrics and machine learning schemes will be deployed and compared on different datasets to determine which combination of fairness definition and bias mitigation approach with machine learning methods are more efficient in different situations and classification objectives. Our focus is on the group fairness approaches, although we use the Theil index metric to measure the impact of these methods on individual fairness.

1.3 Approach and Scope

In this study, we use three different de-biasing methods, including the “Reweighing Algorithm”(Faisal Kamiran & Calders, 2012), “Adversarial De-biasing Method”(Zhang, Lemoine, & Mitchell, 2018), and the “Reject Option Classification Method”(F. Kamiran, Karim, & Zhang, 2012). These three methods are well researched in the literature, have been

applied to different datasets, and there are software toolkits available for them. These methods can represent the different types of group fairness methods introduced in the literature. Reweighing is a pre-processing solution based on removing the dependency between the protected attribute and the class label. Adversarial de-biasing is an in-processing method for mitigating undesired biases related to the demographic groups in the training data by including a variable for the group of interest and simultaneously learning a predictor and an adversary. Reject option classification is a post-processing technique that gives favorable outcomes to unprivileged groups and unfavorable results to privileged groups in a confidence band around the decision boundary.

In order to perform de-biasing methods, we need to divide the classification labels into positive and negative. To avoid complexity in this comparative study, we approach the experiments as binary supervised classification tasks. Fairness enhancing techniques classify the protected attributes into privileged and unprivileged groups. We have limited our experiments on the cases i.e., the protected attributes are binary features, and all unprivileged classes are grouped together into a single value.

The bias mitigating approaches are applied on four different datasets to perform the tasks of "Survival of patients with heart failure", "Prediction of hospital readmission among diabetes patients", "Credit classification of bank account holders", and "The COVID19 related anxiety level classification of Canadians". We choose these datasets to cover a variety of classification applications. To provide with a better comparative experiment, the selected datasets include both the ones that have been considered in the previous fairness studies and the ones that have not been a matter of attention for the possibility of bias.

Five different fairness metrics are considered in this study: "Statistical Parity difference"(Verma & Rubin, 2018), "Equal Opportunity Difference"(Hardt, Price, & Srebro, 2016b), "Average Odds Difference"(Hardt et al., 2016b), "Disparate Impact Ratio"(Feldman, Friedler, Moeller, Scheidegger, & Venkatasubramanian, 2015), and "Theil index"(Speicher et al., 2018).

The experiments are done in two stages: before applying the de-biasing process and after applying the de-biasing process. Each time we are going to do the classification task using different machine learning algorithms, including "Naïve Bayes", "Logistic

Regression", "Neural Network", "Decision Trees", and "Support Vectors Machine". In the end, the results are compared based on their accuracy and fairness measures. The demographic data, including race, age, and gender are considered the protected attributes for the de-biasing process.

1.4 Contributions

This research shows that the Adversarial De-biasing method is the best technique for mitigating bias working with the deep learning classifiers as an in-processing method, which directly changes the classification algorithm and does not work with the initial data or the final classification results. Using the Adversarial de-biasing process prevents the classifier from increasing the bias in favor of the opposite protected group after de-biasing. Whereas using the Reweighing and Reject Option methods, there is a high possibility that fairness measures decrease in the opposite direction.

Reject Option Classification is the most powerful method in alleviating the bias generated through the classification process. However, it causes the most deterioration of prediction accuracy in all datasets. Reweighing Algorithm is a good technique in reducing bias in classification tasks, but its performance is not as strong as the Adversarial De-biasing. The Reweighing algorithm does not have a good performance in reducing the bias with the Neural Network and the SVM classifiers. As these machine-learning schemas work based on a weight adjustment procedure, setting initial weights will not have a strong influence on the final results with these methods.

Our results show that the statistical parity, equal opportunity, and average odds are not incompatible with each other. The proposed de-biasing methods can simultaneously reduce these group fairness measures depending on the dataset characteristics of the used classifiers.

1.5 Outline

The thesis is structured as follows:

- **Chapter 2 – Background and Related work:** This chapter introduces the machine learning models and the common performance metrics utilized in this research. Then we will discuss the fairness measures and bias mitigation methods proposed in the literature and mention some of the previous endeavors that have been made in the field of fairness in machine learning.
- **Chapter 3 – Methodology:** This chapter provides a detailed explanation for the data pre-processing techniques, fairness mitigation procedure, and fairness measurement techniques used in this research and introduces the background framework for each of the applied methods.
- **Chapter 4 – Experiments and Evaluation:** This chapter focuses on the results and evaluation of experiments performed on the datasets. In this chapter, we show the fairness and performance measures of each dataset before and after the de-biasing process and compare the results between different machine learning algorithms and various de-biasing methods.
- **Chapter 5 – Analysis of the Results:** This chapter summarizes the results gained in the previous chapter
- **Chapter 6 – Conclusions and Future Work:** Gives a conclusion over the analysis of the results and discuss the potential future work that can be done on this subject.

2 Background and Related Work

2.1 Machine Learning Approaches

Data mining is the process of extracting and discovering potentially useful patterns from a large amount of data(Ngo, 2011). Machine learning provides the technical basis of data mining. It involves using different learning algorithms (learning schemes) and statistical methods to build a model for a set of data that automatically improves through experience and can predict future outcomes(Ngo, 2011).

According to Ngo (2011), the inputs of a ML system can be divided into three different forms of "concept", "instances," and "attributes". The thing that will be learned is called a "concept," and the output produced by a learning scheme is the "concept description". The information that the learner is given is called a set of "instances". Each instance is characterized by the values of "attributes" that measure different aspects of the instance. There are many kinds of attributes, including numeric, nominal, binary, time series, image, text, audio, and video. However, typical data-mining schemes deal only with numeric and nominal or categorical ones (Ngo, 2011).

2.1.1 Learning Algorithms

The types of machine learning algorithms differ in their approach, the kind of data they input and output, and the type of task or problem they are intended to solve. Machine Learning tasks can be divided into four main categories according to the amount and type of supervision they receive during training, including supervised, unsupervised, semi-supervised, and reinforcement learning (Géron, 2017).

- **Supervised**

In supervised learning, algorithms build a mathematical model from a set of data that contains both the inputs and the desired outputs (the classification labels). Training happens through a stage of an iterative process to get the most optimized objective function. Objectives function is the mathematical function that maps the value of the features to the desired outputs. The loss function or cost function are examples of objective functions that

we try to minimize through classification process. The optimal objective function lets the algorithm to predict the output associated with the new data ("Machine learning,"). According to Mitchell (1997):" An algorithm that improves the accuracy of its outputs or predictions over time is said to have learned to perform that task."

Supervised learning algorithms include classification and regression. Classification algorithms are used for a discrete, limited set of outputs. In contrast, regression algorithms are used when the outcomes have continuous numerical values within a range.

Classification: Classification learning is considered supervised learning as the data sets are provided with their actual outcomes. According to (Sokolova, Lapalme, & management, 2009), considering the categorical labels when data entries have to be assigned to predefined classes, classification tasks are divided into four categories:

1. Binary: When the input has to be classified only to one of the two non-overlapping classes.
2. Multi-class: When the input is to be classified to one of some non-overlapping classes.
3. Multi-labelled: When the input is to be classified into some of the non-overlapping classes.
4. Hierarchical: The input is to be classified into one of the classes, which are divided into subclasses or grouped into superclasses.

- **Unsupervised**

Unsupervised learning algorithms use a set of training data, which only have inputs and are not labeled with the desired output. Therefore, these algorithms identify commonalities in the data and provide a better understanding of the data based on the relationship among the variables (Géron, 2017).

- **Semi-supervised**

In the case of semi-supervised learning algorithms, some of the training examples are missing training labels, but they, nevertheless, can be used to improve the quality of a model (Ngo, 2011).

- **Reinforcement Learning**

During the reinforcement learning process, the algorithm, called the agent, continuously iteratively learns from the environment. It learns from its experiences of the environment until it explores the full range of possible states. It allows machines and software agents to automatically determine the ideal behavior within a specific context in order to maximize its performance. Simple reward feedback is required for the agent to learn its behavior, known as the reinforcement signal. Reinforcement learning algorithms are used in autonomous vehicles or in learning to play a game against a human opponent (Géron, 2017).

2.1.2 Decision Trees

Decision Trees are a group of machine learning algorithms that can be used for classification and regression. They are of the most common learning methods that are very powerful and capable of fitting complex datasets. Decision Tree algorithms are based on a divide-and-conquer approach to the classification problems (Witten, Frank, Hall, & Pal, 2016). A Decision Tree is made by the continuous process of splitting the dataset on the attributes as best as possible into different classes until a specific stop criterion is reached. In Decision Trees, observations on items are shown in branches, and the conclusions of observations are shown in the nodes. There are three different kinds of nodes: the root nodes that indicate the start of the decision process and have no incoming edges, inner nodes which have exactly one incoming and at least two outgoing edges, and the end nodes (or the leaves). Whenever the target classification label takes discrete values, the Decision Tree is called a *classification tree,* and whenever it takes continuous values, it is called a *regression tree.* One of the merits of using Decision Trees is that they require little data pre-processing, and there is no need for data scaling or data centering (Géron, 2017).

Training a Decision Tree; generally, we try to have a less complicated and more comprehensive tree. The complexity of a Decision Tree can be controlled by using stopping criteria and pruning methods (Rokach & Maimon, 2005). As Rokach and Maimon (2005) mentioned, there are four different metrics to measure the complexity of a Decision Tree: The total number of nodes, the total number of leaves, the depth of the tree, and the number of attributes used.

The algorithms that are used to build a Decision Tree from a dataset are called the inducers. Typically, the objective of these algorithms is to find the optimal Decision Tree by minimizing the generalization error, considering the minimum number of nodes and the minimum tree dept.

Notable Decision Tree algorithms are ID3 (Iterative Dichotomiser 3)(Quinlan, 1986), C4.5 (Salzberg, 1994), CART (Classification And Regression Tree)(Breiman, Friedman, Olshen, & Stone, 1984), Chi-square automatic interaction detection (CHAID)(Kass, 1980), and QUEST (Loh & Shih, 1997).

In this research, we use the CART algorithm. This algorithm first uses a single feature k and a threshold t_k to split the training set into two subsets. It searches for a pair of (k, t_k) which produces the purest node. A node is pure if all the training instances it applies to belong to the same class. The same method is repeated for the subsets recursively, and it stops when it reaches the maximum depth (one of the hyperparameters) or when it cannot find a split, which reduces impurity. This algorithm tries to minimize the following cost function (Géron, 2017).

Equation 1 – Cost function in the CART algorithm

$$J(k, t_k) = \frac{m_1}{m} G_1 + \frac{m_2}{m} G_2 \qquad (1)$$

Where "G_1" and "G_2" are the impurity of the first and second subsets, and "m_1" and "m_2" are the number of instances in each subset.

Decision Tree algorithms work top-down in that they choose the best variable at each stage that can split the data set on a specific attribute. Different inducers use different criteria to find the best variable. In this work, we use Gini impurity (Equation 2) or Entropy (Equation 3) as the impurity criteria, which are calculated with the following equations (Géron, 2017).

Equation 2) Gini impurity

$$G_i = 1 - \sum_{k=1}^{n} {p_{i,k}}^2 \quad (2)$$

$p_{i,k}$ is the ratio of class k instances among the training instances in the i^{th} node.

Equation 3) Entropy

$$H_i = -\sum_{k=1}^{n} p_{i,k} \log_2(p_{i,k}) \quad (3)$$

2.1.3 Bayesian Networks

Bayesian Networks are a group of classification learning methods, which are based on assuming conditional independence between every pair of features in a data set. This assumption is the main reason for calling these methods "naïve" since conditional dependency is never a case in the real world. Surprisingly, this method usually performs better than expected and is considered the baseline for assessing the performance of the other learning algorithms. Bayes theorem provides a way of calculating posterior probability P(c|x) from P(c), P(x), and P(x|c) based on the Equation 4.

Equation 4) Posterior Probability in Bayesian Networks

$$p(c|x) = \frac{p(x|c) \times p(c)}{p(x)} \quad (4)$$

When: P(c|x) is the posterior probability of class (c, target) given predictor (x, attributes), P(c) is the prior probability of the class, P(x|c) is the likelihood which is the probability of predictor given class, and P(x) is the prior probability of predictor.

There are different kinds of Bayesian Network including Gaussian Naive Bayes, Bernoulli Naive Bayes, and Multinomial Naive Bayes, which their main difference is by their assumption on the distribution of $p(x_i|y)$

In our work, we try both the Gaussian and Bernoulli Naïve Bayes classifier on the datasets and use the one that shows a better performance regarding the classification accuracy.

2.1.4 Logistic Regression

Logistic Regression (or Logit Regression) is a classification method based on applying the Linear Regression technique (Sammut & Webb, 2017). Similar to the linear regression method, it assumes a linear relationship between the features and computes the weighted sum of the features plus a bias term. In this method, the result is not used directly, but it calculates the logistic of the result. When w in the weight of features, and the bias term is b, Logistic Regression model estimated probability is as follows.

Equation 5) Labels probability in Logistic Regression

$$\hat{p} = \sigma(w^T x + b) \quad (5)$$

Logistic function σ is calculated using Equation 6.

Equation 6) Logistic function

$$\sigma(z) = \frac{1}{1 + \exp(-z)} \quad (6)$$

Once the Logistic Regression model has estimated the probability $\hat{p}$, the classification prediction is made easily considering the following rule (Equation 7).

Equation 7) Classification prediction in Logistic Regression

$$\hat{y} = \begin{cases} 0 & if \ \ \hat{p} < 0.5 \\ 1 & if \ \ \hat{p} \geq 0.5 \end{cases} \quad (7)$$

The objective of training a classification model is to minimize the difference between the actual and predicted results. In this regard, the Logistic Regression uses the Cross-Entropy Loss (log loss) function. For a single unit, it is calculated from Equation 8.

Equation 8) Cross-Entropy loss function

$$L(\hat{y}, y) = -(\, y log\hat{y} + (1-y)\log(1-\hat{y})) \qquad (8)$$

Measuring how well you are doing an entire training set, the following cost function is calculated (Equation 9).

Equation 9) Logistic Regression cost function

$$J(w,b) = \frac{1}{m}\sum_{i=1}^{m} L\left(\hat{y}^{(i)}, y^{(i)}\right) = \; -\frac{1}{m}\sum_{i=1}^{m}[\, y^{(i)} log\hat{y}^{(i)} + \left(1-y^{(i)}\right)\log\left(1-\hat{y}^{(i)}\right)] \qquad (9)$$

During the training of the Logistic Regression model, we will try to find parameters w and b that minimize the overall costs function. As the cost function is convex, different optimization methods such as Gradient Descent is guaranteed to find the global minimum (Géron, 2017).

In order to deal with multiclass problems, we can calculate a separate loss for each class label per observation and sum the results. Another technique that can generalize the linear regression method to support multiple classes directly is called *Softmax Regression*, or *Multinomial Logistic Regression*

In our work, we use Scitkit-Learn solvers as the optimization methods, including "Liblinear"[1], "newton-cg"(Yu, Huang, & Lin, 2011) solver, and L-BFGS-B[2].

[1] https://www.csie.ntu.edu.tw/~cjlin/liblinear/
[2] http://users.iems.northwestern.edu/~nocedal/lbfgsb.html

2.1.5 Artificial Neural Networks (ANNs)

Artificial Neural Networks (ANN), often just called "Neural Networks" (NN), are a group of non-linear statistical data modeling methods, which at first were inspired by the brains and biological neurons' structures. ANNs are the base of deep learning. They are powerful, scalable, and capable of working with large complex machine learning tasks such as image recognition and speech recognition services.

ANNs are not new to the computation systems as they were first introduced by Warren McCulloch and Walter Pitts in (McCulloch & Pitts, 1943) back in 1943. Since then, the development of these techniques has faced plenty of fluctuation, and there have been some winter periods. Nowadays, thanks to the significant advances in computing power in both hardware and software, it is possible to train complex ANNs, and there is a huge quantity of data available to use in databases (Géron, 2017).

Every Neural Network consists of some nodes (neurons), weighted connections between the nodes, and a computation approach called *activation function* used to define each neuron's output. Different types of activation functions can be used with this technique. The most popular ones are as below (Géron, 2017).

Sigmoid or Logistic function: A common S-shaped curve that is continuous and differentiable. Its output values range from zero in $-\infty$ to one in $+\infty$. It is calculated based on Equation 6.

The hyperbolic tangent function (tanh): Similar to the sigmoid function, it is S-shaped, continuous, and differentiable, but the output ranges from -1 to +1. As a result, it is less centered around 0 and converges faster.

Equation 10) Hyperbolic tangent function

$$\tanh(z) = 2\sigma(2z) - 1 \qquad (10)$$

The Rectified Linear Unit function (ReLU): It is continuous but not differentiable at $z = 0$. It does not have a maximum output value, and its derivative is 0 when $z < 0$. In practice, it works very well.

Equation 11) Rectified Linear Unit function

$$ReLU(z) = \max(0, Z) \qquad (11)$$

NNs consist of different layers. The simplest kind of a Neural Network includes one *input layer* that receives information from external sources, such as attribute values of the input dataset. The output layer generates the output of the network and hidden layers that connect the input and the output layer with one another. The input value of each node in every layer is calculated by the sum of all incoming nodes multiplied by the respective weight of the interconnection between the nodes (Erb, 1993).

According to Singh and Chauhan (2009), Neural Networks can be divided into two main groups:

Feedforward Networks: In these networks, the flow of data moves only in the forward direction from the input layer to the output nodes. There is no feed or loop in the system.

Recurrent Networks: This kind of network can have the feedback option and reuses the data from the later stages to the earlier stages.

The simplified process of a Neural Network is as follows:

1. The input data is given to the network, it propagates through the layers, and the forward process produces the prediction.
2. The error between the predicted and the real output is calculated (cost function).
3. The Neural Network uses an optimization algorithm to adjust the weights in a way to reduce the cost function.
4. The forward process starts again, and it continues until the error rate is minimized.

Optimization algorithms are methods used for minimizing the value of cost function by adjusting the internal model parameters. Some of the most common optimization techniques in the deep learning frameworks are as follows. Stochastic Gradient Descent (SGD) (Schmidt, Le Roux, & Bach, 2017), Momentum (Polyak, 1964), Nesterov Accelerated Gradient (NAG) (Sutskever, Martens, Dahl, & Hinton, 2013), Adaptive gradient (AdaGrad) (Duchi, Hazan, & Singer, 2011), Root mean square prop (RMSprop) (Tieleman & Hinton, 2012), Adaptive moment estimation, (Adam) (Kingma & Ba, 2014), Nesterov and Adam optimizer (Nadam) (Dozat, 2015).

In our work, we use Adam optimizer with the Scikit Learn Neural Network classifier, and other parameters will be chosen separately for each dataset through the hyperparameters tuning.

2.1.6 Support Vector Machine (SVM)

SVM is a strong machine learning approach that can be used for linear, nonlinear, classification, regression, and even outliner detection (Géron, 2017). The fundamental idea behind this technique is that we try to separate the output classes with a hyperplane maximizing the margin (maximum distance between data points of both classes) (Steinwart & Christmann, 2008). Support Vector Machines create one or multiple hyperplanes in an n-dimensional space. The dimension of hyperplanes depends on the number of features. When there are two features, it is just a line, or when there are three features, it is a two-dimensional plane.

In order to tackle nonlinear datasets, one approach is to add more features such as a polynomial feature, or the other approach is to use kernel trick, mapping the n-dimensional input data to a higher-dimensional space, where the data can be separated linearly.

In our work, we use Scikit Learn SVM classifier, and different kernels are going to be tried during the hyperparameters tuning for each dataset.

2.1.7 Evaluating the Learning Model

The overall performance of a classifier is measured by the error rate of the classification process. In classification problems, the classifier predicts the class label of

each instance. If it is predicted correctly, it is considered a success; otherwise, it is an error. The fraction of errors made over the whole dataset is called the **error rate** (Witten et al., 2016).

A well-designed and reliable model is the one, which generalizes well on the new data. To measure an algorithm's generalization performance, we should obtain its performance on a holdout **test set**. Therefore, we split the data into two sets of the **training set** and the **test set**. The training data is used by one or more learning schemes to come up with classifiers. The test data is then used to calculate the error rate of the final, optimized method called the **generalization error** (Géron, 2017). To do this, the model built during the training phase is applied to each test instance, and the result, which is the predicted class for that instance, is compared to the real class value defined for it. Each of these sets of data must be chosen independently. In situations that we need to find the optimized hyper parameters for each classifier, we use a **validation set** in addition to the other two datasets. To be more specific, we split the training dataset into a smaller set of validation set and we train multiple machine learning schemes with different hyper parameters on this reduced training set and select the one which has the best performance on the validation set (Géron, 2017).

The holdout procedure: The holdout procedure is when a certain amount of the data set is held over for testing, and the remainder is used for training. It is common to hold out one-third of the data for testing and the remainder for training. We should ensure that random sampling is done, so each class is represented evenly in both the training and test data. This procedure is called **stratification** (Witten et al., 2016)**.**

Cross-validation: In cross-validation, we decide on a fixed number of folds or partitions of the data (suppose k). The data is then divided into k equal parts; each of them, in turn, is used for testing, and the rest is used for training. This process should be repeated k times in so that every part is used for testing once. Finally, the error estimates are averaged to obtain the overall error estimate (Witten et al., 2016).

Given a single fixed sample of data, the standard way to estimate the error rate of a learning technique is to use stratified ten-fold cross-validation. It means to ensure each fold has the right proportion of each class value.

2.1.7.1 Evaluation Metrics

In a binary classification with yes/no classes, a single prediction has the four different possible outcomes charted in a table called the confusion matrix (Table 1). The true positives (TP) and true negatives (TN) are correct classifications. A false positive (FP) is when the outcome is incorrectly predicted as yes (or positive) when it is actually no (negative). A false negative (FN) is when the outcome is incorrectly predicted as negative when it is actually positive.

Table 1) Confusion Matrix

		Predicted Class	
		No	Yes
Actual Class	No	true negatives (TN)	false positive (FP)
	Yes	false negative (FN)	true positives (TP)

The overall success rate or **accuracy** is the number of correct classifications divided by the total number of classifications:

Equation 12) Accuracy

$$\frac{TP + TN}{TP + TN + FP + FN} \tag{12}$$

Accuracy is a quick illustration of whether a model is being trained correctly and how it may perform generally. However, it does not give detailed information regarding its application to the problem. It does not do well when there is a severe class imbalance.

Precision helps when the costs of false positives are high and shows how accurate the positive predictions are. It is the proportion of predicted correct labels to the total number of actual positive labels. The best value is one, and the worst is zero.

Equation 13) Precision

$$\frac{TP}{TP + FP} \tag{13}$$

Recall helps when the cost of false negatives is high. It is the proportion of predicted correct labels to the total number of predicted positive labels. The best value is one, and the worst is zero.

Equation 14) Recall

$$\frac{TP}{TP + FN} \tag{14}$$

F1 (F measure) is an overall measure of a model's accuracy that combines precision and recall. It can be interpreted as weighted precision and recall. F1 reaches its best and worst values at one and zero, respectively.

Equation 15) F measure

$$F1 = 2 \times \frac{precision \times recall}{precision + recall} \tag{15}$$

AUC (Area under the ROC Curve), which sometimes is referred to as the balanced accuracy, depicts the classifier's ability to avoid false classification (Sokolova et al., 2009). It is more robust than the accuracy in class imbalanced situations as it considers the class distribution. The best value is one, and the worst is zero.

Equation 16) AUC

$$AUC = \frac{1}{2}\left(\frac{TP}{TP + FN} + \frac{TN}{TN + FP}\right) \tag{16}$$

2.2 Bias and Fairness in Machine Learning

With the increasing influence of ML algorithms and AI on the high-impact domains of decision-making, such as university admission, hiring, mortgage lending, prison sentencing, and disease recognition, ethical AI and concepts like fairness and bias attract considerable attention in the recent literature. Algorithmic decision-making can play a beneficial role in everyday humans' lives, as machines do not get bored or tired and are capable of considering a large magnitude of features that humans are not capable of. On the other hand, machine-learning algorithms have the potential to amplify the social inequities and make "unfair" decisions (Mehrabi, Morstatter, Saxena, Lerman, & Galstyan, 2019). Populations that are influenced by the biased model behavior are called the *protected groups* (Rajkomar, Hardt, Howell, Corrado, & Chin, 2018). The protected groups can be divided into privileged and unprivileged groups. The target classification label is divided into positive and negative labels.

Skewed data are one of the most commonly known sources of bias. As an example of skewed data, we can consider a situation in which a group of instances is underrepresented in the training data. As a result, the model will not work accurately for that specific group(Beutel, Chen, Zhao, & Chi, 2017). The protected group can also be the one, which is over-represented among a particular category.

Rajkomar et al. (2018) explain how model design, biases in datasets, and the interactions of model predictions with clinicians and patients may deteriorate health care disparities. The writers argue that rather than guarding against the possibility of discrimination in the use of machine learning models in health care, we should use ML to enhance equity in healthcare systems. They discuss two hypothetical clinical applications that a machine learning system might harm the protected group. The writers discuss different scenarios that can cause bias utilizing ML modeling in healthcare. They categorize the root of these discriminations in various groups, including model design, training data, Biases in interaction with clinicians and patients. Unfortunately, the authors do not provide numerical evidence to support their claims.

One of the most famous examples of the impact of bias in machine learning is the Correctional Offender Management Profiling for Alternative Sanctions (COMPAS) software used in the courts in the United States to measure the risk of a defendant will recommit a crime. Judges used this software to decide whether to release an offender or keep them in prison (Saxena et al., 2019). According to research by Angwin et al. (2016), this software is biased against African-Americans. The researchers used the same benchmark as the creators of the algorithm; they obtained the risk scores assigned by the software to more than 10,000 people arrested in Broward County, Florida, in 2013 and 2014. They compared the risk score to the actual recidivism rates of defendants in the two years after. The scores correctly predicted recidivism 61% of the times but only 20% of the people who were classified as high risk actually recommitted a violent crime. The true positive rate of recidivism for the white and black defendants was almost the same (59% and 63%, respectively). However, the number of black defendants who misclassified as the high risk was two times more than the white defendants who misclassified as the high risk (false positive rates of 45% and 23% respectively). And the false negative rate of black defendants is 28% in comparison with the false negative rate of white defendant with 48%.

Lambrecht and Tucker (2019) did research on an algorithm that would deliver advertisements promoting jobs in Science, Technology, Engineering, and Math (STEM) fields. Although this algorithm was designed to show job advertisements in a gender-neutral way, it illustrated a discriminatory behavior towards women. Fewer females were likely to be shown with these job advertisements. Chen, Szolovits, and Ghassemi (2019) studies the impact of algorithmic bias in medical fields. This study depicts how machine-learning algorithms do not behave all the patients equally regarding their race, gender, and insurance payer type, which might have dangerous consequences for the unprivileged groups. There are two primary sources of unfairness in machine learning. One of them originated from bias in data, and the other is generated through the algorithm (Mehrabi et al., 2019).

2.2.1 Fairness Definition

Although there are many different notations of fairness introduced in recent literature, there is no explicit agreement over a particular definition. According to Saxena et al. (2019, p. 100), “ fairness is the absence of any bias based on an individual’s inherent or acquired

characteristics that are irrelevant in the particular context of decision-making". Narayanan (2018) describes there are at least 21 mathematical definitions of fairness in the literature, which are not only different in how they measure fairness, but also each of them can show a completely different outcome.

Kleinberg, Mullainathan, and Raghavan (2016) claim that theoretically, it is impossible to satisfy all the definitions of fairness simultaneously, and some of the fairness definitions are incompatible with each other. The writers prove that the two conditions of calibration and balancing the positive and negative classes are inherently incompatible. Calibration is satisfied when, for any predicted probability score, the probability of truly belonging to the positive class is equal for all protected groups (Verma & Rubin, 2018). Positive and negative class balance requires the average scores assigned to the instances which belong to the positive and negative classes to be the same among protected groups (Kleinberg et al., 2016).

All of the fairness metrics are expressed in terms of false-positive (FP), true positive (TP), false negative (FN), and true negative (TN) numbers. Narayanan (2018) claims if we pick any three of these metrics, they cannot be achieved at the same time while the prevalence is different between protected groups. As a result, it is crucial to choose fairness definitions that need to be satisfied according to the context that we are working on. Generally, the descriptions of fairness fall into two main groups of *individual fairness* and *group fairness.* The goal of individual fairness is that similar individuals must get similar outcomes, while in group fairness, each of the groups defined by the protected attribute must be treated equally. The fairness measures in this study are selected in a way that they cover both individual and group fairness, and there is no proved incompatibility between them in literature.

According to Friedler, Scheidegger, and Venkatasubramanian (2016), there are two different worldviews regarding the group fairness notions: *We are all equal (WAE),* and *what you see is what you get (WYSIWYG).* WAE states that all groups are equivalent with respect to the task, while based on WYSIWYG, the observation shows each protected group's actual ability. Demographic Parity and Disparate Impact are based on (WAE), Equality of Odds follows the (WYSIWYG) worldview, and equality of opportunity lies between the two

worldviews (Bellamy et al., 2018). In this study, we try to improve fairness through both worldviews, and the above-mentioned fairness metrics are selected for this research to achieve this goal.

Some of the most widely used definitions of fairness are as follows (We Suppose $y = 1$ and $A = 1$ show the positive label and the privileged group, respectively):

Statistical Parity (Demographic Parity): (equal acceptance rate, benchmarking) A predictor $\hat{Y}$ satisfies demographic parity if $P(\hat{Y} = 1|A = 0) = P(\hat{Y} = 1|A = 1)$. According to this definition, protected groups must have an equal opportunity of being assigned to the positive target label (Verma & Rubin, 2018).

Hardt et al. (2016b) proposed a new notion of fairness comparing the previous literature and discussed the advantage of considering the concepts of "Equal opportunity" and "Equalized odds" rather than just relying on the "Demographic parity". The writers offered a method through a post-processing algorithm to derive an unbiased predictor from a possibly discriminatory learned binary predictor.

Equality of Odds: According to Hardt, Price, and Srebro (2016a): " A predictor $\hat{Y}$ satisfies equalized odds with respect to protected attribute A and outcome Y; if $\hat{Y}$ and A are independent conditional on Y. $P(\hat{Y} = 1|A = 0, Y = y) = P(\hat{Y} = 1|A = 1, Y = y) \quad y \in \{0, 1\}$".

In other words, to satisfy the equality of odds, the true positive rate (TPR) and the false positive rate (FPR) should be the same for all of the protected group members (Mehrabi et al., 2019). This definition makes the training model to provide equality of bias and accuracy in all demographics.

Equal Opportunity: "A binary predictor $\hat{Y}$ satisfies equal opportunity with respect to A and Y if $P(\hat{Y} = 1|A = 0, Y = 1) = P(\hat{Y} = 1|A = 1, Y = 1)$".

This definition states that the true positive rate (TPR) must be the same between all protected groups (Hardt et al., 2016a).

Fairness through Unawareness (FTU): According to this definition, an algorithm is fair when none of the protected attributes are explicitly used in the training process. The main fault of this notion is that it does not take into account the possibility of subtle interconnection between the protected attributes and the other attributes that are not removed (Kusner, Loftus, Russell, & Silva, 2017).

Individual Fairness (IF): Individual fairness is satisfied when similar individuals get similar predictions (Kusner et al., 2017). The similarity should be carefully defined through a deep understanding of the domain.

Theil index: Speicher et al. (2018) proposed a fairness metric based on the inequality indices from economics. They argue that the overall individual fairness consists of between-group and within-group fairness. While all the other fairness measurements concentrated on the between-group component, this framework considers both levels of fairness. The researchers considered a family of inequality indices called generalized entropy indices, including the Coefficient of Variation and Theil index, as special cases. They consider tradeoffs between individual-level and group-level fairness. In that research, the focus was on the binary classification, although the researchers believe it is possible to extend their technique. In this paper, unfairness is defined as the inequality in benefits. When y_i is the correct label for the individual i and $\hat{y}_i$ is the predicted label, the benefit for the individual i is as according to Equation 17.

Equation 17) Benefit in Theil index

$$b_i = \hat{y}_i - y_i + 1 \qquad (17)$$

For a constant α, the generalized entropy of benefits $b_1, b_2, \dots, b_n$ with mean benefit μ is defined as follows (Equation 18).

Equation 18) generalized entropy of benefits in Theil index

$$\varepsilon^{\alpha}(b_1, b_2, \dots, b_n) = \frac{1}{n\alpha(\alpha - 1)} \sum_{i=1}^{n} \left[\left(\frac{b_i}{\mu} \right)^{\alpha} - 1 \right] \qquad (18)$$

Speicher et al. (2018) tested their framework on two common datasets in fairness research: the Adult income dataset with the “yearly income” as the classification label, and the ProPublica COMPAS dataset with the “crime recommitment” as the classification dataset. In both experiments, race and gender were considered as protected attributes. Logistic regression, support vector machine with RBF kernel (SVM), and random forest classifiers were applied during the classification experiment.

The fairness metrics that we use in this study include: “Statistical Parity difference”, “Equal Opportunity Difference”, “Average Odds Difference”, “Disparate Impact Ratio, and “Theil index.

2.3 Methods for Fair Machine Learning

There have been several attempts to reduce bias in artificial intelligence in order to maintain fairness in machine learning projects. Generally, these methods fall under three categories of *pre-processing, in processing,* and *post-processing.*

Pre-processing techniques try to eliminate the underlying discrimination by transforming the data. These techniques can be used just in case the modification of the training data is allowed (d'Alessandro, O'Neil, & LaGatta, 2017). The idea behind the pre-processing algorithms is that if the classifier is trained on balanced data, its predictions will be more balanced as well (Calders, Kamiran, & Pechenizkiy, 2009). These techniques are as follows. *Reweighing, Optimized pre-processing, Learning fair representations,* and *Disparate impact remover (Bellamy et al., 2018)*

In processing methods, try to modify the state of the art learning algorithms to mitigate discrimination during the model training process. If it is allowed to make changes into the training process, then in-processing can be used either by incorporating changes into the objective function or imposing a constraint(Mehrabi et al., 2019). *Adversarial de-biasing* and *Prejudice remover* are in-processing techniques.

Post-processing methods utilize a holdout set, which was not involved in the training process to improve the predictions' fairness (d'Alessandro et al., 2017). When there is no

possibility to make changes to the training data or the model training procedure, only post-processing techniques can be used. Equalized odds post-processing, Calibrated equalized odds, and Reject Option Classification are some of the post-processing algorithms.

We use three different methods (one from each group) and compare their results. These methods are reweighing (pre-processing), Adversarial De-biasing (in processing), and Reject Option Classification (post-processing).

2.3.1 Reweighing Algorithm

Reweighing (Faisal Kamiran & Calders, 2012) is a pre-processing solution based on removing the dependency between the protected attribute and the class label. This method was first introduced in Calders et al. (2009), concentrating on the bias resulting from discriminatory input datasets.

According to the reweighing approach, different weights are assigned to the instances so that instances from the unprivileged group with the desirable label will get higher weights compared to the ones labeled as an undesirable class. On the other hand, the objects in the privileged group with the desirable label will get lower weights than those that are undesirable.

In (Faisal Kamiran & Calders, 2012), the research is restricted to a binary protected attribute Z, and a binary classification problem with target attribute classes $Y = [y^+, y^-]$ where, y^+ and y^- are representations of desirable and undesirable classes, and the objects with $Z = 1$ and $Z = 0$ belong to the privileged and unprivileged community respectively. The dependency of the prediction of a classifier $\hat{y}$ and the protected attribute Z is defined by Equation 19.

Equation 19) The dependency of the prediction

$$p(\hat{Y} = y^+ | Z = 1) - p(\hat{Y} = y^+ | Z = 0) \qquad (19)$$

A positive dependency reflects that the object from the privileged community (Z=1) has a higher chance of being labeled as positive than the one from the deprived group (Z=0), and vice versa.

The researchers implemented the reweighing algorithm with five different classifiers on two datasets, including the Census Income and the Communities and Crimes datasets, available in the UCI ML repository. The classifiers that the researchers chose to use are Naïve Bayes Classifier (NBS), three nearest neighbor classifiers with, respectively, 1, 3, and 7 neighbors (IBk1, IBk3, and IBk7), and a Decision Tree learner. In that research, statistical parity difference is introduced as the discrimination measure, and two baseline approaches were used. In one of them, the classifiers were applied to the original dataset when no pre-processing was done on the data. In the other one, the experiments were repeated on the data while the sensitive attributes were removed. The proposed pre-processing technique is applied only on the training set for learning a classifier. The reweighing method had the best performance with the IBK7 method, which showed a reduction of discrimination from 19.45% to about seven percent. In comparison, the accuracy stayed almost the same, at about 82.5% with the Census Income dataset. With the IBK1 classifier, the discrimination reduced from about 18% to 16%. Using the J48 classifier, there was a reduction from 16 to 9% for Naïve Bayes from 16 to 10. Their accuracy stayed almost the same at 80.5%, 86%, and 83%, respectively.

The experimental results illustrated that with some trade-off for the accuracy, all the methods reduced discrimination from the datasets more efficiently than the simple approach of removing the sensitive attribute from the training data.

2.3.2 Adversarial De-Biasing

Zhang et al. (2018) presented a framework for mitigating undesired biases concerning demographic groups in the training data by including a variable for the group of interest and simultaneously learning a predictor and an adversary. A supervised deep learning task is considered in which the mission is to predict an output variable Y given an input variable X while remaining unbiased concerning some variable Z. We refer to Z as the protected variable. The objective is to maximize the predictor's ability to predict Y while minimizing the adversary's ability to predict Z. This may achieve more accurate predictions that exhibit

less evidence of stereotyping Z. This method was applied to two different machine learning problems including a standard supervised learning task and an analogy task. For the supervised learning problem, the researchers used the UCI Adult dataset and tried to enforce the "Equality of Odds" on a model for the task of income prediction. The experiment was done as a binary classification in which the income amount was divided into two classes of "more than $50K" and "less than $50K". The protected variable is a binary-valued variable for the two sexes (male and female). Their experiment showed that with a small reduction of the overall accuracy from 86% to 84.5%, the false negative rate and false positive rate are approximately equal between the two genders. In other words the difference of equality of odds between the sex subgroups is statistically insignificant after applying the adversarial de-biasing method.

The paper provides the theoretical guarantee that the method can enforce the constraints that are claimed across multiple definitions of fairness – "Demographic Parity", "Equality of Odds", and "Equality of Opportunity" –, regardless of the complexity of the predictor's model or the nature (discrete or continuous) of the predicted and protected variables in question. All models are trained using Adam optimizer for both predictor and adversary.

Using this method, we first introduce a primary model called the *predictor* to predict the output (e.g., hospital readmission or the ASD diagnosis) given the vector of attributes as the input. Then the output layer of the predictor network will be used as the input to an *adversary* network, which tries to predict the protected group (e.g., gender or race). The logic behind using this method is that if our primary model produces a representation of the attributes that encodes information about the protected group (sensitive attribute), an adversarial model can recover the sensitive data using that representation. In opposition, if the adversary fails to predict the sensitive data, it proves that our primary model is trained without any discrimination against the protected group.

The "adversarial de-biasing" framework proposed by Zhang, Lemoine, and Mitchell (2018) relies on adversarial training to reduce the latent bias from the trained model. To achieve this goal, multiple networks are trained in a way that one of them tries to predict the

target class while simultaneously prevent the other one from predicting the protected attribute. The satisfaction of fairness definitions is theoretically proven (Zhang et al., 2018).

This framework is applicable to various gradient-based models – such as Gradient Descent, AdaGrad (adaptive gradient algorithm), and Adam (Adaptive Moment Estimation) optimizers. What makes it remarkable is that the method is not dependent on the nature of the output and the protected group. It can be used both for the regression and the classification models, whether the protected variables are continuous or discrete.

Using an adversary for training a generative model for the first time was introduced by Goodfellow et al. (2014). In this framework (*adversarial nets),* two models are trained: a generative model (G), which tries to generate samples from the desired distribution, and a discriminative model (D), which estimates the probability of samples coming from the generative or the real data. The objective is to train the discriminative model to the best accuracy of labeling the training examples and the examples produced by the generative model while simultaneously train G in a way to deceive D as much as possible. This process corresponds to a minimax two-player game in game theory when a player tries to make the most optimized decision considering all of the opponent's best responses. The name "minimax" comes from minimizing the loss involved when the opponent selects the strategy that gives maximum loss, and two-player games are the ones that include exactly two competitors. Goodfellow et al. (2014) applied the framework to the unsupervised training process of image recognition. The framework is much easier to implement while both models are multilayer perceptron. (Goodfellow et al., 2014).

Beutel et al. (2017) used the adversarial training method to de-bias the latent representation inside the models with discrete output variables while accepting the model harms training accuracy. They discuss the capability to improve fairness, even when a small number of adversarial examples are available. They showed that the method works much more successfully if they have a balanced distribution over the sensitive attribute. The researchers tested the model on the Adult dataset from the UCI repository for the income-predicting task, using ReLU hidden layer and Adagrad optimizer for the training process.

2.3.3 Reject Option Classification

Reject option classification is a post-processing technique that gives favorable outcomes to unprivileged groups and unfavorable results to privileged groups in a confidence band around the decision boundary. Each time only one of the fairness metrics can be chosen for the optimization. This technique was introduced by F. Kamiran et al. (2012). The researchers developed this method based on the relationship between decision-theoretic concepts and the discrimination model proposed in Žliobaite, Kamiran, and Calders (2011). Based on this hypothesis, because the bias happens at the time of decision making, discriminatory decisions are often made close to the decision boundary. The Reject Option Classification (RO) recognizes the instances that need to be labeled differently from the others by using the posterior probabilities produced by probabilistic classifiers. This technique can be used on any probabilistic classifier, and it does not make any change to the learning algorithm or the pre-processing of the original data.

In order to measure the performance of this technique, the researchers did experiment on the “Adult” and the “Communities and Crimes” datasets, available on the UCI repository, with several classifiers including Naïve Bayes, Logistic Regression, k-nearest neighbor, and Decision Tree using single and multiple probabilistic classifiers. The “Adult” dataset includes 16,281 instances of people's demographic information, which their income is the class attribute. Gender is considered as a protected attribute, and female is the deprived or unprivileged group. The “Communities and Crimes” consists of the information of 1,994 United States criminals, predicting the total number of violent crimes per 100k population is the classification task, and black individuals are considered as the unprivileged group. In this research, the Statistical parity difference is utilized as the fairness metric.

According to the results shown by the researchers, there is a constant decrease in the accuracy of the classifiers with the reduction of discrimination.

2.3.4 Some Other Endeavors in De-Biasing Techniques

One of the other algorithmic techniques that learns a fair representation of data was introduced by Louizos, Swersky, Li, Welling, and Zemel (2016). This algorithm is called Variational Fair Autoencoder (VFAE), an extension of the semi-supervised variational

autoencoder. In this framework, “fairness” is defined as representations that are explicitly invariant with the specific characteristic of a dataset. The technique was performed on three datasets for fair classification scenarios.

Debiasing Variational Autoencoder (Amini, Soleimany, Schwarting, Bhatia, & Rus, 2019) is an algorithm that combines the original learning task with a variational autoencoder to learn the hidden structure within the dataset to rebalance the dataset. In other words, this model first learns which inputs come from the under-represented groups and change their importance weight during the training process.

Hashimoto, Srivastava, Namkoong, and Liang (2018) proposed an approach based on distributionally robust optimization (DRO), which essentially minimizes the worst-case loss of each group in the dataset. It directly modifies the optimization objective.

Lum and Johndrow (2016) discussed that merely removing the protected variable from the dataset cannot avoid discrimination, as there might exist other variables that have a hidden correlation with the protected variable. The researchers tried to make a framework for fair prediction from the statistical perspective, where a chain of conditional models is produced.

3 Methodology

In this study, we are going to deploy the proposed theoretical frameworks, including "Reweighing Algorithm", "Adversarial De-biasing Method", and "Reject Option Classification Method" to alleviate the demographical bias in the trained machine learning models in different settings. This research wants to understand whether the de-biasing techniques provide trained models with acceptable accuracy and fairness.

We compare the results of classification with and without de-biasing regarding *accuracy* and *fairness*. To evaluate the accuracy of the model, we consider a baseline performance when there is no de-biasing performed. We want the difference between the accuracy of training with and without de-biasing to be statistically insignificant or in favor of the de-biasing method

3.1 Data Pre-Processing

Machine Learning is not only about choosing the correct model. Preparing the data for the process in advance has an essential effect on the final result. In this research, the preprocessing stage is independent of the de-biasing process to provide a consistent comparison across algorithms. We can generally divide the pre-processing stage into three parts: data cleaning, feature engineering, and feature selection.

3.1.1 Data Cleaning

Data cleaning is the process of tackling the missing data. Missing data means there is no value recorded for a particular observation within the dataset. Three different techniques can be used to clean the data from missing values. We can remove the corresponding instances, remove the whole attribute when it has a high rate of missing values, or data imputation (replacing the missing values with other available data, for example, the mean or the median of the corresponding attribute) (Brown & Kros, 2003). It is worth mentioning that some algorithms like KNN and Naïve Bayes can deal with the missing values. Still, the ScikitLearn library of Python is not capable of performing these algorithms with missing values.

De-biasing methods show sensitivity to small changes in the datasets, as a result, in order to reduce noise in our datasets, in this work, we remove the instances, which include missing values, and in a few cases, the whole attributes that have a large number of missing values are removed.

3.1.2 Feature Engineering

Depending on the specific algorithms that we are going to use, and with an in-depth knowledge of the problem domain and the data structure, we need to make some changes to the data before using them for the classification process.

3.1.2.1 Feature Scaling

Feature scaling is a process to standardize the range of independent variables or features of data. Most of the machine learning algorithms cannot perform well when the attributes have very different scales. As a result, we need to do a normalization process before implementing the algorithm. Two of the most common scaling methods are *min-max* and *Standardization* techniques *(Géron, 2017).*

Min-max scaling shifts and rescale the data so that they range from 0 to 1.

Standardization subtracts the mean value and then divide them by the standard deviation so that the resulting attribute has unit variance. This method does not assign values within a limited range that cause problems for some algorithms such as Neural Networks.

3.1.2.2 Feature Encoding

We must transform the categorical values into numbers so that algorithms can handle those values. In this regard, we first do integer encoding (or label encoding), which means we assign an inter value to each category. The problem with this straightforward method is that the integer values have natural order that might be detected by the algorithm and have an effect on the final result. To solve this problem, we apply a one-hot encoding to the integer representation. Using this method, the integer-encoded variable is removed, and a new binary variable is added for each unique integer value(Géron, 2017).

For example, we can consider the race attribute, which includes three different values of “Caucasian”, “African American,” and “Others”. Performing a simple integer encoding will substitute them with three consecutive numbers. If we do one-hot encoding, the race feature will be removed, and three new binary features will be added (Table2).

Table 2) Example of Feature Encoding

Original	integer encoding	one-hot encoding		
race	race	Caucasian	African American	Other
Caucasian	0	1	0	0
African American	1	0	1	0
Other	2	0	0	1

3.1.3 Feature Selection

In a dataset with a large number of features, not all of them play a significant role in the analysis of the data. Besides, irrelevant attributes have a negative effect on most of the machine learning schemes. As a result, a common stage in pre-processing data is the attribute selection stage. The best way to select relevant attributes is manual, based on the deep understanding of the learning problem and the meaning of attributes. On the other hand, automatic feature selection can be useful if there are a vast number of attributes, and it would prevent any human bias during the selection process.

For selecting a suitable attribute subset, there are two fundamentally different approaches. “Filter method” means making an independent assessment based on the general characteristics of the data. And the “wrapper method” is the evaluation of the subset using a machine learning algorithm (Witten et al., 2016). There are several feature selection methods. The techniques that we are going to use are as follows:

“Variance Threshold”: It is the most straightforward baseline approach for feature selection, which removes all the zero-variance features. In other words, it removes all the features that have the same values in all samples (Hall, 1999).

"Univariate Selection": In this technique, the statistical tests are used to find the features that have the strongest correlation with the output variable. These tests include chi2, ANOVA F-value, and Mutual information (Ben-Bassat, 1982).

"Tree-based feature selection": In this method, tree-based estimators are used to compute impurity-based feature importance. It gives a score for each attribute of the dataset. The higher the score is, the more important or relevant feature toward the classification process (Duda, Hart, & Stork, 2001).

3.1.4 Imbalanced Classification Problems

An imbalanced classification problem is a classification problem where the distribution of instances across the known classes is biased or skewed. Imbalanced classifications pose a challenge for predictive modeling as most machine learning algorithms are designed to assume an equal number of examples for each class.

One approach to address the problem of class imbalance is to resample the training dataset randomly. The two main random resampling methods of an imbalanced dataset are to delete examples from the majority class, called under-sampling, and to duplicate examples from the minority class, called oversampling. Synthetic Minority Oversampling Technique or SMOTE is an improved technique of oversampling proposed by Chawla, Bowyer, Hall, and Kegelmeyer (2002). This method works based on creating synthetic examples of the minority class in the feature space rather than the data space. Each minority class is taken, and depending on the oversampling size, "k" nearest neighbors are joined by lines, and the new synthetic examples are introduced on those lines. The authors recommend using a combination of SMOTE and an under-sampling performs better than a simple method.

3.2 Reweighing Algorithm

Before we discuss the Reweighing method's details, we should introduce the notion of dependency, as mentioned in Faisal Kamiran and Calders (2012).

We supposed our labeled dataset D, consists of a set of attributes $X = [x_1, x_2, \dots, x_n]$ including the protected attribute $Z = [z = 1, z = 0]$ and the class labels $Y = [y^+, y^-]$. Where

$z = 1$ is the privileged group, and $z = 0$ is the unprivileged one, and "y^+" and "y^-" are the desirable and undesirable groups. Dependency between protected attributes and the class in D (dep D) is defined as the difference between the probability of being in the positive class between the tuples having $z = 1$ and those having $z = 0$ in D; that is:

Equation 20) Dependency between protected attributes in Reweighing

$$dep(D) := \frac{|\{x \in D | x(Z) = 1, x(Y) = y^+\}|}{|\{x \in D | x(Z) = 1\}|} - \frac{|\{x \in D | x(Z) = 0, x(Y) = y^+\}|}{|\{x \in D | x(Z) = 0\}|} \qquad (20)$$

A dependency greater than zero shows that the tuples with the (z=0) are less likely to be in the positive class than tuples with (z=1).

In the reweighing algorithm, our goal is to reduce the dependency to zero. In order to achieve this goal, we give different weights to the instances. For example, objects with $z = 0$ and y^+ get higher values than the objects with $z = 0$ andy^-, and objects with $z = 1$ and y^+ get higher values than the objects with $z = 1$ andy^-. According to these new weights, the new dataset is without dependency.

The weights are calculated based on some basic probability theory that if the dataset D is unbiased or, in other words, Z and Y are independent of each other, the expected probability of being from the deprived group and desirable label is according to Equation 21.

Equation 21) The expected probability of being from the deprived group and desirable label

$$P_{exp}(z = 0 \wedge y^+) := (z = 0) \times y^+ \qquad (21)$$

In reality, however, this probability might be different:

Equation 22) The real probability of being from the deprived group and desirable label

$$P_{act}(z = 0 \wedge y^+) := (z = 0) \wedge y^+ \qquad (22)$$

We will assign weights to $z = 0$ as Equation 23 and Equation 24 show.

Equation 23) New weights to z=0

$$W(z=0|y^{+}) := \frac{P_{exp}(z=0 \wedge y^{+})}{P_{act}(z=0 \wedge y^{+})} \qquad (23)$$

Equation 24) New weights to z=0

$$W(z=0|y^{-}) := \frac{P_{exp}(z=0 \wedge y^{-})}{P_{act}(z=0 \wedge y^{-})} \qquad (24)$$

The weights for $z = 1$ will be calculated in the same way. The array of the new weights are assigned to each individual training sample. These weights scale the loss function for each of the instances. In other words, higher weights force the classifier to put more emphasis on these points. On this balanced dataset, a discrimination-free classifier is learned.

3.3 Adversarial De-biasing Method

To implement the Adversarial De-biasing framework, we consider our model as a supervised deep-learning procedure where at first, a primary Neural Network model called the *predictor* is introduced, which is responsible for predicting the output label (y)given the vector of attributes as the input(x). The predictor will be trained by modifying the weights W to minimize some loss function $L_p(\hat{y}, y)$. To this end, we are going to perform a gradient-based optimization method.

After that, the predictor network's output layer will be used as the input to the adversary network, which tries to predict the protected group (demographic attribute). This structure is shown in figure1 as depicted in (Zhang et al., 2018). The adversary network will be trained using the same procedure as we suppose it has loss term $L_A(\hat{z}, z)$ that must be minimized by updating its weights U.

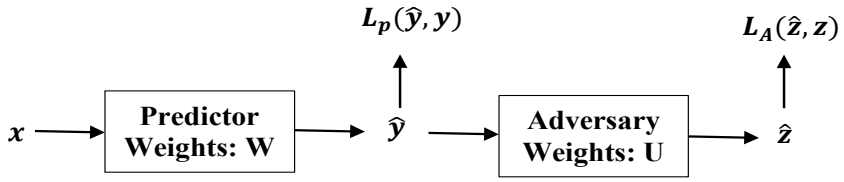


Figure 1: The architecture of the adversarial network (Borrowed from (Zhang et al., 2018)).

Considering the definition of fairness we are going to achieve, the adversary may have access to other inputs as follows.

For Statistical Parity, the adversary will only have access to the predicted label $\hat{y}$, for Equality of Odds, the adversary gets $\hat{y}$ and the true label Y, and for Equality of Opportunity on a given class of y, we restrict the training set of the adversary to training examples where the true label is the given class.

In each iteration, the adversary updates its weights just trying to minimize its loss L_A, while the predictor's weights will be updated according to the expression below.

Equation 25) The predictor's weights updating funtion

$$\nabla_w L_p - proj_{\nabla_w L_A} \nabla_w L_p - \alpha \nabla_w L_A \qquad (25)$$

$\nabla_w L_p$: The gradient of the predictor's loss

$proj_{\nabla_w L_A} \nabla_w L_p$: the projection of gradient of the predictor's loss onto the gradient of the adversary

$\nabla_w L_A$: the gradient of the adversary with respect to the predictor's weights

α: is the hyperparameter that controls the accuracy/de-biasing tradeoff

The projection term is necessary to prevent the model from working in favor of the adversary. This term removes all the components of the predictor's gradient update that would contribute to the adversary.

Adversarial networks are notoriously hard to train. First, trying to train the adversarial de-biasing method, we faced with divergence issue, changing the α hyperparameter in ($\nabla_w L_p - proj_{\nabla_w L_A} \nabla_w L_p - \alpha \nabla_w L_A$) equation solved the problem. Zhang has recommended in

his paper to set $\alpha = \sqrt{t}$ (t is the step counter) with inverse time decay on the learning rate to help the predictor to deceive the adversary (step size $\frac{1}{t}$). We used the coefficient with exponential learning rate decaya^t, which helped the model converge.

3.4 Reject Option Classification Method (RO classification)

To perform the RO classification method, we consider a binary classification problem with a potentially discriminated dataset defined over instances x, where $Y \in \{y^+, y^-\}$ is the classification label, and y^+ and y^- are the desirable and undesirable labels, respectively. We suppose that the instances are distinguishable between those belonging to the privileged x^f and unprivileged x^d groups.

Employing a classifier, it produces the posterior probability of instance X as $p(y^+|X)$. As long as it is close to 1 or 0, the instance X can be classified with a high degree of certainty. But as it gets closer to 0.5, the label becomes less certain. This is when a *reject option* can be adopted. F. Kamiran et al. (2012) call this region as the *critical region* where the instances (rejected instances) might be affected by decision discrimination. In order to mitigate this potential bias, the rejected samples would be labeled in a manner different from the conventional method. As a result, if an instance belongs to the unprivileged group x^d, it is labeled as y^+, otherwise it is labeled as y^-. While according to the standard decision rule, if $p(y^+|X) > p(y^-|X)$ then y^+ would be assigned to the instance.

3.5 Fairness Metrics

For evaluating fairness, we will follow the procedure offered in Beutel et al. (2017) and Zhang et al. (2018), and also the metric proposed in Speicher et al. (2018). The fairness of classifiers is calculated based on the following formulas. We suppose $\hat{Y}$ represents the predicted label while $\hat{Y} = 1$ and $\hat{Y} = 0$ are the desirable and undesirable labels respectively. $A = 1$ and $A = 0$ stand for the privileged and unprivileged groups.

3.5.1 Statistical Parity Difference

To illustrate the *demographic parity,* we measure the statistical parity difference between the privileged and unprivileged groups. In other words, it is the difference in the rate of favorable outcomes received by the unprivileged group to the privileged group (Bellamy et al., 2018). According to the equation, the ideal value of this metric is 0. Negative values show a higher value of demographic parity in privileged groups, while values greater than 0 show higher statistical parity for the unprivileged group.

Equation 26) Statistical Parity difference

$$P(\hat{Y} = 1|A = 0) - P(\hat{Y} = 1|A = 1) = \frac{TP_{A=0} + FP_{A=0}}{N_{A=0}} - \frac{TP_{A=1} + FP_{A=1}}{N_{A=1}} \quad (26)$$

N_A is the number of examples with the sensitive attribute set to A. $TP_A$and FP_A are the number of true positive and false positives in class A, respectively.

3.5.2 Equal Opportunity Difference

To understand the *equality of opportunity,* we calculate the true positive rate (TPR) difference between the unprivileged and privileged groups. This metric's ideal value is zero, and a value less than zero illustrates a higher benefit for the privileged group. In comparison, a positive value is a sign of higher benefit for the unprivileged group.

Equation 27) Equal Opportunity Difference

$$P(\hat{Y} = 1|A = 0, Y = 1) - P(\hat{Y} = 1|A = 1, Y = 1) = \frac{TP_{A=0}}{FN_{A=0} + TP_{A=0}} - \frac{TP_{A=1}}{FN_{A=1} + TP_{A=1}} \quad (27)$$

3.5.3 Average Odds Difference

To demonstrate the *equality of odds*, we present the false-positive rates (FPR) and true-positive rates (TPR) and compare them across different protected groups' values, which should show approximate equality. In other words, the best value for this metric is zero, same

as and the other two metrics, negative values show benefit in favor of the privileged group, and positive values are the representation of benefic in favor of the unprivileged group. The average odds difference is calculated with the following equation.

Equation 28) Average odds difference

$$\frac{1}{2}[(FPR_{A=0} - FPR_{A=1}) + (TPR_{A=0} - TPR_{A=1})] = \frac{1}{2}\left[\left(\frac{FP_{A=0}}{TN_{A=0} + FP_{A=0}} - \frac{FP_{A=1}}{TN_{A=1} + FP_{A=1}}\right) + \left(\frac{TP_{A=0}}{FN_{A=0} + TP_{A=0}} - \frac{TP_{A=1}}{FN_{A=1} + TP_{A=1}}\right)\right] \quad (28)$$

3.5.4 Disparate Impact Ratio

According to Feldman et al. (2015), a dataset with the classification label Y and sensitive attribute A has disparate impact when the following equation has an amount less than 0.8 or more than 1.25. The ideal value for this metric is 1. Values greater than one show benefit in favor of the unprivileged, and values smaller than one show benefit in favor of the privileged group.

Equation 29) Disparate Impact Ratio

$$\frac{P(Y = 1|A = 0)}{P(Y = 1|A = 1)} \quad (29)$$

This definition is based on the given dataset and real outcomes. We are going to use it on the original dataset before applying any algorithm to show the potential of unfairness in the initial dataset. The same metric will be used for measuring the fairness of the classification algorithms when the predicted labels $\hat{Y}$ are used instead of the real ones.

3.5.5 Theil Index

Theil index is the generalized entropy index (section 2.3.4) with $\alpha = 1$, and is calculated based on the following equation to show individual and group fairness (b_i is the benefit for the individual i). A value of 0 implies perfect fairness.

Equation 30) Benefit in Theil index

$$b_i = \hat{y}_i - y_i + 1 \qquad (30)$$

Equation 31) Theil index

$$\varepsilon^{\alpha}(b_1, b_2, \dots, b_n) = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{b_i}{\mu}\right) Ln\left(\frac{b_i}{\mu}\right) \qquad (31)$$

3.6 Tools

We will do all the experiments in the python programming language using "PyCharm" IDE. We use the "scikit-learn"[3] library for applying the feature engineering methods, classification algorithms, and classification performance metrics (Pedregosa et al., 2011). We will perform the necessary dataset loading and pre-processing techniques using the "Pandas"[4] (Reback et al., 2020)and "Numpy"[5] libraries, while the "Adversarial de-biasing" method is going to be run through the "TensorFlow"[6] (Abadi et al., 2016) library environment.

To implement the de-biasing algorithms and measure the fairness metrics, we will use the "aif360"[7] toolkit, an open-source Python toolkit for algorithmic fairness (Bellamy et al., 2018).

[3] https://scikit-learn.org/stable/
[4] https://pandas.pydata.org/
[5] https://numpy.org/
[6] https://www.tensorflow.org/
[7] https://aif360.readthedocs.io/en/latest/index.html#

4 Experiments and Evaluation

We compare the performance of the three different de-biasing methods (Reweighing, Adversarial de-biasing, and Reject Optimization Classification) through experiments with various datasets. The datasets include: “Heart Failure”, “Diabetes”, and the “German Credit”, and “CAMH mental health and the covid-19 pandemic”. We use five different classification methods (Naïve Bayes, Logistic Regression, Neural Networks, Decision Trees, and Support Vector Machines). All the performance and fairness metrics are calculated and compared with the baseline status where no de-biasing method is applied.

To validate our models' performance, we will perform the ten-fold cross-validation technique on all the experiments. We will use the Grid search method to optimize the hyperparameters to the baseline algorithms' highest accuracy. Instances with missing values are removed from the datasets. Prior to the training process, we will standardize all the numerical values. As gradient-based learning schemes are sensitive to feature scaling, do min-max scaling for the adversarial and Neural Network methods. To assure that all of the algorithms can handle the dataset's features, we transform all of the categorical values to numbers through the *One-Hot encoding* procedure.

In order to reduce the dimensionality of the diabetes dataset and to make it easier to work on this dataset, we are going to utilize the feature engineering methods introduced in section 3.1, and the irrelevant features will be removed for the sake of better performance.

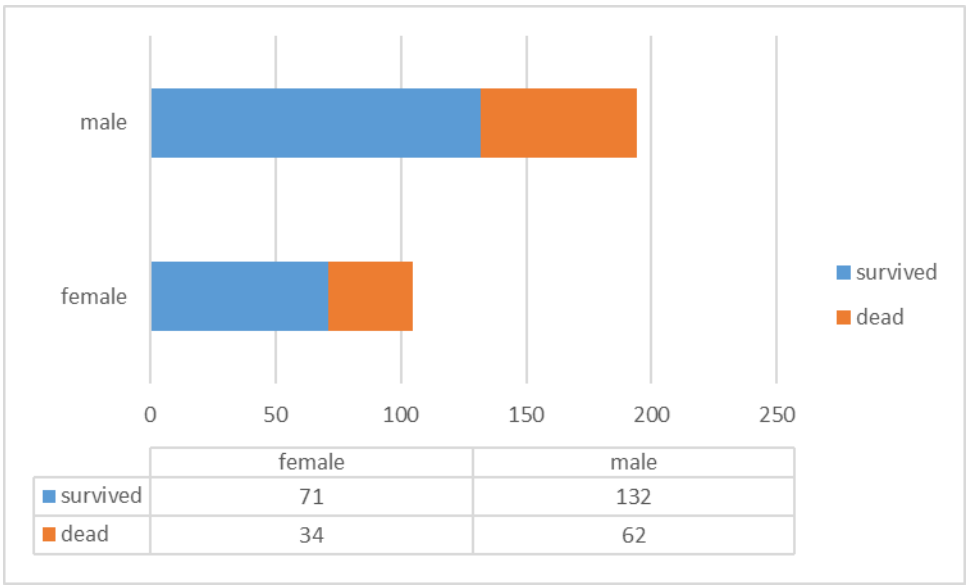
4.1 Heart Failure Dataset

We used the Heart failure clinical records Data Set available on the UCI Machine Learning Repository website. It was collected at the Faisalabad Institute of Cardiology and the Allied Hospital in Faisalabad (Punjab, Pakistan) from April to December 2015. The most recent version of the dataset was elaborated by Davide Chicco (Krembil Research Institute, Toronto, Canada) and donated to the University of California Irvine Machine Learning Repository ("Heart failure clinical records Data Set," 2020). The dataset is composed of the medical records of 299 patients who had heart failure. It contains 13 features, which report clinical, body, and lifestyle information (Chicco & Jurman, 2020). The details of the features

are shown in the appendix. As the dataset is skewed regarding the gender of the patients (105 women and 194 men), this attribute is considered the protected variable, and the men label is the privileged group. The experiment is done as a binary classification with the “Death Event” as the classification label (survived= 0, dead=1).

4.1.1 Dataset Distribution

The Heart Failure dataset includes 299 instances with 203 negative (survived) and 96 positive (dead) cases. If we predict all the labels as positive, the accuracy of the prediction would be 32%, and the balanced accuracy would be 0.5. As it shows in “figure 2,” although the number of male cases (194) is higher than female cases (105), in both groups, 32% of cases are labeled as “dead”. The percentage of male cases in both death event labels is 65 “figure 3”. The dataset initial disparate impact ratio of the unprivileged group to the privileged group is 1.03. As the ideal value of this metric is one, there is a low potential of bias in the original dataset. During the following experiments, we test the bias that is generated through machine learning classifiers.



	female	male
survived	71	132
dead	34	62

Figure 2) bar plot of "death event" by the gender groups

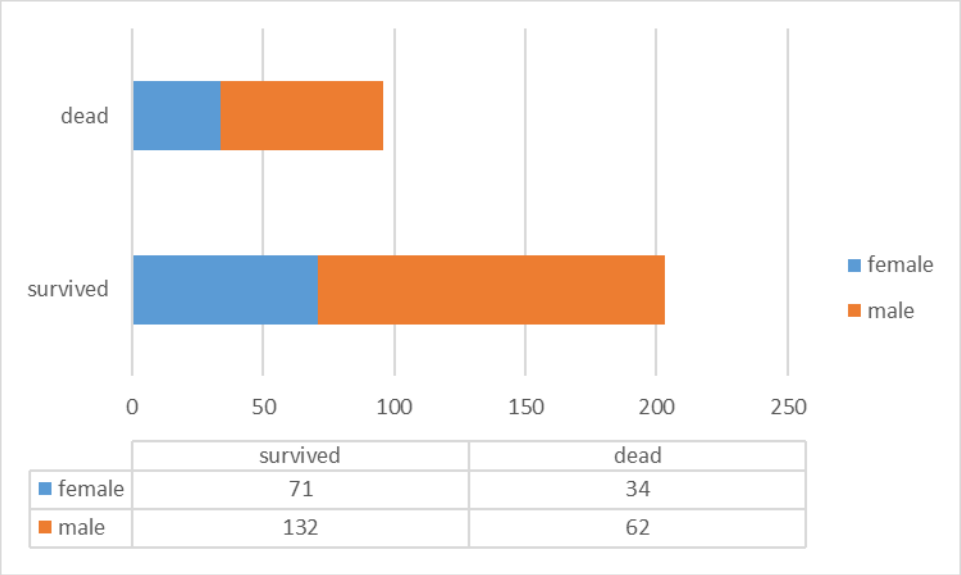


Figure 3) bar plot of gender groups by "death event" labels

4.1.2 Classification Results Before De-Biasing

Training the Logistic Regression classifier[8], different optimizers, and parameter tuning were tried to get the highest accuracy. "LIBLINEAR"[9] optimization algorithm with the "L1" penalty and the regularization parameter $C = 0.001$ showed the best performance during the test, and the maximum number of iterations is set to 100. The Neural Network classifier[10] showed the best results with one hidden layer of size 200 and 500 times of iteration, constant learning rate, and regularization parameter alpha=0.05 with "Relu" activation function and "Adam" optimizer (Kingma & Ba, 2014). The Decision Tree classifier[11] with the "Gini" criterion, and a maximum depth of two was chosen as the baseline. As a Support Vector Machine, SVM[12] with the "RBF" kernel, Regularization parameter C=100, and kernel coefficient gamma=0.001 was chosen. For the Naïve Bayes classifier,[13] the likelihood of the features is assumed Gaussian. The rest of the parameters are set as the "Scikit Learn" default settings.

[8] https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html
[9] https://www.csie.ntu.edu.tw/~cjlin/liblinear/
[10] https://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPClassifier.html
[11] https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html
[12] https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html
[13] https://scikit-learn.org/stable/modules/naive_bayes.html

The details of the experiment with the evaluation and fairness metrics measurements before and after de-biasing are shown in the following tables.

Table 3) Evaluation of classifiers on Heart Failure dataset before De-biasing

Heart Failure dataset											
De-biasing Method: No											
Classification Method											
		Naïve Bayes (Gaussian)		Logistic Regression		Neural Network		Decision Trees		SVM	
Evaluation Metrics	Accuracy	0.773		0.810		0.766		0.843		0.753	
	Precision	0.733		0.808		0.761		0.855		0.634	
	Recall	0.463		0.527		0.446		0.633		0.532	
	F1	0.556		0.630		0.528		0.717		0.570	
	AUC	0.690		0.734		0.681		0.787		0.695	
Fairness Metrics	Statistical Parity	-0.051		0.036		-0.016		-0.022		0.033	
	Equality of Opportunity	-0.085		0.017		-0.006		-0.109		0.034	
	Average odds Difference	-0.075		0.026		-0.019		-0.056		0.021	
	Disparate Impact Ratio	1.151		1.355		1.015		1.134		1.581	
	Theil Index	0.210		0.180		0.217		0.140		0.198	
		prediction									
confusion matrix		**N**	**P**	**N**	**P**	**N**	**P**	**N**	**P**	**N**	**P**
	N	186	17	191	12	186	17	191	12	174	29
	P	51	45	45	51	53	43	35	61	45	51

Performing the “death event” classification task on the “heart failure” dataset, shows among the five “Naïve Bayes”, “Logistic Regression”, “Neural Networks”, “Decision Tree”, and the “Support Vector Machine” classifiers “Decision Tree” has the best prediction result

related to all of the five classification metrics used in this experiment. The average accuracy obtained with this classifier is 84%. At the same time, the lowest accuracy is related to the SMV classifier with an accuracy of 75%.

Considering the fairness measures, the best possible value for the “Statistical Parity”, “Equal Odds”, “Equal Opportunity”, and the “Theil index” metrics is zero. The ideal value for the “Disparate Ratio” is one. “Statistical Parity”, “Equal Odds”, “Equal Opportunity” have positive values in the Logistic Regression and the SVM classifiers that show an average benefit in favor of the unprivileged group while they are negative in the other classifiers. This difference indicates that different classifiers have different behaviors regarding fairness with the same dataset. “Neural Networks” classifier has the best performance in all the fairness measures among the classifiers except for the Theil index, which has the best value with the Decision Tree classifier. In the case of statistical parity and the average odds, the poorest performance belongs to the Naïve Bayes classifier. At the same time, Equality of Opportunity in Decision Trees and the disparate ratio in SVM show the worst values.

4.1.3 Classification Results after De-Biasing With Reweighing

This section investigates the effect of the reweighing preprocessing method on the evaluation and fairness measures of classifiers on the Heart Failure dataset. The details of these experiments are shown in the table 4.

The reweighing method shows a positive effect on Naïve Bayes, Neural Network, and the SVM classifiers because there is a rise in the evaluation metrics values, while the bias amount has decreased after using this method. With regard to the Logistic Regression classifier, we can observe a slight reduction in the evaluation metrics, such as accuracy that shows one percent decrease from 81% to 80%. Still, except for the statistical parity, which has got closer to zero (from 0.036 to -0.025), there is no improvement in the other fairness measures. There is deterioration in the evaluation metrics with the Decision Tree classifier, including a 4% reduction in the accuracy (from 84% to 80%), although this difference is statistically insignificant. Performing a two-tail t-test between the two accuracy values yields a p-value of 0.24, which is not less than the significance level (typically 0.001 or 0.05) and shows an insignificant difference between the accuracy of the two methods. Using the

reweighing technique with the Decision Tree classifier has led to the improvement in the equality of opportunity, odds, and the disparate impact fairness.

Table 4) Evaluation of classifiers on Heart Failure dataset after De-biasing with Reweighing

Heart Failure dataset											
De-biasing Method											
Classification Method											
		Naïve Bayes (Gaussian)		Logistic Regression		Neural Network		Decision Trees		SVM	
Evaluation Metrics	Accuracy	0.792		0.803		0.769		0.806		0.792	
	Precision	0.732		0.713		0.646		0.754		0.697	
	Recall	0.600		0.689		0.651		0.630		0.654	
	F1	0.642		0.680		0.627		0.666		0.654	
	AUC	0.750		0.781		0.745		0.766		0.765	
Fairness Metrics	Statistical Parity	-0.036		-0.025		0.007		-0.024		-0.046	
	Equality of Opportunity	-0.042		-0.025		0.014		0.006		-0.184	
	Average odds Difference	-0.045		-0.026		-0.006		-0.012		-0.103	
	Disparate Impact Ratio	0.885		0.963		1.062		1.016		1.032	
	Theil Index	0.181		0.144		0.162		0.157		0.160	
		prediction									
confusion matrix		**N**	**P**	**N**	**P**	**N**	**P**	**N**	**P**	**N**	**P**
	N	183	20	176	27	169	34	182	21	177	26
	P	42	54	32	64	35	61	37	59	36	60

4.1.4 Classification Results after De-Biasing With Reject Option Classification

The Reject Option classification technique needs to be performed based on (statistical parity, Equality of odds, and equal opportunity) fairness metrics separately. Therefore the results are shown in three different tables as follows.

4.1.4.1 Reject Option-Statistical Parity Difference

The highest evaluation measures using the Reject Option classification (Statistical Parity) belongs to the Logistic Regression classifier, which shows improvements in all metrics except for the precision metric. Similarly, Neural Network and the SVM methods show improvement in the evaluation metrics except for the precision value in the Neural Network. There is a reduction in the accuracy and the precision values with the Naïve Bayes and the Decision Trees classifiers. A t-test between the accuracy of classifiers before and after de-biasing resulted in a p-value of 0.09 for the Naïve Bayes and a p-value of 0.01 for the Decision Tree classifiers, which are indications of statistical insignificance for Naïve Bayes but significant decrease for the Decision Tree classifier.

This method successfully reduced the statistical parity difference in the Naïve Bayes, Logistic regression, and the SVM classifiers. However, it has not shown any improvement in the Neural Network and the Decision Tree classifiers. On the other hand, the disparate impact ratio, which is another representation of statistical parity, has been reduced in all classifiers. It is worth mentioning that the Theil index has decreased in all the classifiers that can be a sign of improvement in both the group and individual fairness.

Table 5) Evaluation of classifiers on Heart Failure dataset after De-biasing with Reject Option (Statistical parity difference)

Heart Failure dataset											
De-biasing Method											
Classification Method											
		Naïve Bayes (Gaussian)		Logistic Regression		Neural Network		Decision Trees		SVM (linear)	
Evaluation Metrics	Accuracy	0.679		0.836		0.786		0.756		0.813	
	Precision	0.530		0.742		0.689		0.670		0.715	
	Recall	0.870		0.830		0.832		0.731		0.818	
	F1	0.630		0.767		0.721		0.652		0.737	
	AUC	0.739		0.830		0.798		0.755		0.814	
Fairness Metrics	**Statistical Parity**	0.009		0.013		0.018		-0.082		0.015	
	Equality of Opportunity	0.090		0.044		-0.025		-0.138		0.032	
	Average odds Difference	0.043		0.020		0.024		-0.100		0.014	
	Disparate Impact Ratio	1.018		1.034		1.042		0.886		1.048	
	Theil Index	0.102		0.087		0.088		0.137		0.091	
		prediction									
confusion matrix		**N**	**P**	**N**	**P**	**N**	**P**	**N**	**P**	**N**	**P**
	N	122	81	170	33	155	48	158	45	164	39
	P	15	81	16	80	16	80	28	68	17	79

4.1.4.2 Reject Option-Average Odds Difference

The results of the Reject Option classification (odds difference) method show a slight reduction in accuracy except for the Naïve Bayes and the SVM classifiers. In contrast, the balanced accuracy of all of them except for the SVM has increased. It shows that this method is the right way of dealing with the imbalance in the dataset. The results of the T-test on the Logistic Regression, Neural Networks, and the Decision Trees are P-values of 0.64, 0.88, and

0.32, respectively. Considering the P-values, we can conclude the reduction of the accuracy values is statistically insignificant.

Average odds difference and the Theil index have decreased significantly in all of the classifiers. The reduction is a good sign of improvement in the group and individual fairness.

Table 6) Evaluation of classifiers on Heart Failure dataset after De-biasing with Reject Option (Average odds difference)

Heart Failure dataset											
De-biasing Method: Reject Option-Average odds difference											
Classification Method											
		Naïve Bayes (Gaussian)		Logistic Regression		Neural Network		Decision Trees		SVM	
Evaluation Metrics	Accuracy	0.783		0.796		0.759		0.816		0.786	
	Precision	0.635		0.657		0.640		0.735		0.726	
	Recall	0.871		0.896		0.900		0.754		0.797	
	F1	0.712		0.730		0.707		0.715		0.681	
	AUC	0.810		0.823		0.797		0.810		0.791	
Fairness Metrics	Statistical Parity	-0.009		0.007		0.022		0.020		0.033	
	Equality of Opportunity	0.079		-0.007		-0.063		0.004		-0.042	
	Average odds Difference	0.010		0.003		0.002		0.007		0.009	
	Disparate Impact Ratio	0.991		1.013		1.080		1.125		1.086	
	Theil Index	0.089		0.076		0.076		0.127		0.104	
		prediction									
confusion matrix		**N**	**P**	**N**	**P**	**N**	**P**	**N**	**P**	**N**	**P**
	N	152	51	152	51	141	62	175	28	159	44
	P	14	82	10	86	10	86	27	69	20	76

4.1.4.3 Reject Option-Equal Opportunity Difference

Applying the Reject Option (Equality of Opportunity) technique, the best performance is observed with the Neural Network classifier. There are considerable increases in the accuracy value (from 76% to 78%), recall (from 44% to 92%), F1 (from 52% to 73%), and AUC (from 68% to 82%). Its opportunity difference has successfully reduced to zero.

Opportunity difference and the Theil index values have decreased in all of the classifiers. Except for the SVM and the Neural Networks classifiers with a two percent increase in accuracy values, it has fallen in other classifiers.

Table 7) Evaluation of classifiers on Heart Failure dataset after De-biasing with Reject Option (Equal opportunity difference)

Heart Failure dataset											
De-biasing Method: Reject Option-Equal opportunity difference											
Classification Method											
		Naïve Bayes (Gaussian)		Logistic Regression		Neural Network		Decision Trees		SVM	
Evaluation Metrics	Accuracy	0.735		0.752		0.786		0.709		0.773	
	Precision	0.570		0.595		0.662		0.414		0.666	
	Recall	0.976		0.985		0.925		0.485		0.932	
	F1	0.705		0.722		0.736		0.438		0.735	
	AUC	0.797		0.816		0.823		0.644		0.814	
Fairness Metrics	Statistical Parity	-0.036		0.120		0.078		-0.046		0.083	
	Equality of Opportunity	0.013		-0.002		0.000		-0.079		-0.003	
	Average odds Difference	-0.029		0.080		0.057		-0.077		0.055	
	Disparate Impact Ratio	0.966		1.301		1.255		nan		1.269	
	Theil Index	0.057		0.052		0.068		0.225		0.062	
		prediction									
confusion matrix		**N**	**P**	**N**	**P**	**N**	**P**	**N**	**P**	**N**	**P**
	N	127	76	131	72	147	56	166	37	142	61
	P	3	93	2	94	8	88	50	46	7	89

4.1.5 Classification Results after De-Biasing With Adversarial De-Biasing

The adversarial de-biasing classifier is a one layer Neural Network with 200 hidden units. The number of epochs and the batch size is 100 and 27, respectively. The adversary loss weight is set to the default of the aif360 toolkit that is 0.1 with exponential learning rate decay. In comparison with the results of a Neural Network classification without the de-

biasing technique, the accuracy has not been reduced, and there is an improvement in the average odds difference from 0.019 to 0.003, but the other bias measures have not reduced.

Table 8) Evaluation of Adversarial de-biasing on Heart Failure dataset after De-biasing

Heart Failure dataset				
De-biased: Adversarial De-biasing				
Evaluation Metrics				
Accuracy	Precision	Recall	F1	AUC
0.773	0.733	0.463	0.556	0.690
Fairness Metrics				
Statistical Parity	Equality of Opportunity	Average odds Difference	Disparate Impact Ratio	Theil Index
0.017	-0.037	0.003	1.189	0.122

confusion matrix		
	N	**P**
N	155	48
P	24	72

4.1.6 Comparison of De-Biasing Methods on the Heart Failure Dataset

Considering “figure 4,” we can compare the performance of the de-biasing techniques regarding their prediction accuracy. Generally, it shows that the reweighing approach, on average, has better results than the other methods. The highest rate of reduction in the accuracy value is related to the Decision Tree classifier. Both the reject option and the reweighing methods have deteriorated the accuracy value.

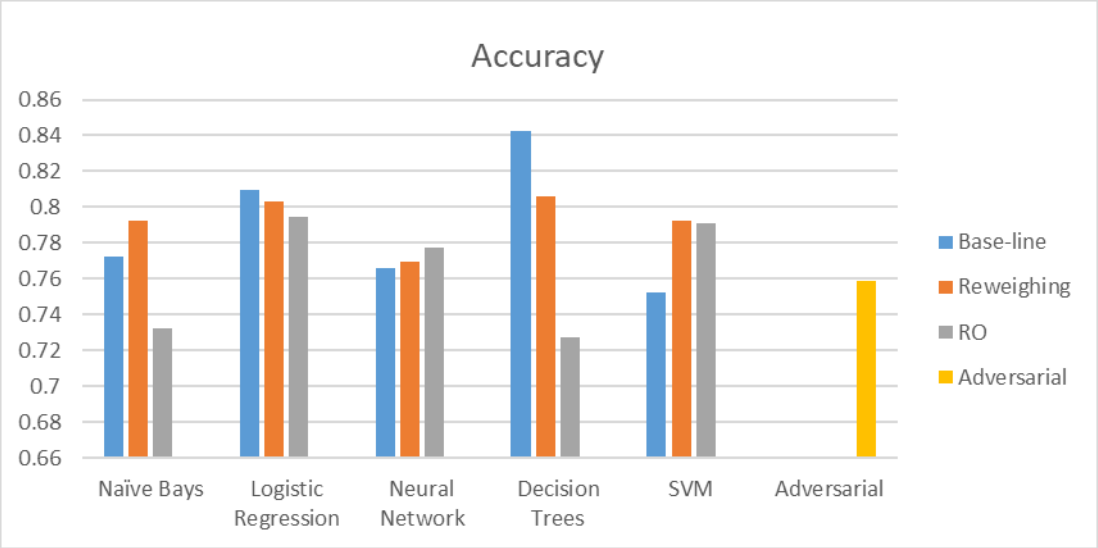


Figure 4) Comparison of de-biasing methods on the Heart Failure Dataset regarding the accuracy measure

In terms of statistical parity difference, the Reject Option classification method shows a better performance except for the Decision Trees and Neural Network classifiers. The Decision Tree algorithm has the lowest statistical parity difference when there is no de-biasing applied. Neural Network shows a better performance with the reweighing technique.

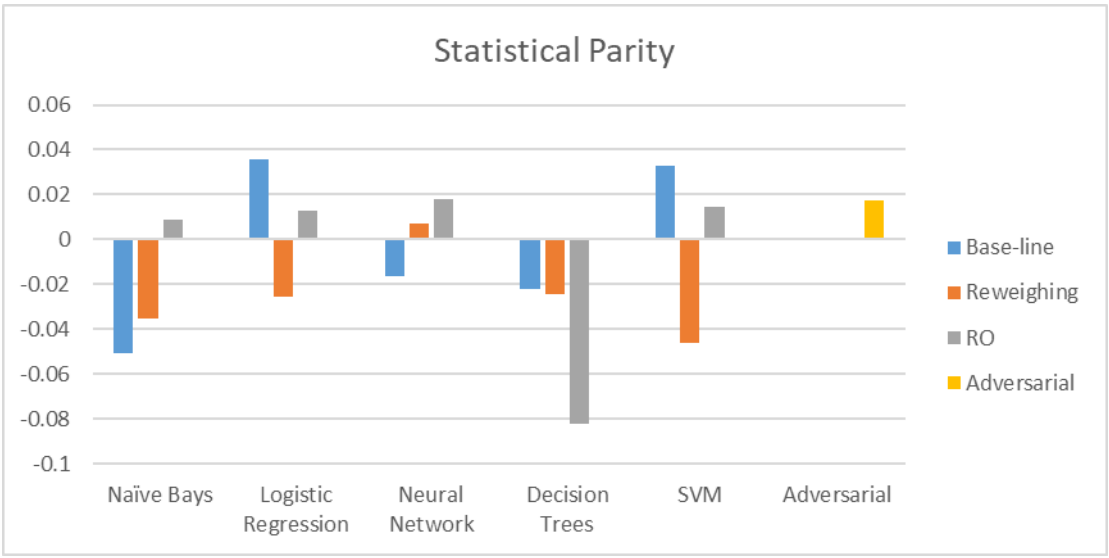


Figure 5) Comparison of de-biasing methods on the Heart Failure Dataset regarding the Statistical Parity measure

According to figure 6, the Reject Option method is more successful in reducing the opportunity difference in the Naïve Bayes, Logistic Regression, Neural Network, and the SVM classifiers. The reweighing technique decreases the opportunity difference more than the Reject Option with the Decision Tree classifier.

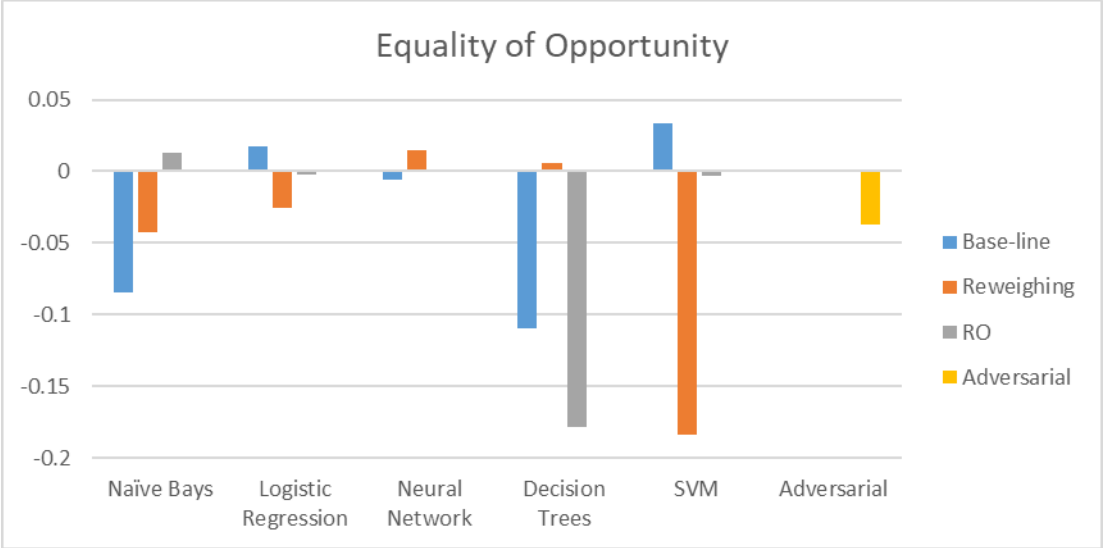


Figure 6) Comparison of de-biasing methods on the Heart Failure Dataset regarding the Equality of Opportunity measure

The lowest values of average odds difference are obtained using the adversarial de-biasing method and the Reject Option method with the Neural Network, Logistic Regression, Naïve Bayes, and the SVM classifiers. For the Decision Tree classifier, only the reweighing method seems capable of reducing the odds difference.

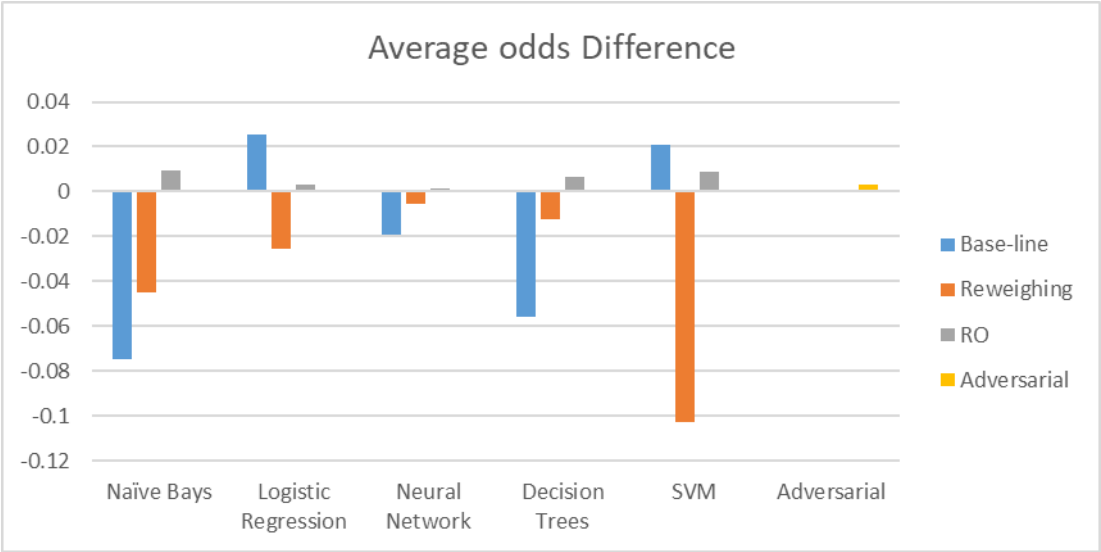


Figure 7) Comparison of de-biasing methods on the Heart Failure Dataset regarding the Average of Odds measure

As depicted in Figure 8, except for the Decision Tree classifier, the Reject Option method yields lower values of the Theil index. As a result, it shows a lower bias in individual and group levels. In terms of the Decision Tre classifier, the baseline version has a lower Theil index than the de-biased versions.

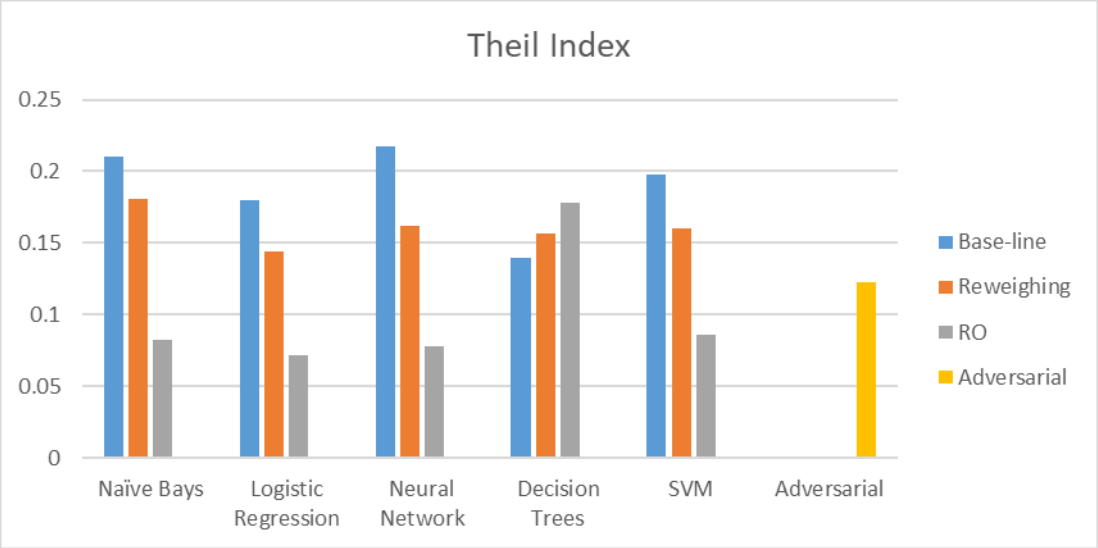


Figure 8) Comparison of de-biasing methods on the Heart Failure Dataset regarding the Theil Index measure

4.2 Diabetes Dataset

We use the "Diabetes 130-US Hospitals" data set, available at UCI Machine Learning Repository ("Diabetes 130-US hospitals for years 1999-2008 Data Set," 2014).

The data set is extracted by Strack et al. (2014) from the Health Facts database (Cerner Corporation, Kansas City, MO). This national data warehouse collects comprehensive clinical records across hospitals throughout the United States. The data set represents the clinical care data among 130 hospitals across the United States during a period of 10 years from 1999 to 2008. The initial dataset consisted of 101767 records of male and female gender combined, each belonging to one of the following races: Caucasian, African American, Asian, Hispanic, and other (Tran & Sokolova, 2016). Out of 117, 55 features were retained in the original dataset describing encounters, demographics, diagnoses, diabetic medications, number of visits, and payment details.

In many pieces of research on diabetic data mining and machine learning, the study entails learning the presence or absence of diabetes, and HBA1c has great significance as a diabetes marker. Such a learning process is a binary classification.

This dataset includes 101767 instances and 50 features. To align with the work done on the data set by Strack et al. (2014), we are going to perform the de-biasing process on the

early readmission classification of the diabetic patients considering covariates such as demographic, severity, and type of disease, and kind of admission. As we are interested in early readmission, we defined two values for this attribute: “readmitted”, if the patient was readmitted within 30 days of the previous discharge, and “otherwise,” which covers both readmissions after 30 days or no readmission at all.

The diabetes dataset is known as a complex dataset in terms of heterogeneity, high dimensionality, and a considerable rate of missing data (Strack et al., 2014). As a result, an efficient pre-processing procedure plays a vital role in achieving reliable results. In this regard, we removed three of the features due to the high number of missing values. These features are “weight”, “payer code,” and “medical specialty”. Instances with missing values are removed. In the original dataset, there are multiple inpatient visits recorded for some of the patients. In order to keep the cases statistically independent, we only used the first encounter of each patient. Also, to minimize the possibility of biasing our analysis, we removed the cases that ended up with discharge to a hospice or patient death. Performing feature selection techniques on the dataset, the “examide”, “citoglipton”, and “metformin-rosiglitazone” features are removed due to zero variance in all of the instances. To make the classification task easier, categorical features are converted to numerical using the One Hot encoder method, and standardization is done on all numerical values.

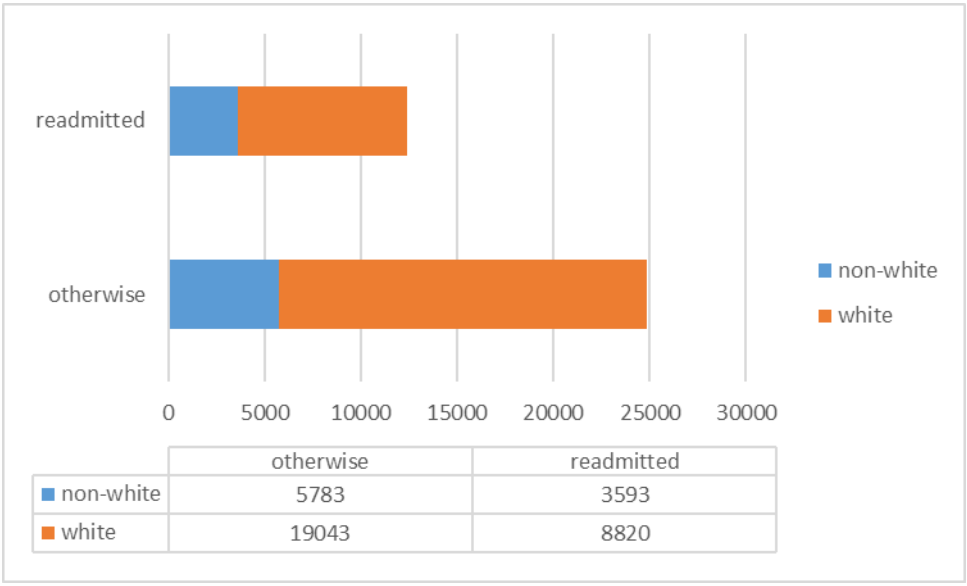
4.2.1 Dataset Distribution

We consider the “readmitted” as the positive label. The total number of the positive label is 6115, which, in comparison to the negative label “otherwise” with 62065 samples, is about ten times less. This difference shows an imbalance in our dataset. One approach for addressing the problem of class imbalance is randomly resampling the training dataset. We are going to apply SMOTE oversampling and the random under-sampling methods to the dataset. At first, using the SMOTE method to oversample the positive label to 0.2 of the negative labels. Then delete some samples from the majority class to reach the desired ratio of the number of minority class to the majority to be 0.5. Using this technique, we reduce the imbalance in the dataset while making sure not too much useful information is removed from the majority class.

We chose the race of patients as the protected attribute, and due to the high frequency of the “Caucasian” race in comparison to other races, we put them as the privileged (white) and unprivileged (non-white) groups, respectively.

The final dataset is left with 37239 instances (24826 negative and 12413 positive examples) and 44 features. If all examples are classified as positives, our accuracy would be 0.33 and, our Balanced Accuracy would be 0.5. From the total amount of dataset, 9376 and 27863 cases are “non-white” and “white” races, respectively.

As it is depicted in figure 9, 77% of the readmitted cases are from the white group, and 71% of the otherwise labeled instances are from the white group.



	otherwise	readmitted
non-white	5783	3593
white	19043	8820

Figure 9) bar plot of race groups by the "readmitted" label

In case we examine the white people visits to the hospital, 32% of them have been readmitted, while this ratio is about 38% among the non-white people (figure10). The disparate impact ratio of the unprivileged group to the privileged is 1.21 for the original dataset.

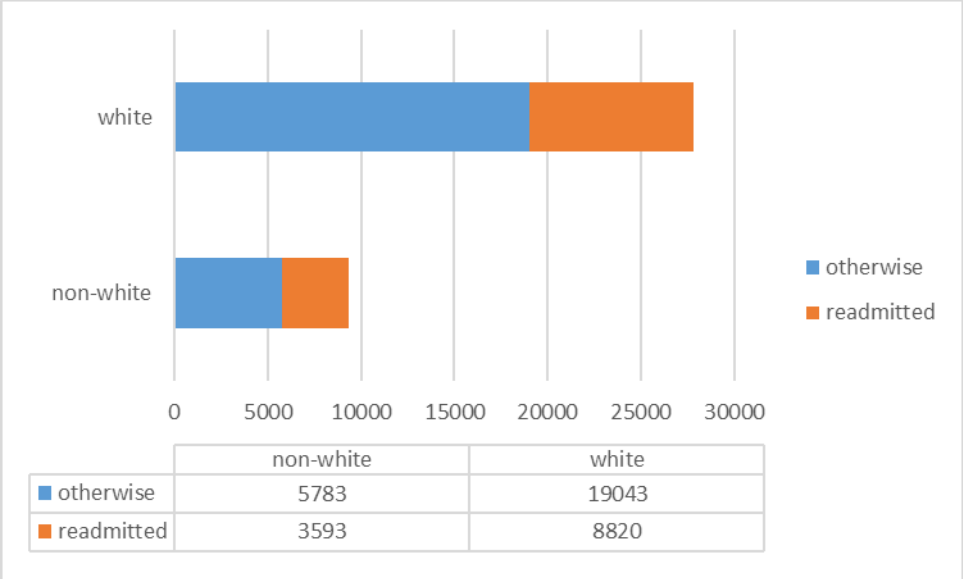


Figure 10) bar plot of readmitted by the race feature

4.2.2 Classification Results before De-Biasing

Five different classifiers with tuned hyperparameters are tested on the dataset to get the highest possible accuracy. The evaluation metric results and the fairness measures before and after de-biasing with different methods are shown in the following sections.

The details of the hyperparameter tuning of the Diabetes dataset is as follows:

Naïve Bayes: Gaussian likelihood
Logistic regression: newton-cg solver, L2 penalty, C=25 regularization parameter
Neural Networks: hidden layer of size 100, 500 times of iteration, adaptive learning rate and regularization parameter alpha=0.0001, "logistic" activation function, "Adam" optimizer
Decision Trees: "entropy" criterion, maximum depth of 12
SVM: "RBF" kernel, Regularization parameter C=1, and kernel coefficient gamma=0.01

The rest of the parameters are set as the "Scikit Learn" default settings.

According to "table 9", when there is no de-biasing method used, the highest accuracy and precision are obtained using the SVM classifiers with 77% and 83% values. The accuracy resulted from using the other classifiers is as follows: Naïve Bayes 0.34, Logistic Regression

0.718, Neural Network 0.72, and Decision Tree 0.74. The highest values of recall and F1 belong to the Naïve Bayes (0.99) and the Decision Tree (0.529) classifiers, respectively. The lowest amount of bias is observed with the Naïve Bayes classifier, while the Logistic Regression and Neural Network classifiers show the highest values of bias measures. The positive values of the statistical parity, opportunity, and odds difference show the bias in favor of the unprivileged group.

Table 9) Evaluation of classifiers on Diabetes dataset before De-biasing

Diabetes Data set											
De-biasing Method											
Classification Method											
		Naïve Bayes (Gaussian)		Logistic Regression		Neural Network		Decision Trees		SVM	
Evaluation Metrics	Accuracy	0.344		0.718		0.729		0.745		0.769	
	Precision	0.336		0.648		0.672		0.688		0.837	
	Recall	0.991		0.336		0.378		0.430		0.380	
	F1	0.502		0.443		0.479		0.529		0.522	
	AUC	0.506		0.622		0.641		0.666		0.671	
Fairness Metrics	Statistical Parity	0.003		0.101		0.101		0.068		0.075	
	Equality of Opportunity	0.003		0.143		0.148		0.115		0.120	
	Average odds Difference	0.002		0.098		0.097		0.061		0.067	
	Disparate Impact Ratio	1.003		1.690		1.631		1.361		1.567	
	Theil Index	0.048		0.275		0.258		0.237		0.243	
		prediction									
confusion matrix		**N**	**P**	**N**	**P**	**N**	**P**	**N**	**P**	**N**	**P**
	N	521	24305	22553	2273	22459	2367	22407	2419	23907	919
	P	108	12305	8237	4176	7722	4691	7081	5332	7701	4712

4.2.3 Classification Results after De-Biasing With Reweighing

Performing the Reweighing method on the Diabetes dataset leads to a small reduction of accuracy in the Logistic Regression (from 0.718 to 0.715), Neural Network (from 0.729 to 0.722), Decision Tree (from 0.745 to 0.735), and SVM (from 0.769 to 0.759). We did two-tail t-tests with a significance level of 0.05 on the classifiers as mentioned above to find out whether the reduction of accuracy values after the reweighing procedure is statistically significant or not. The resulted p-values are as follows: Logistic Regression 0.58, Neural Network 0.69, Decision Tree 0.02, SVM 0.0004. It shows that the difference is significant with the Decision Tree and the SVM classifiers. The reweighing technique successfully reduces all of the bias measures with the Logistic Regression, Decision Tree, and the SVM classifiers. However, there are increases in the bias measures with the Naïve Bayes and the Neural Network classifiers.

Table 10) Evaluation of classifiers on Diabetes dataset after de-biasing with Reweighing

Diabetes Data set											
De-biasing Method: reweighing											
Classification Method											
		Naïve Bayes (Gaussian)		Logistic Regression		Neural Network		Decision Trees		SVM	
Evaluation Metrics	Accuracy	0.391		0.715		0.722		0.735		0.759	
	Precision	0.349		0.643		0.730		0.648		0.809	
	Recall	0.949		0.328		0.406		0.450		0.361	
	F1	0.510		0.434		0.479		0.531		0.499	
	AUC	0.531		0.619		0.643		0.664		0.659	
Fairness Metrics	Statistical Parity	0.023		0.017		0.126		0.048		0.052	
	Equality of Opportunity	0.026		0.012		0.165		0.092		0.081	
	Average odds Difference	0.020		0.003		0.118		0.041		0.042	
	Disparate Impact Ratio	1.026		1.106		1.798		1.223		1.388	
	Theil Index	0.067		0.279		0.246		0.233		0.252	
		prediction									
confusion matrix		**N**	**P**	**N**	**P**	**N**	**P**	**N**	**P**	**N**	**P**
	N	2788	22038	22570	2256	21859	2967	21776	3050	23767	1059
	P	628	11785	8345	4068	7389	5024	6827	5586	7931	4482

4.2.4 Classification Results after De-Biasing With Reject Option Classification

Reject Option Classification performs de-biasing on the data based on each required fairness metric separately. As a result, we repeated the experiment three times with this method to reduce “Statistical Parity”, “Equality of Opportunity”, and “Average of Odds difference” separately. The results of applying the Reject Option classification method on the Credit dataset are shown in the following tables.

4.2.4.1 Reject Option-Statistical Parity Difference

Performing the Reject Option classification method with the statistical parity metric as the optimization method on the Diabetes dataset yields a significant decrease in the accuracy values of all the classifiers except for the Naïve Bayes. The results of the two-tail t-test with a significance level of 0.05 on the classifiers are as follows. Two percent decrease of accuracy in Logistic Regression with a p-value of 0.0002, five percent reduction of accuracy in Neural Network with a p-value of 0.053, eight percent decrease of accuracy in Decision Tree with a p-value of 0.0005, 19% decrease of accuracy in the SVM classifier with a p-value of 0.00065. The results illustrate that, except for the Neural Network classifier, the reduction of accuracy is significant after de-biasing with the reject option method.

As shown in “table 11”, by applying the reject option technique and using the statistical parity optimization method, we successfully reduce all fairness measures with all of the classifiers except for the Naïve Bayes classifier.

Table 11) Evaluation of classifiers on Diabetes dataset after de-biasing with .Reject Option-Statistical parity difference

Diabetes Data set											
De-biasing Method: Reject Option-Statistical parity difference											
Classification Method											
		Naïve Bayes		Logistic Regression		Neural Network		Decision Trees		SVM	
Evaluation Metrics	Accuracy	0.421		0.691		0.677		0.660		0.571	
	Precision	0.357		0.710		0.631		0.506		0.527	
	Recall	0.914		0.127		0.586		0.625		0.699	
	F1	0.513		0.213		0.532		0.550		0.503	
	AUC	0.544		0.551		0.653		0.650		0.602	
Fairness Metrics	**Statistical Parity**	0.048		0.046		0.048		0.045		0.037	
	Equality of Opportunity	0.050		0.083		0.083		0.065		0.058	
	Average odds Difference	0.044		0.049		0.039		0.033		0.030	
	Disparate Impact Ratio	1.058		2.080		1.243		1.119		1.197	
	Theil Index	0.081		0.353		0.190		0.184		0.154	
		prediction									
confusion matrix		**N**	**P**	**N**	**P**	**N**	**P**	**N**	**P**	**N**	**P**
	N	4341	20485	24177	649	17919	6907	16797	8029	12576	12250
	P	1067	11346	10843	1570	5121	7292	4637	7776	3727	8686

4.2.4.2 Reject Option-Average Odds Difference

According to “table 12”, performing the Reject Option method with average odds as the optimization objective results in drops in the accuracy values of all classifiers including Logistic Regression (from 0.718 to 0.663), Neural Network (from 0.729 to 0.685), Decision Tree (from 0.745 to 0.695), SVM (from 0.769 to 0.617). The p-values obtained from two-tail t-tests on the accuracy are 0.09, 0.06, and 0.005, and 0.002, respectively. In other words, the

reduction of accuracy after using the Reject Option method (average odds) is statistically significant with the Decision Tree and the SVM classifiers.

Utilizing this de-biasing method, we can reduce all of the bias measures with all of the classifiers except for the Naïve Bayes classifier, which shows an increase in all of the de-biasing metrics.

Table 12) Evaluation of classifiers on Diabetes dataset after de-biasing with Reject Option- Average odds difference

Diabetes Data set											
De-biasing Method: Reject Option-Average odds difference											
Classification Method											
		Naïve Bayes (Gaussian)		Logistic Regression		Neural Network		Decision Trees		SVM	
Evaluation Metrics	Accuracy	0.436		0.663		0.685		0.695		0.617	
	Precision	0.361		0.672		0.579		0.557		0.563	
	Recall	0.894		0.211		0.624		0.564		0.653	
	F1	0.514		0.244		0.565		0.552		0.519	
	AUC	0.550		0.550		0.670		0.662		0.625	
Fairness Metrics	Statistical Parity	0.054		0.046		0.060		0.055		0.053	
	Equality of Opportunity	0.053		0.071		0.083		0.074		0.070	
	Average odds Difference	0.048		0.046		0.047		0.042		0.044	
	Disparate Impact Ratio	1.066		1.950		1.210		1.176		1.243	
	Theil Index	0.090		0.324		0.180		0.202		0.168	
		prediction									
confusion matrix		**N**	**P**	**N**	**P**	**N**	**P**	**N**	**P**	**N**	**P**
	N	5137	19689	22064	2762	17757	7069	18874	5952	14848	9978
	P	1316	11097	9783	2630	4667	7746	5422	6991	4299	8114

4.2.4.3 Reject Option-Equal Opportunity Difference

Using the Reject Option method with the equal opportunity metric deteriorates the accuracy of all of the classifiers except for the Naïve Bayes, which its accuracy has increased from 0.344 to 0.435. The accuracy of the Logistic Regression classifier has reduced about 26% with a p-value of zero that shows a statistical significance. For the Neural Network classifier, a drop of about 14% in accuracy has happened with a zero p-value. Pertaining to the Decision Tree and the SVM classifiers, the percentage of reduction in the accuracy values is 14 and 25%. The p-values for both of them are about zero. So we can conclude that using this method decreases the accuracy of classifiers significantly. On the other hand, the process can considerably reduce all of the bias metrics with all of the classifiers except for the Naïve Bayes.

Table 13) Evaluation of classifiers on Diabetes dataset after de-biasing with Reject Option- Equal opportunity difference

Diabetes Data set										
De-biasing Method										
Classification Method										
		Naïve Bayes (Gaussian)		Logistic Regression		Neural Network		Decision Trees		SVM
Evaluation Metrics	Accuracy	0.435		0.457		0.583		0.597		0.515
	Precision	0.362		0.421		0.470		0.449		0.430
	Recall	0.898		0.779		0.759		0.729		0.826
	F1	0.515		0.442		0.550		0.549		0.530
	AUC	0.550		0.537		0.628		0.630		0.592
Fairness Metrics	Statistical Parity	0.058		0.059		0.040		0.033		0.037
	Equality of Opportunity	0.048		0.044		0.047		0.045		0.040
	Average odds Difference	0.050		0.052		0.027		0.022		0.027
	Disparate Impact Ratio	1.074		1.236		1.096		1.068		1.089
	Theil Index	0.088		0.127		0.138		0.151		0.113
		prediction								
confusion matrix		**N**	**P**	**N**	**P**	**N**	**P**	**N**	**P**	**N**	**P**
	N	5055	19771	7330	17496	12316	12510	13186	11640	8917	15909
	P	1258	11155	2736	9677	3004	9409	3350	9063	2140	10273

4.2.5 Classification Results after De-Biasing With Adversarial De-Biasing

The adversarial de-biasing classifier is a one layer Neural Network with 100 hidden units. The number of epochs and the batch size is 100 and 30, respectively. The adversary loss weight is set to the default of the aif360 toolkit that is 0.1 with exponential learning rate decay. Compared with the Neural Network classification results without the de-biasing technique, there is no reduction in the accuracy value as it has increased from 0.73 to 0.77.

This method has slightly reduced some of the bias metrics, including statistical parity (from 0.101 to 0.092), average odds difference (from 0.097 to 0.085), and the Theil index (from 0.258 to 0.235).

Table 14) Evaluation of classifiers on Diabetes dataset after De-biasing with Adversarial de-biasing

Diabetes Data set				
De-biased: Adversarial De-biasing				
Evaluation Metrics				
Accuracy	Precision	Recall	F1	AUC
0.77	0.78	0.41	0.54	0.68
Fairness Metrics				
Statistical Parity	Equality of Opportunity	Average odds Difference	Disparate Impact Ratio	Theil Index
0.092	0.147	0.085	1.607	0.235

confusion matrix		
	N	**P**
N	23415	1411
P	7326	5087

4.2.6 Comparison of De-Biasing Methods on the Diabetes Dataset

This section will compare the effects of different de-biasing techniques on the accuracy and fairness measures of the classification of the Diabetes dataset.

According to the “figure11’, the reject option classification method considerably decreases the accuracy of classifiers, while using the reweighing or adversarial de-biasing techniques do not cause any significant reduction in the accuracy of classifiers.

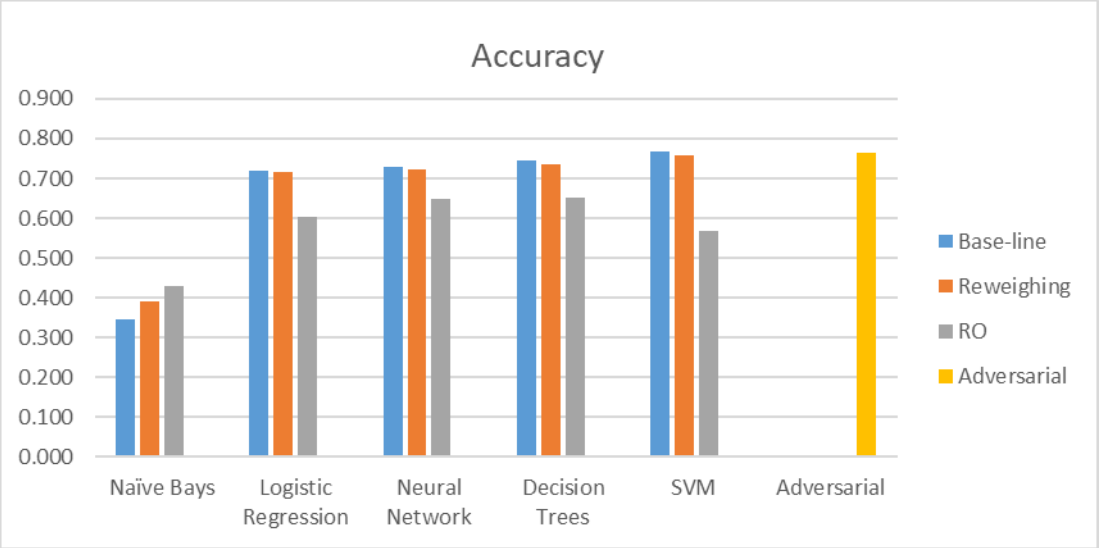


Figure 11) Comparison of de-biasing methods on the Diabetes Dataset regarding the accuracy measure

The Reject Option method successfully reduces the statistical parity difference with the Neural Network, Decision Tree, and the SVM classifiers. Regarding the Logistic Regression classifier, the reweighing method shows a better performance, while with the Naïve Bayes classifier, the best state is when there is no de-biasing method applied.

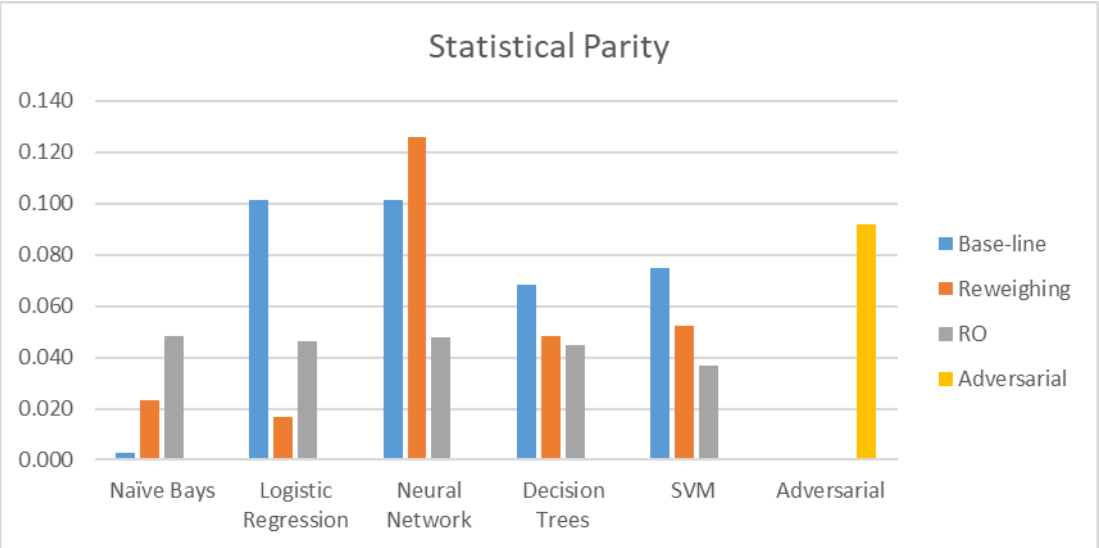


Figure 12) Comparison of de-biasing methods on the Diabetes Dataset regarding the Statistical Parity measure

Among the three different de-biasing methods tested, the reject option classification can reduce the equality of opportunity difference more than the other methods, except when a Logistic Regression or a Naïve Bayes classifier is used. In this case, the reweighing method is more effective.

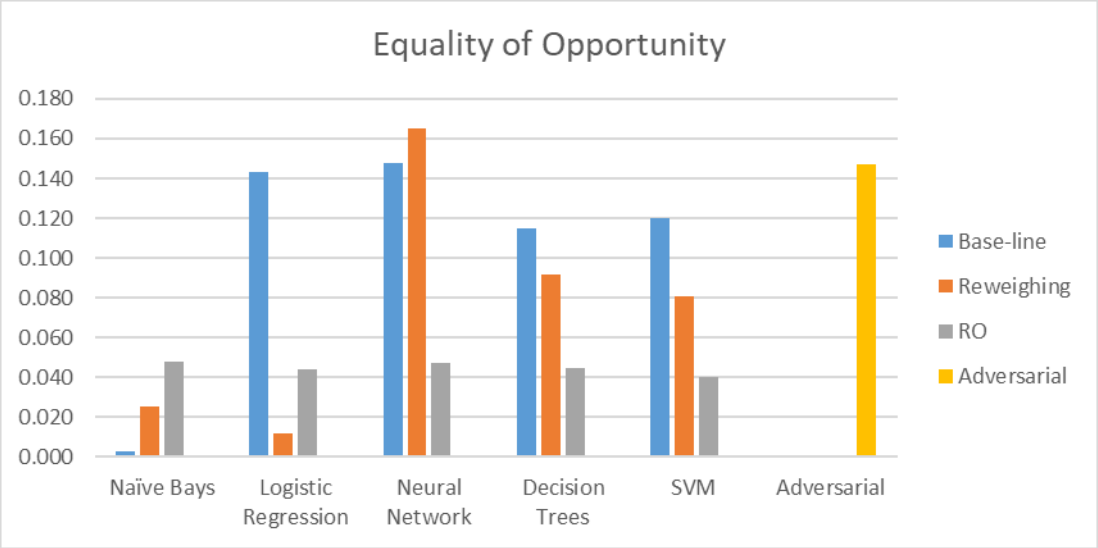


Figure 13) Comparison of de-biasing methods on the Diabetes Dataset regarding the Equality of Opportunity measure

For the Neural Network classifier, the reject option method is the most effective method to reduce odds difference. Using the Logistic Regression classifier, the reweighing process is the best way to reduce the average odd difference. Reject option and reweighing techniques work similarly in lowering the odds difference with the Decision Tree and the SVM classifiers. Naïve Bayes classifier has the lowest odds difference when no de-biasing methods are applied.

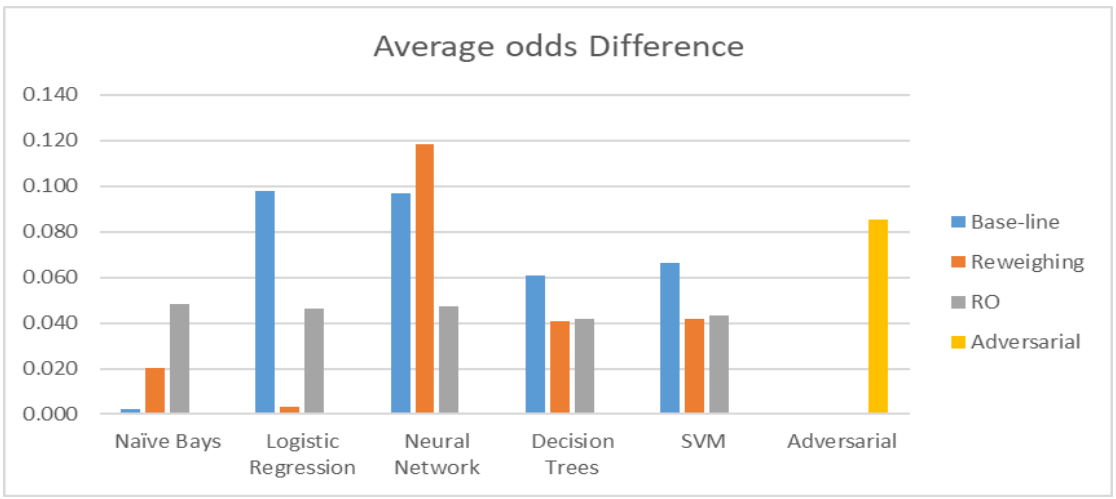


Figure 14) Comparison of de-biasing methods on the Diabetes Dataset regarding the Average odds Difference measure

As depicted in “Figure 15”, except for the Naïve Bayes classifier, the reject option method has a better performance in reducing the Theil Index. However, not a considerable change is observed with the Logistic Regression classifier.

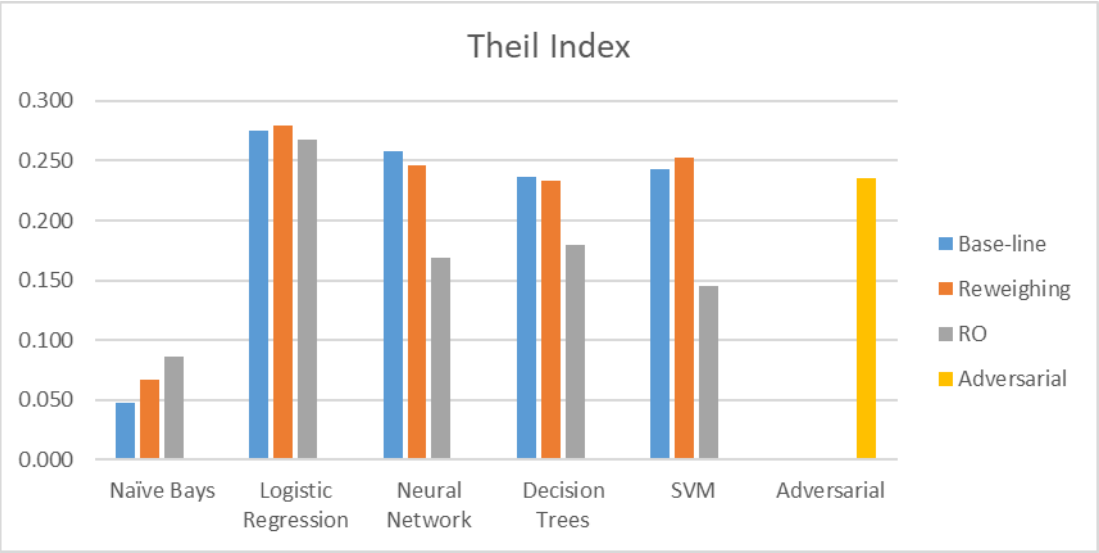


Figure 15) Comparison of de-biasing methods on the Diabetes Dataset regarding the Theil Index

4.3 The German Credit Dataset

We used the German Credit dataset available at the UCI machine learning repository (Dua & Graff, 2019). This data set includes the bank account information of 1000 cases and classifies them as good or bad credit account holders using a cost matrix. Professor Hans Hofmann from the University of Hamburg provided the original dataset in the year 1994. It is one of the most common datasets used in fairness researches. The dataset includes 13 categorical and seven numerical attributes as follows (Verma & Rubin, 2018).

1. Credit amount (numerical); 2. Credit duration (numerical); 3. Credit purpose (categorical); 4. Status of existing checking account (categorical); 5. Status of savings accounts and bonds (categorical); 6. Number of existing credits (numerical); 7. Credit history (categorical); 8. Installment plans (categorical); 9. Installment rate (numerical); 10. Property (categorical); 11. Residence (categorical); 12. Period of present residency (numerical); 13. Telephone (binary); 14. Employment (categorical); 15. Employment length (categorical); 16. Personal status and

gender (categorical); 17. Age (numerical); 18. Foreign worker (binary); 19. Dependents (numerical); 20. Other debtors (categorical); 21. Credit score (binary).

In this experiment, "age" is the protected attribute. In order to follow the previous studies done on this data set by F. Kamiran and Calders (2009) we chose the age of 25 as the discrimination threshold, which showed the highest rate of discrimination in Kamran's studies. Hence, (age <= 25) categorized as "Young", and (age > 25) classified as "Aged" are the unprivileged and the privileged groups, respectively.

4.3.1 Dataset Distribution

In the "German Credit Dataset", "Credit Score" is chosen as the target classification in which "Good Credit" is the positive label, and "Bad Credit" is the negative label. From the total 1000 instances, 700 cases are labeled as good, and 300 cases are labeled as bad credits (figure 16). If we predict all the labels as positive, accuracy would be 0.7, and balanced accuracy is 0.5. Considering the histogram of the age of costumers by their credit score (figure 17), although the number of young costumers is high in comparison with the aged people, the ratio of good credit to the bad credit people is much higher in the aged group (about 2.68) comparing young group (about 1.37). Therefore, there is a potential for bias against people age 25 or less. The initial disparate ratio of the dataset is 0.79.

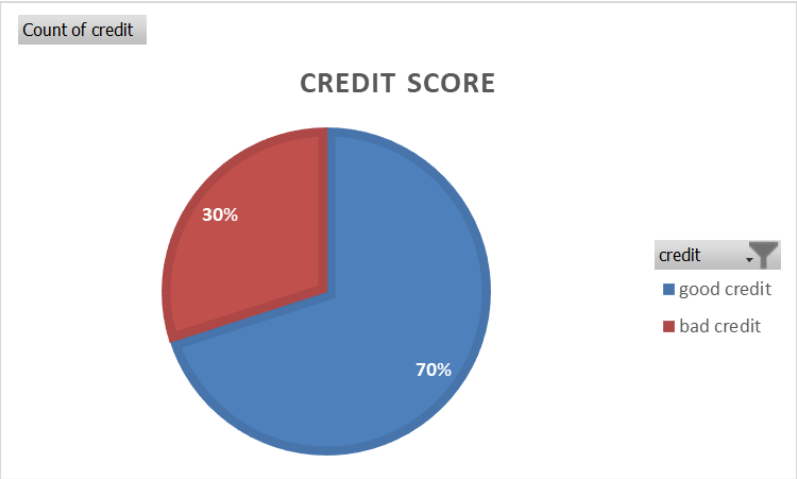


Figure 16) Distribution of account holders' credit score

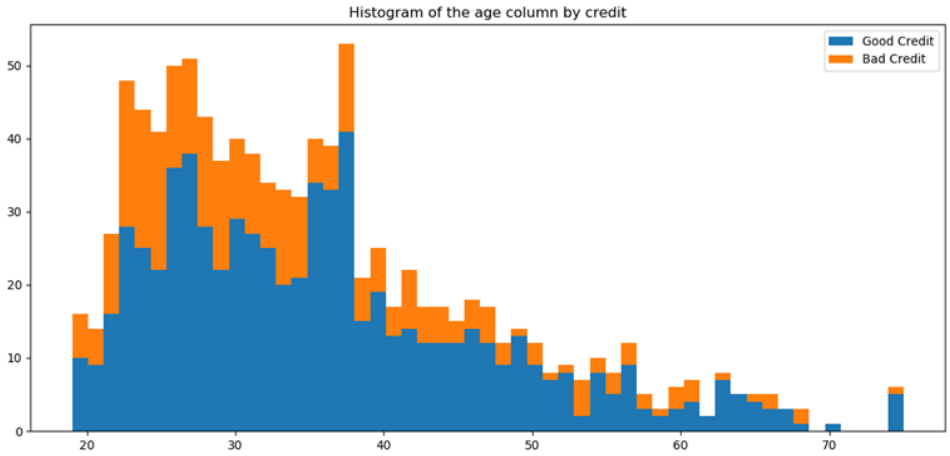


Figure 17) Histogram of the age column by credit

4.3.2 Classification Results of the Credit Dataset before De-Biasing

Five different classifiers with tuned hyper-parameters are tested on the dataset to get the highest possible accuracy. The evaluation metric results and the fairness measures before and after de-biasing with different methods are shown in the following sections.

The details of the hyper-parameters tuning of the German Credit dataset is as follows:

Naïve Bayes: Gaussian likelihood
Logistic regression: L-BFGS-B[14] solver, L2 penalty, the C=25 regularization parameter
Neural Networks: hidden layer of size 200, 500 times of iteration, adaptive learning rate and regularization parameter alpha=0.0001, "logistic" activation function, "Adam" optimizer
Decision Trees: "entropy" criterion, maximum depth of 6
SVM: "RBF" kernel, Regularization parameter C=100, and kernel coefficient gamma=0.001

The rest of the parameters are set as the "Scikit Learn" default settings.

Applying the five classifiers without any de-biasing process resulted in almost the same values of accuracy, including Naïve Bayes (0.67), Logistic Regression (0.751), Neural Network (0.750), Decision Tree (0.710), and the SVM classifier (0.758). The Decision Tree

[14] http://users.iems.northwestern.edu/~nocedal/lbfgsb.html

classifier shows the fairest result. It has the lowest values of the statistical parity, odds, and opportunity difference and the closest value of disparate impact ratio to one among the other classifiers. The Lowest Theil index belongs to the Naïve Bayes classifier. The values of the statistical parity, average odds, and the equal opportunity difference for all of the classifiers are negative, showing bias in the classification against the unprivileged group. There is a lower possibility for the unprivileged group to be labeled as positive.

Table 15) Evaluation of classifiers on Credit dataset before De-biasing

German Credit Data set											
De-biasing Method											
Classification Method											
		Naïve Bayes (Gaussian)		Logistic Regression		Neural Network		Decision Trees		SVM	
Evaluation Metrics	Accuracy	0.670		0.751		0.750		0.710		0.758	
	Precision	0.848		0.793		0.795		0.768		0.792	
	Recall	0.647		0.873		0.867		0.839		0.890	
	F1	0.724		0.830		0.829		0.801		0.837	
	AUC	0.685		0.670		0.672		0.624		0.670	
Fairness Metrics	Statistical Parity	-0.174		-0.212		-0.193		-0.151		-0.202	
	Equality of Opportunity	-0.161		-0.202		-0.178		-0.125		-0.157	
	Average odds Difference	-0.109		-0.145		-0.128		-0.104		-0.152	
	Disparate Impact Ratio	0.698		0.738		0.757		0.807		0.757	
	Theil Index	0.140		0.206		0.204		0.232		0.209	
confusion matrix		**N**	**P**	**N**	**P**	**N**	**P**	**N**	**P**	**N**	**P**
	N	453	247	611	89	607	93	587	113	623	77
	p	83	217	160	140	157	143	177	123	165	135

4.3.3 Classification Results after De-Biasing With Reweighing

According to the results shown in “table 16”, utilizing the reweighing de-biasing method has caused a slight reduction in the accuracy of the Logistic Regression (from 0.75 to 0.73) and Decision Tree (from 0.71 to 0.69) classifiers. Doing two-tail t-tests with a significance level of 0.05 yield p-values of 0.51 and 0.44, respectively. P-values less than the significance level show that the reduction of accuracy values after de-biasing is statistically insignificant.

Except for the Neural Network classifier, the reweighing method shows an acceptable performance in reducing all of the bias measures in the four left classifiers.

Table 16) Evaluation of classifiers on Credit dataset after De-biasing with Reweighing

German Credit Data set											
De-biasing Method: Reweighing											
Classification Method											
		Naïve Bayes (Gaussian)		Logistic Regression		Neural Network		Decision Trees		SVM (linear)	
Evaluation Metrics	Accuracy	0.714		0.736		0.751		0.695		0.753	
	Precision	0.822		0.784		0.810		0.749		0.794	
	Recall	0.756		0.864		0.842		0.853		0.877	
	F1	0.784		0.820		0.825		0.795		0.832	
	AUC	0.676		0.656		0.692		0.592		0.675	
Fairness Metrics	Statistical Parity	-0.049		-0.024		-0.227		0.002		-0.053	
	Equality of Opportunity	-0.050		-0.041		-0.152		-0.042		-0.020	
	Average odds Difference	0.019		0.049		-0.176		0.046		0.003	
	Disparate Impact Ratio	0.922		0.972		0.709		1.003		0.937	
	Theil Index	0.226		0.147		0.159		0.162		0.136	
confusion matrix		**N**	**P**	**N**	**P**	**N**	**P**	**N**	**P**	**N**	**P**
	N	531	169	604	96	589	111	596	104	613	87
	P	117	183	168	132	138	162	201	99	160	140

4.3.4 Classification Results after De-Biasing With Reject Option Classification

Reject Option Classification performs de-biasing on the data based on each required fairness metric separately. As a result, we repeated the experiment three times with this method to reduce “Statistical Parity”, “Equality of Opportunity”, and “Average of Odds

difference" separately. The results of applying the Reject Option classification method on the Credit dataset are shown in the following tables.

4.3.4.1 Reject Option-Statistical Parity Difference

We observe a considerable reduction in the accuracy value of all the classifiers except the Naïve Bayes. The details of these reductions are as follows. Three percent reduction in the accuracy of the Logistic Regression from 0.75 to 0.72 with a p-value of 0.27, two percent reduction in the Neural Network accuracy from 0.75 to 0.73 with a p-value of 0.56, six percent reduction is the accuracy of Decision Tree from 0.71 to 0.65 with a p-value of 0.07, and four percent reduction in the SVM accuracy value from 0.75 to 0.71 with a p-value of 0.14. As all of the p-values are less than the significance level (0.05), we can conclude that the decrease in accuracy values are statistically negligible.

As illustrated in the "table 17," all of the fairness measures with all of the five classifiers have been reduced after performing the Reject Option method with the use of statistical parity as the optimization method. However, the Theil index of all classifiers has increased, which represents the ineffectiveness of this method in reducing individual fairness.

Table 17) Evaluation of classifiers on Credit dataset after De-biasing with Reject Option (Statistical parity difference)

German Credit Data set											
De-biasing Method: Reject Option(Statistical parity difference)											
Classification Method											
		Naïve Bayes (Gaussian)		Logistic Regression		Neural Network		Decision Trees		SVM	
Evaluation Metrics	Accuracy	0.706		0.720		0.737		0.651		0.709	
	Precision	0.879		0.875		0.865		0.838		0.905	
	Recall	0.681		0.717		0.742		0.650		0.656	
	F1	0.761		0.773		0.793		0.707		0.746	
	AUC	0.731		0.735		0.735		0.659		0.741	
Fairness Metrics	**Statistical Parity**	-0.016		-0.016		-0.002		-0.011		-0.021	
	Equality of Opportunity	-0.004		0.021		-0.003		-0.002		0.026	
	Average odds Difference	0.064		0.062		0.093		0.050		0.062	
	Disparate Impact Ratio	0.970		0.969		0.998		0.955		0.958	
	Theil Index	0.285		0.261		0.233		0.333		0.305	
confusion matrix		**N**	**P**	**N**	**P**	**N**	**P**	**N**	**P**	**N**	**P**
	N	475	225	499	201	520	180	453	247	461	239
	P	69	231	79	221	83	217	102	198	52	248

4.3.4.2 Reject Option -Average Odds Difference

This section compares the results of classification with Reject Option using the average odd as the optimization method with the baseline statues when there is no de-biasing process.

Considering the results depicted in “table 18”, there is about a four percent increase in the Naïve Bayes classifier's accuracy. At the same time, its statistical parity, average odds,

and equal opportunity difference have decreased. With the Logistic Regression, Neural Network, and the SVM classifiers, there are about one percent (from 0.75 to 0.74), two percent (0.75 to 0.73), and three percent (from 0.76 to 0.0.73) reduction in the accuracy values, respectively. The p-values of these classifiers are 0.83, 0.57, and 0.42, representing the insignificance of these declines in the accuracy of classifiers. Regarding the Decision Tree classifier, there is about a six percent decrease in the accuracy from 0.71 to 0.65, which is not negligible (p-value of 0.029).

We observe a noticeable fall in statistical parity values, equal opportunity, and average odds difference in all of the classifiers after performing the Reject Option method with the average odds optimization. The disparate impact ratio of all classifiers has got closer to the ideal value of one. On the other hand, this de-biasing method does not successfully reduce the Theil index metric with any of the classifiers.

Table 18) Evaluation of classifiers on Credit dataset after De-biasing with Reject Option (Average odds difference)

German Credit Data set											
De-biasing Method: Reject Option (Average odds difference)											
Classification Method											
		Naïve Bayes (Gaussian)		Logistic Regression		Neural Network		Decision Trees		SVM	
Evaluation Metrics	Accuracy	0.712		0.746		0.737		0.652		0.734	
	Precision	0.879		0.861		0.877		0.854		0.893	
	Recall	0.690		0.768		0.725		0.618		0.707	
	F1	0.766		0.804		0.790		0.702		0.778	
	AUC	0.735		0.740		0.746		0.667		0.751	
Fairness Metrics	Statistical Parity	-0.098		-0.087		-0.100		-0.055		-0.082	
	Equality of Opportunity	-0.080		-0.087		-0.079		-0.032		-0.049	
	Average odds Difference	-0.018		0.007		-0.011		0.003		0.003	
	Disparate Impact Ratio	0.828		0.862		0.833		0.882		0.859	
	Theil Index	0.279		0.215		0.244		0.346		0.260	
confusion matrix		**N**	**P**	**N**	**P**	**N**	**P**	**N**	**P**	**N**	**P**
	N	481	219	536	164	508	192	435	265	496	204
	P	69	231	90	210	71	229	83	217	62	238

4.3.4.3 Reject Option - Equal Opportunity Difference

Performing the Reject Option classification method with the equal opportunity difference optimization has reduced the accuracy of the Logistic Regression, Neural Network, Decision Tree, and the SVM accuracy. To make sure the reduction in the accuracy values is negligible; we do a two-tail t-student test. The accuracy of classifiers and their p-

values are as follows. Logistic Regression accuracy reduction from 0.751 to 0.737 and p-value of 0.57, Neural Network accuracy reduction from 0.75 to 0.71 and p-value of 0.15, Decision Tree accuracy reduction from 0.71 to 0.63 and p-value of 0.004, and SVM classifier accuracy reduction from 0.76 to 0.73 and a p-value of 0.31. According to the mentioned results, the accuracy reduction is insignificant in all classifiers except for the Decision Tree. This method has successfully improved all of the fairness measures in all classifiers except for the Theil index.

Table 19) Evaluation of classifiers on Credit dataset after De-biasing with Reject Option (Equal opportunity difference)

German Credit Data set											
De-biasing Method: Reject Option (Equal opportunity difference)											
Classification Method											
		Naïve Bayes (Gaussian)		Logistic Regression		Neural Network		Decision Trees		SVM (linear)	
Evaluation Metrics	Accuracy	0.718		0.737		0.715		0.629		0.727	
	Precision	0.876		0.883		0.903		0.867		0.904	
	Recall	0.702		0.720		0.667		0.558		0.682	
	F1	0.773		0.786		0.760		0.669		0.768	
	AUC	0.735		0.744		0.746		0.669		0.756	
Fairness Metrics	Statistical Parity	-0.025		-0.055		-0.088		-0.052		-0.073	
	Equality of Opportunity	0.003		-0.009		-0.022		-0.015		-0.013	
	Average odds Difference	0.055		0.028		-0.008		0.005		0.005	
	Disparate Impact Ratio	0.966		0.901		0.839		0.884		0.869	
	Theil Index	0.267		0.247		0.290		0.399		0.279	
confusion matrix		**N**	**P**	**N**	**P**	**N**	**P**	**N**	**P**	**N**	**P**
	N	490	210	506	194	468	232	393	307	479	221
	p	72	228	69	231	53	247	64	236	52	248

4.3.5 Classification results after de-biasing with Adversarial de-biasing

The adversarial de-biasing classifier is a one layer Neural Network with 200 hidden units. The number of epochs and the batch size is 100 and 27, respectively. The adversary loss weight is set to the default of the aif360 toolkit that is 0.1 with exponential learning rate decay. Compared with the Neural Network classification results without the de-biasing technique, there is no reduction in the accuracy value as it has increased from 0.75 to 0.76. This method has slightly reduced some of the bias metrics, including statistical parity (from -0.193 to -0.164), average odds difference (from -0.128 to -0.098), opportunity difference (from -0.178 to -0.144), and Theil index (from 0.204 to 0.180).

Table 20) Evaluation of classifiers on Credit dataset after De-biasing with Adversarial de-biasing

German Dataset				
De-biased: Adversarial De-biasing				
Evaluation Metrics				
Accuracy	Precision	Recall	F1	AUC
0.76	0.81	0.85	0.83	0.70
Fairness Metrics				
Statistical Parity	Equality of Opportunity	Average odds Difference	Disparate Impact Ratio	Theil Index
-0.1641	-0.1442	-0.0985	0.7848	0.1808

confusion matrix		
	N	**P**
N	592	108
P	135	165

4.3.6 Comparison of De-Biasing Methods on the German Credit Dataset

Among all the de-biasing methods, the adversarial approach shows the highest accuracy, while the Reject Option method causes the most reduction in the classifiers' accuracy values.

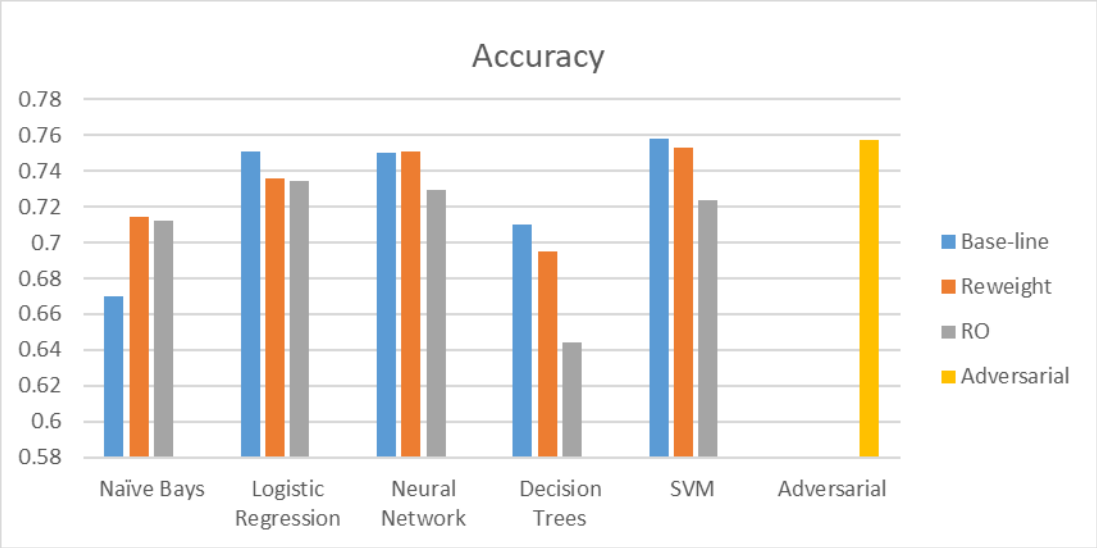


Figure 18)Comparison of de-biasing methods on the German Credit Dataset regarding the accuracy measure

The Reject Option method is the most effective in reducing the statistical parity difference except for the Decision Tree classifier, in which the reweighing technique shows a better performance.

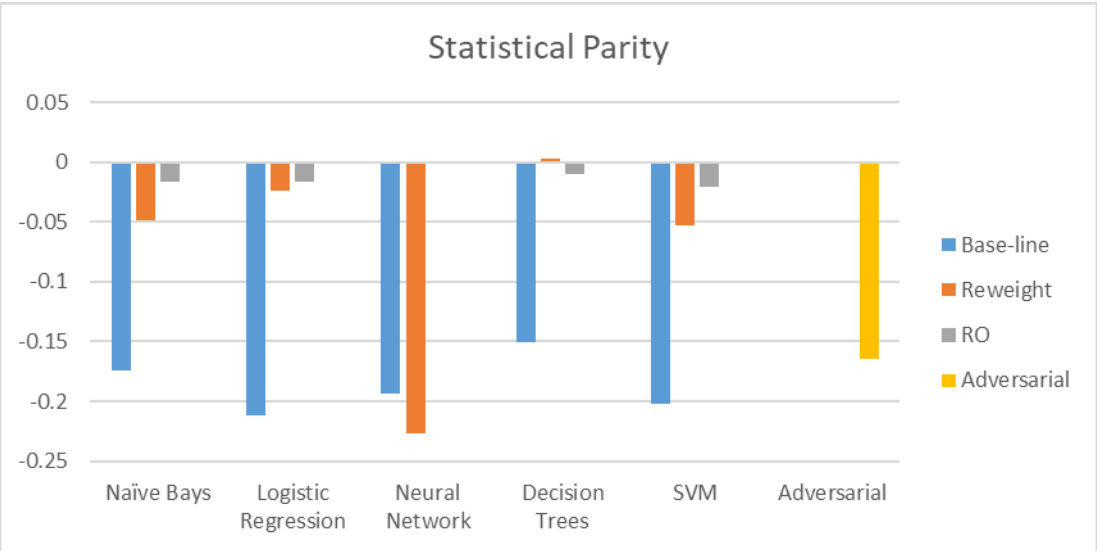


Figure 19) Comparison of de-biasing methods on the German Credit Dataset regarding statistical parity measure

As "Figure 20" shows, the lowest value of opportunity difference is related to the reject option method in all of the classifiers.

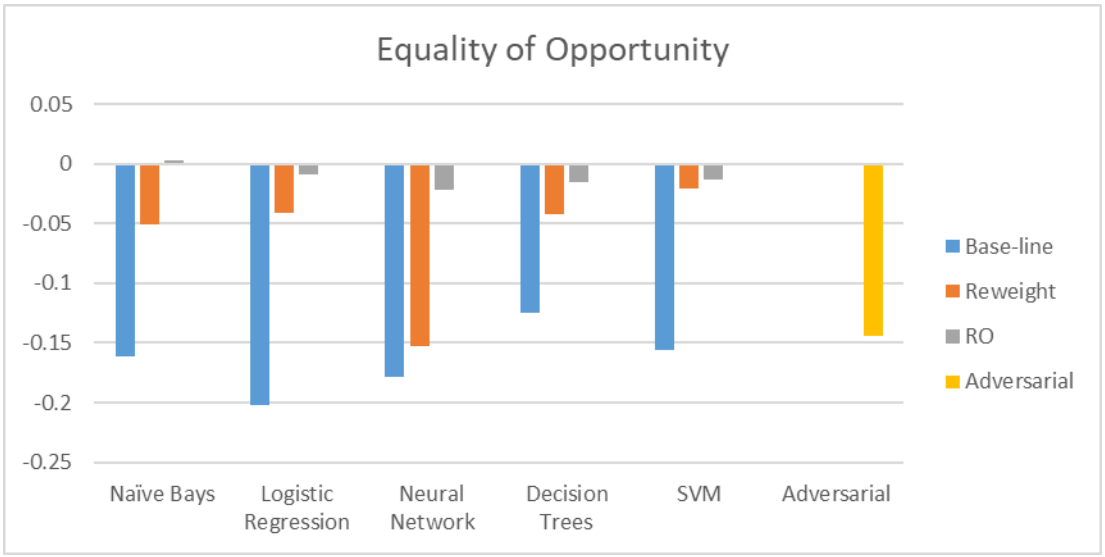


Figure 20) Comparison of de-biasing methods on the German Credit Dataset regarding Equality of Opportunity measure

Same as the parity and the opportunity difference, the reject option method has the most effect on reducing the average odds difference with all of the classifiers.

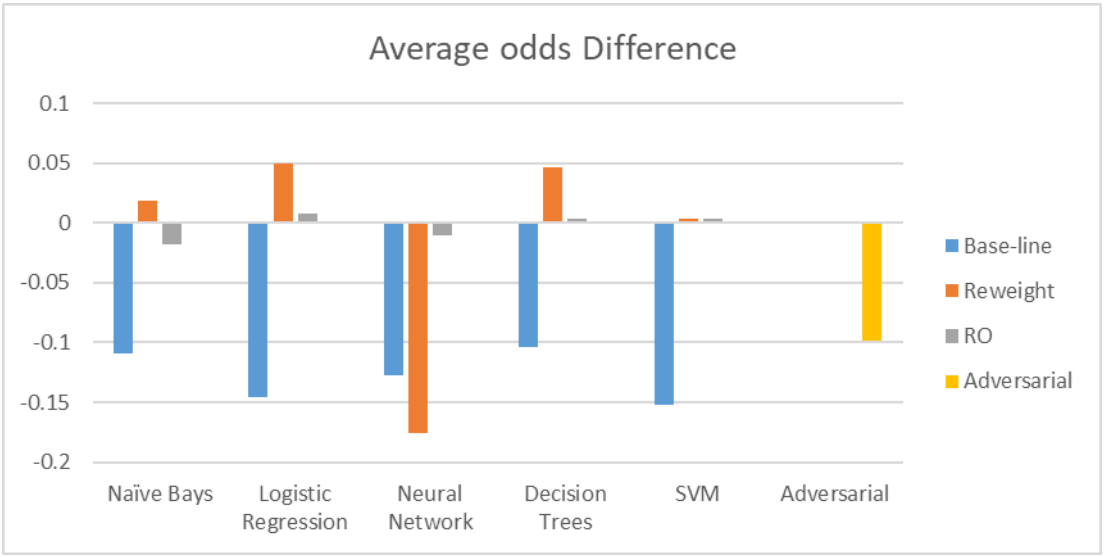


Figure 21) Comparison of de-biasing methods on the German Credit Dataset regarding Average odds Difference measure

Except with the Naïve Bayes classifier, which its lowest Theil index is obtained when there is no de-biasing, the reweighing method is able to reduce the Theil index in all other

classifiers. The adversarial de-biasing technique successfully reduces the Theil index of the Neural Network, but its performance is not as good as the reweighing method.

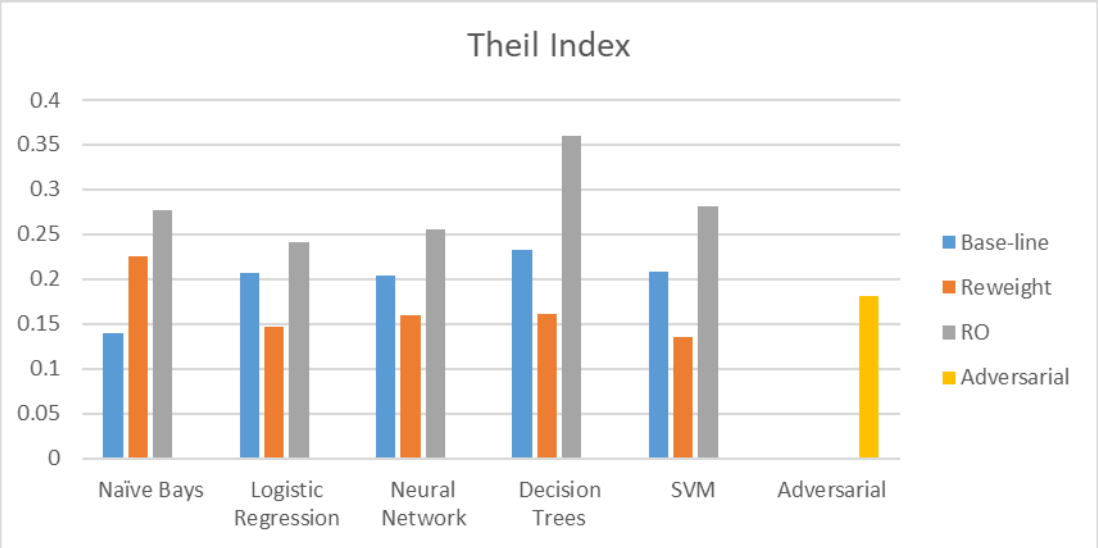


Figure 22) Comparison of de-biasing methods on the German Credit Dataset regarding Theil Index measure

4.5 CAMH Mental Health and the Covid-19 Pandemic Dataset

This dataset is collected in a series of national surveys among Canadians conducted by CAMH (The Centre for Addiction and Mental Health) in collaboration with the research technology company Delvinia[15] to investigate the impact of the COVID 19 pandemic on the mental health of people. The dataset contains 4015 instances collected in 4 different periods during the outbreak of COVID 19 from the 8th of May 2020 to the 12th of July 2020. The survey is based on English speaker Canadians' responses aged 18 or above via online surveys of the Asking Canadians web panel ("COVID-19 National Survey Dashboard,"). The quota sampling method was used to collect the data in a way that the distribution by gender, age, and region represents the distribution of English speaking Canadians ("COVID-19 National Survey Dashboard,"). The dataset includes biographic information, the impact of the pandemic on the respondents' financial and social lives, and the symptoms of anxiety, binge drinking, loneliness, and depression. The details of the features of the CAMH dataset are shown in the Appendix section.

As shown in the above-mentioned table, the questioner includes some information of the demographic data, the effects of COVID pandemic on candidates' personal, job and social lives, financial and residential situation, and use of drinks drugs, feelings of anxiety, loneliness, and depression. We remove four out of 37 features due to a high number of missing values, including Q16 (1625 missing values), Q17 (447 missing values), Q19 (889 missing values), and Q6b (1636 missing values).

We chose moderate or severe anxiety as the target label, measured with the 7-item Generalized Anxiety Disorder (GAD7) scale based on the proposed framework by Spitzer, Kroenke, Williams, and Löwe (2006). According to this research, the scores of 10 or above out of 21 show moderate or severe anxiety. The generalized anxiety disorder 7-item (GAD-7) scale is shown in the following table.

[15] https://www.delvinia.com/camh-coronavirus-mental-health/

Table 21) the generalized anxiety disorder 7-item (GAD-7) scale

Over the last two weeks, how often have you been bothered by the following problems?	Not at all	Several days	More than half the days	Nearly every day
1. Feeling nervous, anxious, or on edge	0	1	2	3
2. Not being able to stop or control worrying	0	1	2	3
3. Worrying too much about different things	0	1	2	3
4. Trouble relaxing	0	1	2	3
5. Being so restless that it is hard to sit still	0	1	2	3
6. Becoming easily annoyed or irritable	0	1	2	3
7. Feeling afraid as if something awful might happen	0	1	2	3

Note. From Spitzer, R. L., Kroenke, K., Williams, J. B. W., & Löwe, B. (2006). A Brief Measure for Assessing Generalized Anxiety Disorder: The GAD-7. Archives of Internal Medicine, 166(10), 1092-1097. doi:10.1001/archinte.166.10.1092 % Archives of Internal Medicine

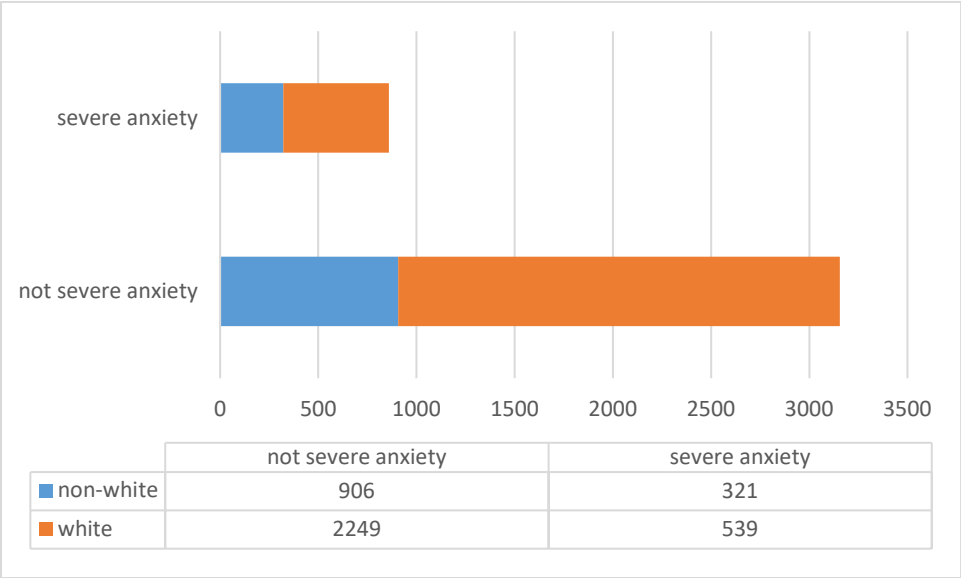
To prepare our target label, we calculated the summation of the values of the anxiety-related features (Q8×1 to Q8×7). Instances with scores less than ten are labeled as 0 (not severe anxiety), and samples with scores ten or higher are labeled as 1 (severe anxiety). We replaced the "anxiety" label with the seven anxiety features. The final dataset includes 27 features.

Our goal is to predict candidates' anxiety levels based on their personal life situation and the influence of the COVID pandemic on their lives. Moreover, we are going to investigate whether the classification is biased toward the race of people. Severe anxiety is considered the desired label, and the privileged group is the "white" race against the "non-white".

4.5.1 Dataset Distribution

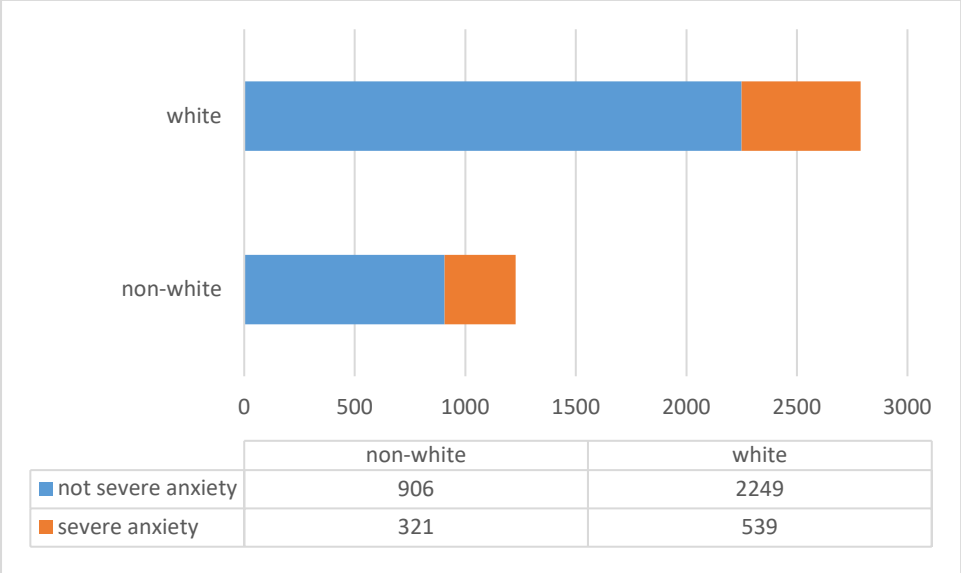
We consider severe anxiety as a positive label. From the 4015 instances, there are 860 positive and 3155 negative cases. The dataset includes 1227 non-white and 2788 white candidates. If we classify all of the instances as positive, the accuracy and balanced accuracy of classification would 0.27 and 0.5, respectively. As the figure23 shows, 26% of the non-white patients and 19% of the white cases are labeled with severe anxiety. While according to figure 24, 71% of the cases without severe anxiety and 63% of the cases with severe anxiety are from the white group.

The initial disparate impact of the unprivileged group to the privileged group is 1.35, which has little difference from the ideal impact ratio of one. There is a possibility of bias in the original dataset.



	not severe anxiety	severe anxiety
non-white	906	321
white	2249	539

Figure 23) bar plot of race groups by the "anxiety" label



	non-white	white
not severe anxiety	906	2249
severe anxiety	321	539

Figure 24) bar plot of "anxiety" label by race groups

4.5.2 Classification Results of the Credit Dataset before De-Biasing

Five different classifiers with tuned hyperparameters are tested on the dataset to get the highest possible accuracy. The results of the evaluation metrics and the fairness measures before and after de-biasing with different methods are shown in the following sections.

The details of the hyperparameter tuning of the CAMH dataset is as follows:

Naïve Bayes: Gaussian likelihood
Logistic regression: "liblinear" solver, L1 penalty, the C=0.09 regularization parameter
Neural Networks: hidden layer of size 200, 500 times of iteration, constant learning rate and regularization parameter alpha=0.001, "logistic" activation function, "Adam" optimizer
Decision Trees: "Gini" criterion, maximum depth of 2
SVM: "linear" kernel, Regularization parameter C=1, and kernel coefficient gamma=1

The rest of the parameters are set as the "Scikit Learn" default settings.

Applying five different classifiers of "Naïve Bayes", "Logistic Regression", "Neural Network", "Decision Tree", and "Support vector Machine" on the CAMH data set with no debiasing method yielded almost the same value of accuracy, including 0.80, 0.86, 0.86, 0.85 and 0.86 respectively. The "Decision Tree" classifier shows the least biased results as it has the best results in three of the fairness metrics, including statistical parity difference (0.048), odds difference (0.001), and disparate impact ratio (1.270). The lowest opportunity difference belongs to the SVM classifier, and Naïve Bayes have the best Theil index. The positive values of the fairness measures show that the classification results on this dataset are biased in favor of the unprivileged group. Although the unprivileged race has a smaller proportion of the instances, there is a higher probability of it being predicted as the positive label.

Table 22) Evaluation of classifiers on CAMH dataset before De-biasing

CAMH Data set											
De-biasing Method											
Classification Method											
		Naïve Bayes (Gaussian)		Logistic Regression		Neural Network		Decision Trees		SVM	
Evaluation Metrics	Accuracy	0.801		0.861		0.860		0.858		0.861	
	Precision	0.527		0.726		0.717		0.685		0.719	
	Recall	0.760		0.567		0.574		0.628		0.576	
	F1	0.621		0.636		0.637		0.654		0.638	
	AUC	0.786		0.754		0.756		0.774		0.757	
Fairness Metrics	Statistical Parity	0.122		0.070		0.071		0.048		0.060	
	Equality of Opportunity	0.058		0.053		0.038		-0.019		0.013	
	Average odds Difference	0.075		0.042		0.037		0.001		0.021	
	Disparate Impact Ratio	1.454		1.507		1.496		1.270		1.411	
	Theil Index	0.095		0.115		0.114		0.105		0.114	
confusion matrix		**N**	**P**	**N**	**P**	**N**	**P**	**N**	**P**	**N**	**P**
	N	2562	593	2969	186	2958	197	2905	250	2960	195
	p	206	654	372	488	366	494	320	540	365	495

4.5.3 Classification Results after De-Biasing With Reweighing

The classification results on the CAMH dataset after applying the reweighing technique are shown in table 24. There is a slight reduction in the accuracy of the Naïve Bayes (from 0.801 to 0.798), Logistic Regression (from 0.861 to 0.859), and the SVM (from 0.861 to 0.858) classifiers. The difference between the accuracy of the classifiers before and after reweighing is statistically insignificant as two-tail t-tests yield p-values of 0.79 for the

naïve Bayes classifier, 0.84 for the Logistic Regression, and 0.79 for the SVM classifier, which are not greater than the significant level of 0.001 or 0.05. The reweighing technique has successfully reduced bias measures in the Naïve Bayes, Logistic Regression, and the Neural Network classifiers. With the Decision Trees classifier, all fairness measures have improved except for the Statistical Parity difference, which has increased from 0.048 to 0.051. Except for the Statistical parity difference (from 0.06 to 0.021), reweighing did not reduce other bias measures with the SVM classifier.

Table 23) Evaluation of classifiers on Credit dataset after De-biasing with Reweighing

CAMH Data set											
De-biasing Method: Reweighing											
Classification Method											
		Naïve Bayes (Gaussian)		Logistic Regression		Neural Network		Decision Trees		SVM	
Evaluation Metrics	Accuracy	0.798		0.859		0.861		0.858		0.858	
	Precision	0.520		0.721		0.714		0.683		0.714	
	Recall	0.778		0.562		0.584		0.629		0.569	
	F1	0.623		0.629		0.641		0.653		0.631	
	AUC	0.791		0.751		0.760		0.775		0.754	
Fairness Metrics	Statistical Parity	0.092		0.036		0.068		0.051		0.021	
	Equality of Opportunity	0.008		-0.035		0.032		-0.015		-0.074	
	Average odds Difference	0.038		-0.010		0.033		0.003		-0.035	
	Disparate Impact Ratio	1.337		1.247		1.481		1.319		1.160	
	Theil Index	0.092		0.117		0.112		0.105		0.116	
confusion matrix		**N**	**P**	**N**	**P**	**N**	**P**	**N**	**P**	**N**	**P**
	N	2537	618	2969	186	2955	200	2905	250	2958	197
	p	192	668	380	480	359	501	320	540	373	487

4.5.4 Classification Results after De-Biasing With Reject Option Classification

Reject Option Classification performs de-biasing on the data based on each required fairness metric separately. As a result, we repeated the experiment three times with this method to reduce “Statistical Parity”, “Equality of Opportunity”, and “Average of Odds difference” separately. The results of applying the Reject Option classification method on the CAMH dataset are shown in the following tables.

4.5.4.1 Reject Option-Statistical Parity Difference

Performing the reject option classification method on the CAMH dataset has caused a reduction in the accuracy value of the Logistic Regression (from 0.86 to 0.79), Neural Networks (from 0.86 to 0.77), Decision Trees (from 0.85 to 0.703), and the SVM (from 0.86 to 0.83). To ensure that the accuracy value reduction is not significant and is negligible, we did two-tail t-tests that yielded to p-values of 0.27, 0.16, 0.09, and 0.014, respectively. It shows that the difference between accuracy values before and after debiasing with the reject option classification is insignificant except for the SVM classifier. Statistical parity difference, opportunity, and odds difference have reduced in all classifiers using this method, but the disparate impact and the Theil index have no considerable change.

Table 24) Evaluation of classifiers on Credit dataset after De-biasing with Reject Option-Statistical parity difference

CAMH Data set											
De-biasing Method: Reject Option-Statistical parity difference											
Classification Method											
		Naïve Bayes (Gaussian)		Logistic Regression		Neural Network		Decision Trees		SVM	
Evaluation Metrics	Accuracy	0.826		0.798		0.770		0.703		0.833	
	Precision	0.633		0.683		0.642		0.569		0.681	
	Recall	0.491		0.577		0.617		0.705		0.547	
	F1	0.541		0.565		0.529		0.543		0.554	
	AUC	0.705		0.717		0.717		0.702		0.730	
Fairness Metrics	**Statistical Parity**	0.059		0.044		0.042		0.016		0.038	
	Equality of Opportunity	0.049		-0.005		0.007		-0.037		-0.022	
	Average odds Difference	0.037		0.008		0.014		-0.016		-0.001	
	Disparate Impact Ratio	1.443		1.386		1.444		1.166		1.431	
	Theil Index	0.140		0.117		0.114		0.094		0.127	
confusion matrix		**N**	**P**	**N**	**P**	**N**	**P**	**N**	**P**	**N**	**P**
	N	2901	254	2708	447	2566	589	2212	943	2879	276
	p	443	417	365	495	334	526	251	609	394	466

4.5.4.2 Reject Option -Average Odds Difference

According to table 26, there is a reduction in the accuracy of all of the classifiers, including about 2% in Naïve Bayes, 9% in the Logistic regression, 4% in the Neural Networks, 12% in the Decision Trees, and 4% in the SVM classifier. Performing the t-test on the accuracy values of the classifiers before and after de-biasing resulted in p-values of 0.50, 0.11, 0.005, 0.076, 0.018, respectively. It means that the reduction of accuracy is not

negligible in the Neural Networks and the SVM classifiers. Applying the Reject Option classification technique with the Average odds difference as the optimization metric has improved the average odds difference slightly in all classifiers except for the Decision Trees and the SVM classifiers.

Table 25) Evaluation of classifiers on Credit dataset after De-biasing with Reject Option- Average odds difference

CAMH Data set											
De-biasing Method: Reject Option- Average odds difference											
Classification Method											
		Naïve Bayes (Gaussian)		Logistic Regression		Neural Network		Decision Trees		SVM	
Evaluation Metrics	Accuracy	0.786		0.773		0.825		0.734		0.826	
	Precision	0.523		0.547		0.577		0.566		0.575	
	Recall	0.785		0.813		0.802		0.677		0.823	
	F1	0.613		0.628		0.656		0.526		0.670	
	AUC	0.787		0.787		0.815		0.712		0.825	
Fairness Metrics	Statistical Parity	0.113		0.098		0.090		0.032		0.088	
	Equality of Opportunity	-0.007		-0.003		0.009		-0.007		-0.014	
	Average odds Difference	0.046		0.035		0.034		0.003		0.024	
	Disparate Impact Ratio	1.403		1.367		1.365		1.264		1.332	
	Theil Index	0.090		0.079		0.080		0.104		0.077	
confusion matrix		**N**	**P**	**N**	**P**	**N**	**P**	**N**	**P**	**N**	**P**
	N	2484	671	2405	750	2617	538	2357	798	2609	546
	p	187	673	161	699	166	694	271	589	151	709

4.5.4.3 Reject Option - Equal Opportunity Difference

The reject option method with the equal opportunity difference as the optimization metric has led to a considerable reduction in classifiers' accuracy. As it is depicted in table 26, Naïve Bayes has a reduction of 3% in accuracy to 0.77, Logistic Regression with 4% reduction to 0.82, Neural Network with 4% reduction to 0.82, Decision Trees with about 11% fall to 0.74, and the SVM with about 4% decrease to 0.82. The p-values resulted from t-tests on the accuracy of classifiers in order are: 0.25, 0.003, 0.0026, 0.000025, 0.016. In other words, the reduction of accuracy values is statistically significant for all of the classifiers except for the Naïve Bayes.

Table 26) Evaluation of classifiers on CAMH dataset after De-biasing with Reject Option- Equal opportunity difference

CAMH Data set											
De-biasing Method: Reject Option- Equal opportunity difference											
Classification Method											
		Naïve Bayes (Gaussian)		Logistic Regression		Neural Network		Decision Trees		SVM	
Evaluation Metrics	Accuracy	0.771		0.820		0.819		0.741		0.818	
	Precision	0.501		0.566		0.557		0.468		0.562	
	Recall	0.835		0.827		0.852		0.840		0.846	
	F1	0.616		0.662		0.669		0.579		0.668	
	AUC	0.795		0.821		0.831		0.776		0.828	
Fairness Metrics	Statistical Parity	0.157		0.134		0.138		0.096		0.121	
	Equality of Opportunity	0.019		0.022		0.027		0.015		0.026	
	Average odds Difference	0.084		0.067		0.070		0.045		0.060	
	Disparate Impact Ratio	1.504		1.497		1.497		1.385		1.449	
	Theil Index	0.082		0.077		0.073		0.084		0.073	
confusion matrix		**N**	**P**	**N**	**P**	**N**	**P**	**N**	**P**	**N**	**P**
	N	2377	778	2576	579	2556	599	2250	905	2555	600
	p	142	718	145	715	128	732	135	725	131	729

4.5.5 Classification Results after De-Biasing With Adversarial De-Biasing

The classifier used for the adversarial de-biasing method on the CAMH dataset is a one layer Neural Network with 200 hidden units. The number of epochs and the batch size is 100 and 36, respectively. The adversary loss weight is set to the default of the aif360 toolkit that is 0.1 with exponential learning rate decay. In comparison with the Neural Network classification results without the de-biasing technique, there are reductions in the evaluation metrics, including a two percent reduction in accuracy from 0.86 to 0.84, which is statistically significant (p-value of a two-tail t-test with a significance level of 0.05 is equal to 0.01). According to table 28, there is a reduction in the equality of opportunity (from 0.038 to 0.003) and average odds difference (from 0.037 to 0.029), but there is no improvement in the other fairness metrics.

Table 27) Evaluation of classifiers on CAMH dataset after De-biasing with Adversarial Learning

CAMH Data set				
De-biased: Adversarial De-biasing				
Evaluation Metrics				
Accuracy	Precision	Recall	F1	AUC
0.840	0.647	0.573	0.602	0.743
Fairness Metrics				
Statistical Parity	Equality of Opportunity	Average odds Difference	Disparate Impact Ratio	Theil Index
0.077	0.003	0.029	1.502	0.120

confusion matrix		
	N	**P**
N	2881	274
P	367	493

4.5.6 Comparison of De-Biasing Methods on the CAMH Dataset

This section compares the results of the classification of the CAMH dataset with different de-biasing techniques based on the accuracy and fairness measures. As “figure 25" shows, the reweighing method has the least effect on classification accuracy. In contrast, the

Reject Option classification method has led to a considerable reduction in classifiers' accuracy, especially Decision Trees. The adversarial learning technique shows higher accuracy than the Naïve Bayes classifier, but its accuracy on average is about two percent than the other classifiers.

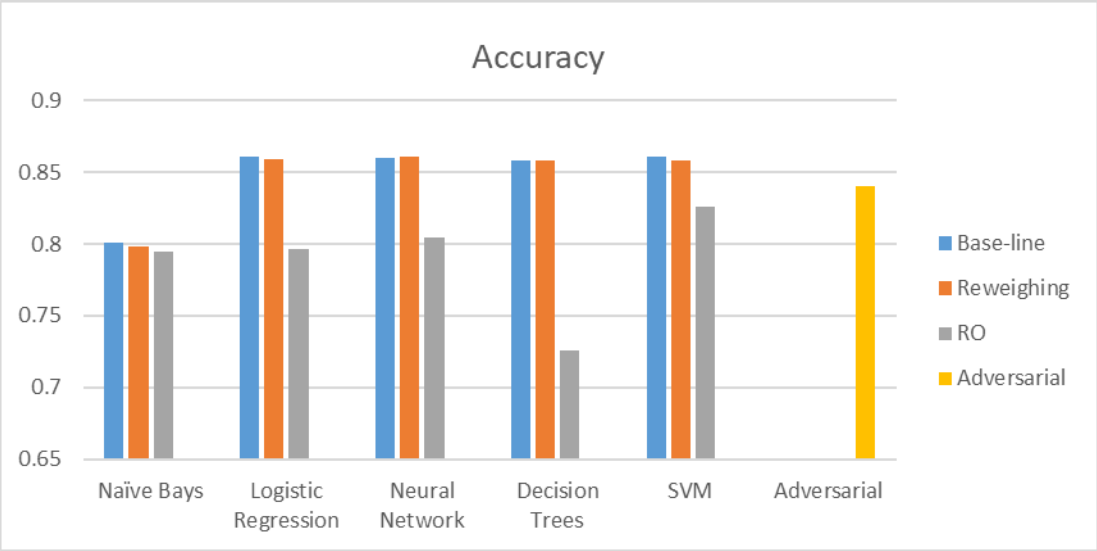


Figure 25) Comparison of de-biasing methods on the CAMH Dataset regarding the accuracy measure

Generally, the Reject Option classification method is more successful in reducing the statistical parity difference except for the SVM and Logistic Regression classifiers that reweighing method shows a better performance.

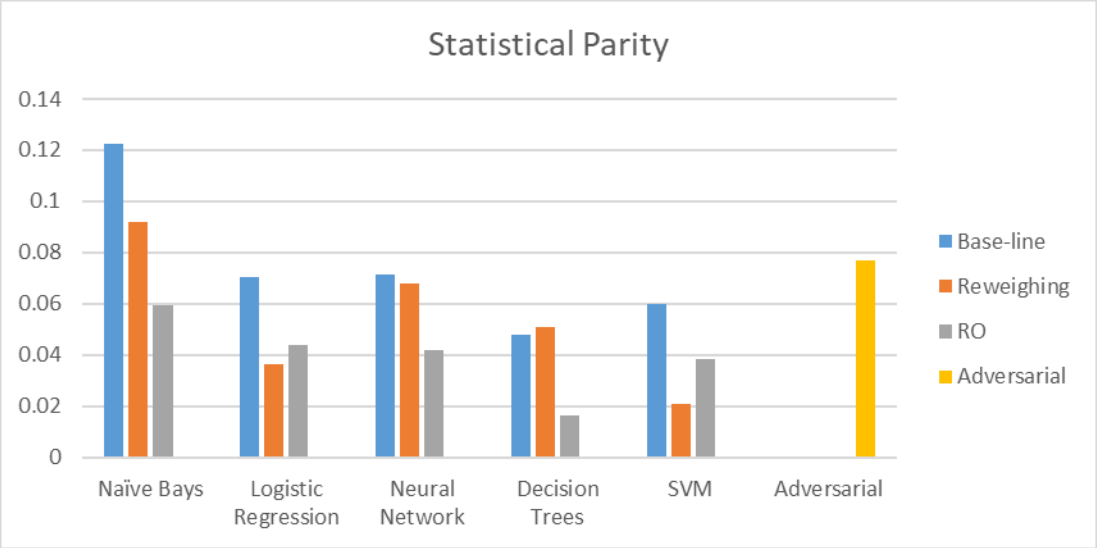


Figure 26) Comparison of de-biasing methods on the CAMH Dataset regarding the statistical parity difference

As the ideal value of equal opportunity difference is zero, we can conclude that the Adversarial Learning method shows a better result for this metric than the other techniques. While the reweighing technique is not successful with the SVM classifier in reducing the opportunity difference.

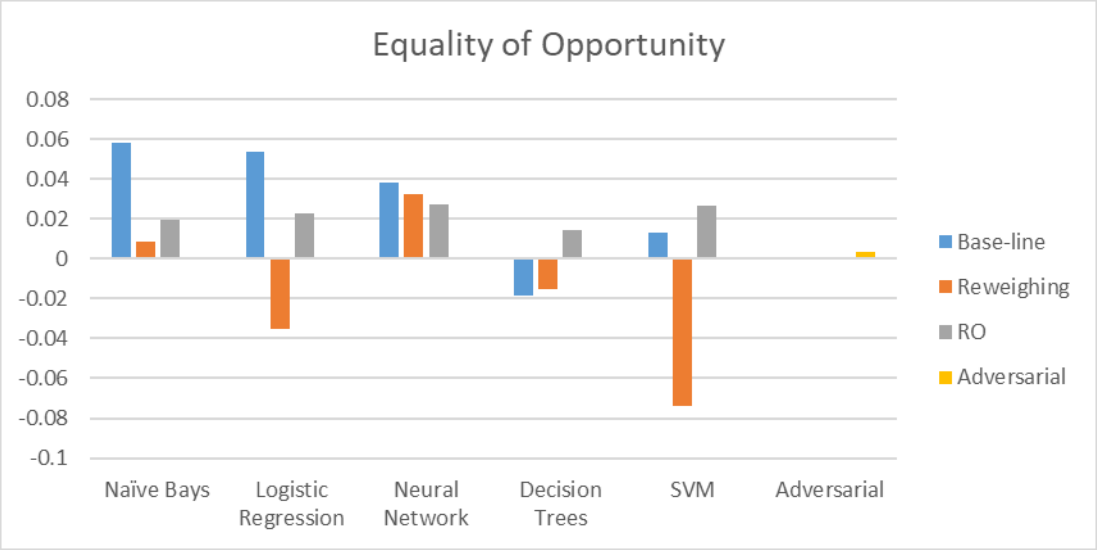


Figure 27) Comparison of de-biasing methods on the CAMH Dataset regarding the equal opportunity difference

Regarding the average odds difference reweighing shows better results than the other methods except for the SVM classifier, which none of the techniques were able to reduce the odds difference.

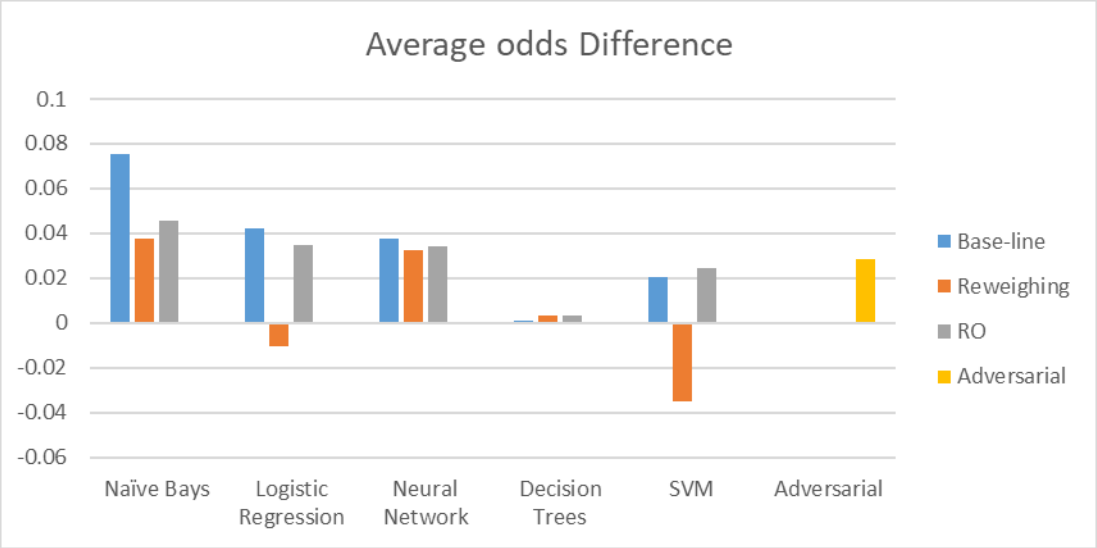


Figure 28) Comparison of de-biasing methods on the CAMH Dataset regarding the average odds difference

According to figure 29, except for the Naïve Bayes classifier, the reject option method is the only one that was able to reduce the Theil index measure in the CAMH dataset.

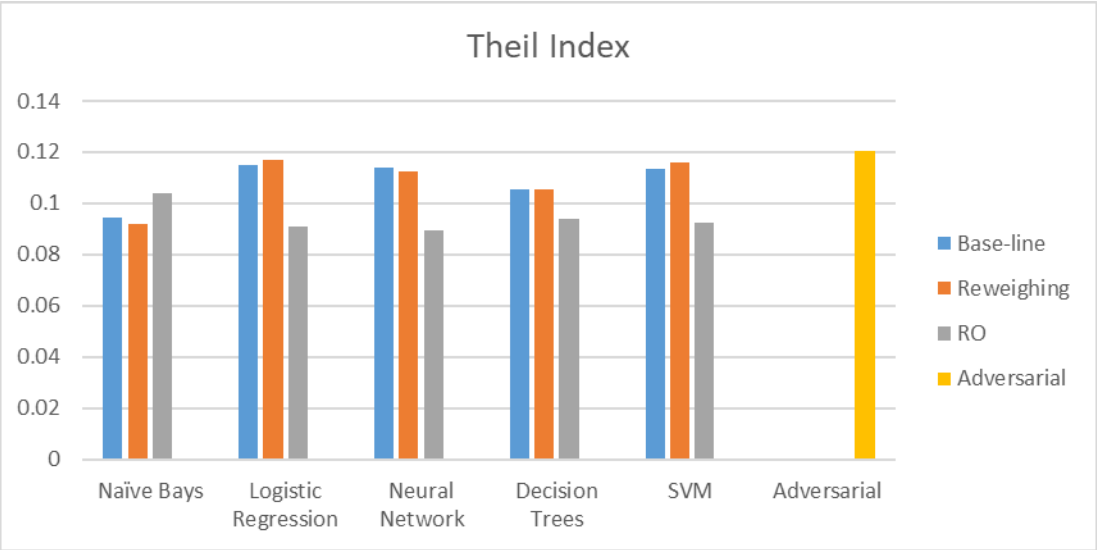


Figure 29) Comparison of de-biasing methods on the CAMH Dataset regarding the Theil index difference

5 Analysis of the Results

In this section, we are going to summarize the results gain through experiments on different datasets. Generally, the results show that the de-biasing methods are sensitive to the choice of classifiers and the changes in the datasets, so it is not possible to define the best de-biasing practices for all projects. However, each of the methods has unique characteristics which make it distinguishable. The following figures provide a compressed representation of the test results in the previous chapter. (In the figures, the combination of classifiers and de-biasing methods are shown in their contracted forms as "NB", "LR", "NN", "DT", "SVM" stand for "Naïve Bayes", "Logistic Regression", "Neural Network", "Decision Tree", and "Support Vector Machine". "No", "RW", "RO", and "Adv" represent "No de-bising", "Reweighing", "Reject Option", and "Adversarial de-biasing" approach, respectively.)

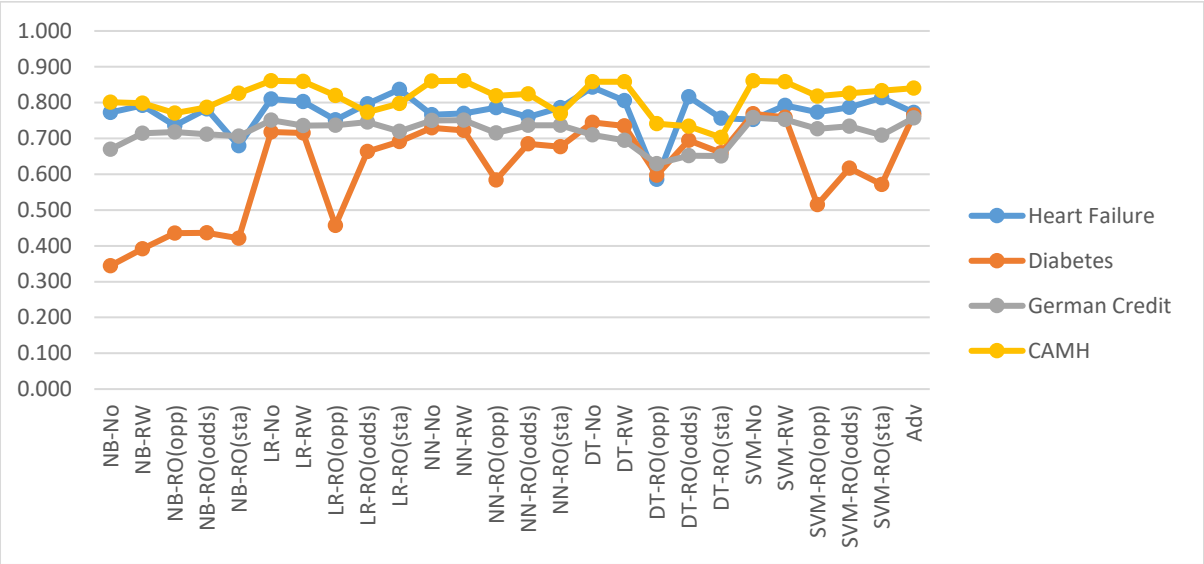


Figure 30) Comparison of Accuracy Measures in all Datasets

According to figure 30, most accuracy reduction is related to the reject option method, mainly when used with the Decision Tree classifier. There is a 25% decrease in accuracy of Decision Tree classifier in "Heart Failure dataset" when reject option method is used with the opportunity difference as the objective function. This de-biasing method, in collaboration with the Decision Tree classifier, has led to a 14%, eight percent, and 11% reduction of

accuracy in the Diabetes, German Credit, and CAMH datasets. The diabetes dataset shows the most sensitivity to the reject option method in comparison to the other datasets. The combination of the reject option method with logistic regression, neural network, and the SVM classifiers has caused a considerable drop in the accuracy of the Diabetes dataset. Considering figure 30, the least reduction in the accuracy rates in datasets are related to the combination of reweighing method with the logistic regression, Neural Network and support vector classifiers, and also the adversarial de-biasing technique.

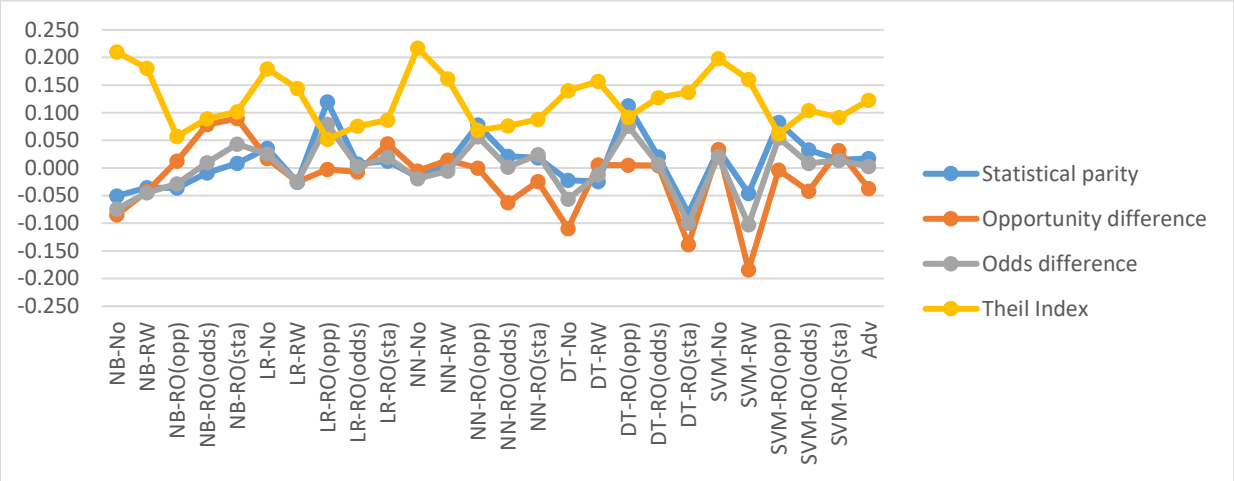


Figure 31) Comparison of fairness measures in the Heart Failure dataset

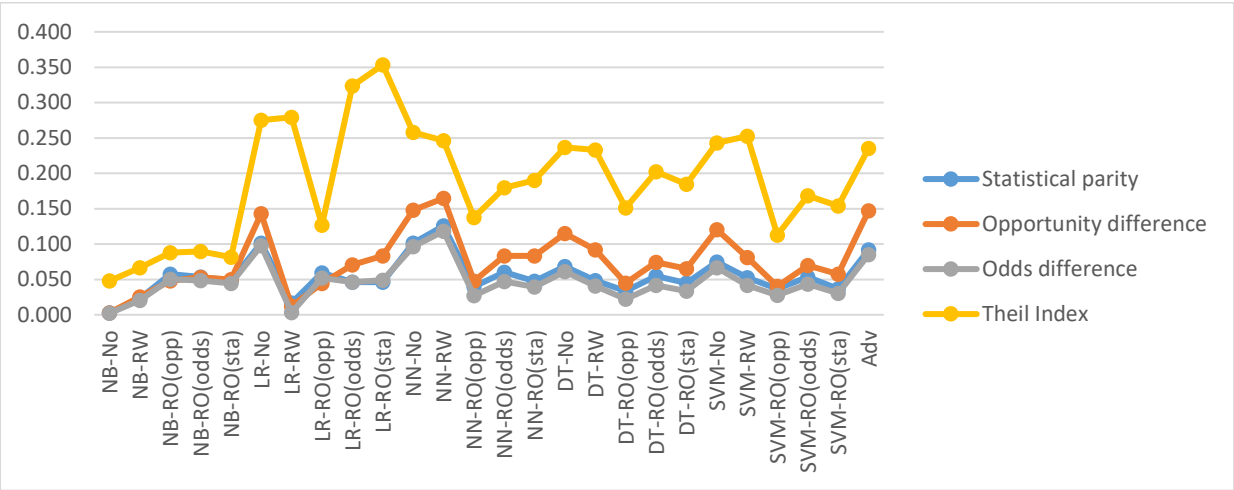


Figure 32) Comparison of fairness measures in the Diabetes dataset

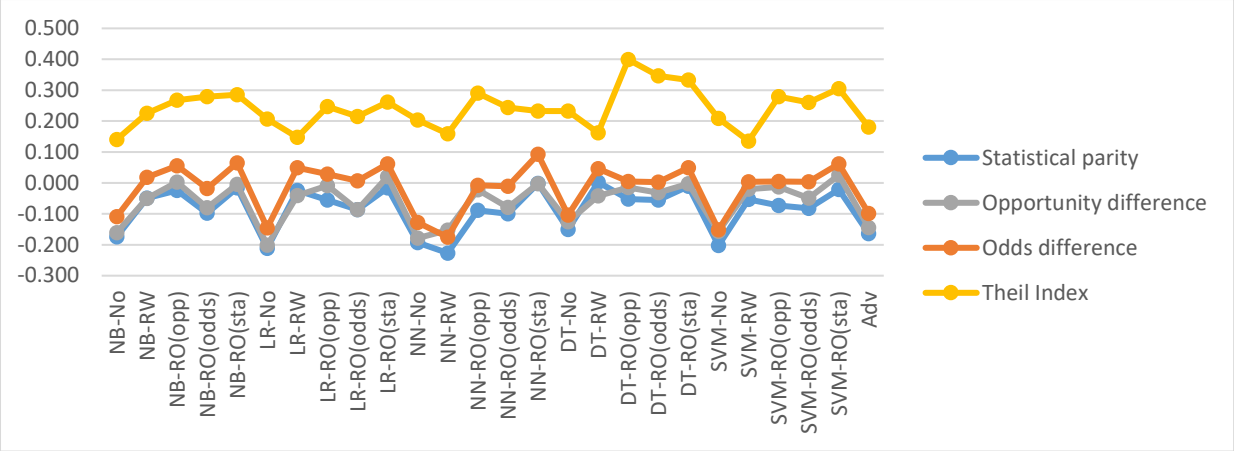


Figure 33) Comparison of fairness measures in the German Credit dataset

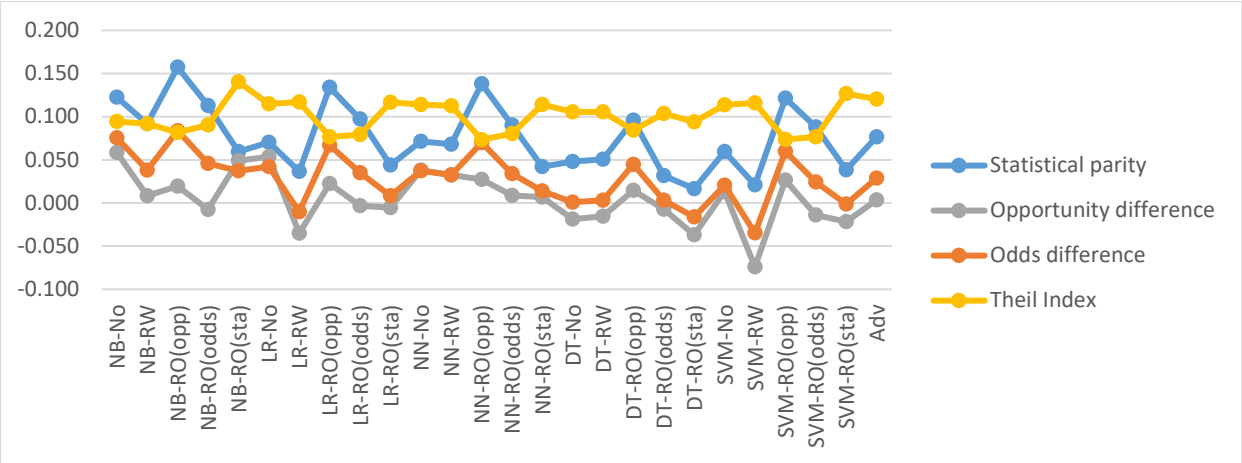


Figure 34) Comparison of fairness measures in the CAMH dataset

With a glance at figures 31 to 34, a similar trend is observable in all fairness measures. In other words, certain combinations of classifiers and fairness approaches can reduce or increase all fairness measures simultaneously. We can conclude that these measures are correlated. Among the three de-biasing methods, applying the Reweighing or Reject Option Classification, has led to an increase of bias in favor of the opposite protected group after de-biasing in some cases, while this issue is not observed with the Adversarial method.

In the Heart Failure and German Credit datasets, the best combinations regarding fairness measures are related to the Logistic Regression and Decision Tree classifiers with the

reject option methods. In the Diabetes dataset, the best one is Logistic Regression with the reweighing technique, while in the CAMH dataset, the Decision Tree classifier and reject option method shows the best fairness measures among the other methods.

The worst fairness measure in the Heart Failure dataset is related to the SVM classifier with reweighing technique that has an opportunity difference of -0.184 and an odds difference of -0.103. Similarly, with the CAMH dataset reweighing method and SVM classifier has led to the least fairness as it shows an opportunity difference and odds difference of -0.074 and -0.035, respectively. In the Diabetes and the German Credit datasets, the lowest fairness rate is observed with the Neural Network and reweighing method. It shows an opportunity difference of 0.165 and statistical parity of 0.125 in the Diabetes dataset and statistical parity and odds difference of -0.227 and -0.176 in the German Credit dataset.

Our experiments on the Reweighing algorithm show that there is not necessarily a tradeoff for accuracy, like what is observed in Faisal Kamiran and Calders (2012). The algorithm generally has not caused a considerable reduction of the accuracy in different datasets. In Kamiran's research, the statistical parity is used as the fairness measure. We showed that this method performs well in reducing the other group fairness measures except with the Neural Network and the SVM algorithms.

Zhang et al. (2018) performed the adversarial de-biasing method on UCI Adult dataset and tried to enforce the "Equality of Odds" on a model for the task of income prediction. Their experiment showed that with a small reduction of the overall accuracy from 86% to 84.5%, the false-negative rate and false-positive rate are approximately equal between the two genders. Applying this method on the four datasets of "Heart Failure", "Diabetes", "German Credit", and "CAMH", illustrated that adversarial de-biasing does not always lead to deterioration of the performance of the classifier. The accuracy of the Heart Failure dataset has increased to 77% from 76%. For Diabetes and Credit datasets, it has risen from 73% to 77% and from 75% to 76%, respectively. However, there is a reduction in the accuracy of the CAMH dataset from 86% to 84%. Although this de-biasing method is purposely made to reduce the Equality of Odds difference, our results show that it can lower other fairness measures as well. It has successfully reduced statistical parity in the Diabetes and Credit

datasets, opportunity difference in the Credit and CAMH datasets, Odds difference in all datasets, and Theil index in Heart, Diabetes, and Credit datasets.

Same as the results in F. Kamiran et al. (2012), applying the Reject Option method on our datasets shows a considerable trade for the accuracy of the classifiers, which means that the lower the discrimination, the lower the accuracy is. In the previous literature, this method was used to reduce the statistical parity difference between the protected groups. Our results indicate that it is useful in reducing the opportunity and odds differences.

6 Conclusions and Future Work

In this study, we applied three different de-biasing techniques on four different datasets. The utilized bias mitigation methods were “Reweighing Algorithm,” a pre-processing method, “Adversarial de-biasing” an in-processing method, and “Reject Object Classification,” a post-processing method. The pre-processing and the post-processing methods were tested on the datasets through five different machine learning schemes. The results were compared regarding their performance and the fairness measures before and after the de-biasing process. The adversarial learning method, an in-processing algorithm based on deep learning schemes, was performed on a one-hidden-layer Multi-Layer Perceptron. The results were compared before and after the bias mitigation process.

In this research, we did a comparative study between different types of fairness enhancing methods. Each of these methods make an effect on a different stage of the classification process. In contrast with the previous literature that concentrated on one specific metric to measure fairness through their experiments, we chose five different metrics that show no incompatibility with each other. This study showed that although each de-biasing method has been introduced to improve a particular measure, they can reduce bias based on several fairness definitions. Our focus was on group fairness approaches, but we used the Theil index method to determine whether these methods have any influences on individual fairness. All the experiments were performed as supervised binary classification tasks.

The adversarial de-biasing method is the best technique for mitigating bias working with the deep learning classifiers. It should be considered that this method is an in-processing method, which directly changes the classification algorithm and does not work with the initial data or the final classification results. As a result, this method can be used just if we are capable of changing the classification process. Using the Adversarial de-biasing method prevents the classifier from increasing the bias in favor of the opposite protected group after de-biasing. Whereas using the Reweighing and Reject Option methods, there is a high possibility that fairness measures decrease in the opposite direction. Using this method had not led to a considerable reduction of accuracy except for the CAMH dataset. Even in some cases with the Diabetes and the Credit dataset, we observed a slight increase in accuracy. The

best performance with the Adversarial De-biasing method was related to the German Credit dataset, where it showed a reduction in all of the fairness measures. With the Heart failure dataset, it reduced the average odds difference. With the Diabetes dataset, there was a reduction in the statistical parity and the average odds. With the CAMH dataset, we observed improvements in the opportunity and average odds measures.

Generally, the "Reweighing Algorithm" does not cause a considerable reduction in the accuracy measure, as the reduction of accuracy after applying this method was not significant during this experiment except for the Diabetes dataset with the Decision Tree and the SVM classifiers. Although this technique can reduce bias in classification tasks, its performance was not as strong as the Reject Option classifier. In the heart failure dataset, the Reweighing method was more successful in reducing the bias with the Decision Tree classifier. In the diabetes dataset, its best performance was related to the Logistic Regression classifier. With the German Credit dataset, this method was the best to reduce the Theil index, although it was not successful with the Naïve Bayes classifier. The Reweighing algorithm did not significantly reduce the bias with the Neural Network and the SVM algorithms as these machine-learning schemas work based on a weight adjustment procedure; setting initial weights would not have a strong influence on the final results with this method.

Among all of the three techniques in this experiment, the "Reject Option Classification" causes the most deterioration of prediction accuracy in all datasets with the default threshold of 50%. This de-biasing method especially did not work well with the Decision Tree classifier as it resulted in a considerable reduction in the accuracy of classification. On the other hand, this method had the best performance in alleviating the bias generated through the classification process. In the heart failure dataset, the Reject Option method showed the lowest bias measures with all of the algorithms except for the Decision Tree classifier. With the diabetes dataset, this method offered the best performance in reducing the bias measures except for the Naïve Bayes and the Logistic Regression classifiers. In the German Credit dataset, the Reject Option classifier showed the most reduction of all the fairness measure except for the Theil index, and its performance in the CAMH dataset is also considerable, though it had not demonstrated the highest reduction with all of the algorithms. (For a recommended fairness enhancing procedure and for a glance

of the methods' properties and their conclusions, you can see the following tables, respectively).

Table 28) Recommended fairness enhancing instruction

Table 29) Summary of the properties and results of methods

Methods	Properties	Results
Reweighing Algorithm	• Pre-processing method • Based on change to the training dataset to reduce dependency between protected groups • New weights are assigned to training samples according to equations 23 and 24 • The weights scale the loss function	• Suitable for fixing unfairness originated from dataset. • Small reduction of the accuracy of classification • Not effective for bias generated by the Neural Networks or SVM classifiers. • Potential of increasing bias in favor of the opposite protected group.
Adversarial De-Biasing	• In-processing method • Based on a deep learning procedure of simultaneous learning of a predictor and an adversary • The objective is to maximize the predictor's ability to predict labels while minimizing the adversary's ability to predict the protected attributes	• The most Suitable method for adapting to different deep learning classifiers • No potential of increasing bias in favor of the opposite protected group • No or small reduction of accuracy
Reject Option Classification	• Post-processing method • Based on change of the test dataset labeling method • Instances within a critical region are labeled differently from the conventional method	• Suitable for situations that we cannot change the training process • Highest rate of fairness in classification • Considerable reduction of accuracy • Potential of increasing bias in favor of the opposite protected group • Most effective with Decision Trees

6.1 Future Work

All the bias mitigating methods that have been introduced in the literature so far are applied to the binary classification problems, and the software toolkits that are designed based on these methods are not capable of performing multi-class and multi-labeled tasks. We also approached the experiments as binary supervised classification tasks. All

unprivileged classes were grouped together into a single value. Although most of the methods can, theoretically, be extended to the multi-class and multi-labeled problems, there are not enough works done in this regard. Future work may consider this deficiency.

In this research, the pre-processing stage was independent of the de-biasing process to provide a consistent comparison across algorithms, and the missing values were removed from the datasets. The experiments show that the fairness enhancing methods have considerably different results with different datasets and classifiers. As a result, there is a need for more studies to calculate the sensitivity of these methods to the various pre-processing and classification methods. Besides, future works can perceive the possibility of different ways of handling the missing values.

The recent literature mostly focuses on the group fairness approaches, and it seems that individual fairness definitions are neglected by researchers. They can perform more studies on the de-biasing methods capable of reducing both the individual and group bias in classification.

Bibliography

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., . . . Devin, M. (2016). Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:.04467*.

Alvarado, O., & Waern, A. (2018). Towards Algorithmic Experience: Initial Efforts for Social Media Contexts. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems* (pp. Paper 286): Association for Computing Machinery.

Amini, A., Soleimany, A. P., Schwarting, W., Bhatia, S. N., & Rus, D. (2019). *Uncovering and mitigating algorithmic bias through learned latent structure.* Paper presented at the AIES '19: AAAI/ACM Conference on AI, Ethics, and Society, Honolulu HI USA.

Angwin, J., Larson, J., Mattu, S., & Kirchner, L. (2016). Machine bias risk assessments in criminal sentencing. *ProPublica, 23*.

Bellamy, R. K., Dey, K., Hind, M., Hoffman, S. C., Houde, S., Kannan, K., . . . Mojsilovic, A. (2018). AI Fairness 360: An extensible toolkit for detecting, understanding, and mitigating unwanted algorithmic bias. *arXiv e-prints*, arXiv:1810.01943.

Ben-Bassat. (1982). Pattern recognition and reduction of dimensionality. *Handbook of Statistics, 2*(1982), 773-910.

Beutel, A., Chen, J., Zhao, Z., & Chi, E. H. (2017). Data Decisions and Theoretical Implications when Adversarially Learning Fair Representations. *arXiv e-prints*, arXiv:1707.00075. https://ui.adsabs.harvard.edu/abs/2017arXiv170700075B

Bosch, N., D'Mello, S. K., Baker, R. S., Ocumpaugh, J., Shute, V., Ventura, M., . . . Zhao, W. (2016). *Detecting student emotions in computer-enabled classrooms.* Paper presented at the IJCAI.

Breiman, L., Friedman, J. H., Olshen, R. A., & Stone, C. (1984). Classification and regression trees. Belmont, CA: Wadsworth. *International Group, 432*, 151-166.

Brown, M. L., & Kros, J. (2003). Data mining and the impact of missing data. *Industrial Management & Data Systems*.

Bucher, T. (2017). The algorithmic imaginary: exploring the ordinary affects of Facebook algorithms. *Information, Communication & Society, 20*(1), 30-44. doi:10.1080/1369118X.2016.1154086

Calders, T., Kamiran, F., & Pechenizkiy, M. (2009, 6-6 Dec. 2009). *Building Classifiers with Independency Constraints.* Paper presented at the 2009 IEEE International Conference on Data Mining Workshops.

Chen, I. Y., Szolovits, P., & Ghassemi, M. (2019). Can AI help reduce disparities in general medical and mental health care? *AMA journal of ethics, 21*(2), 167-179.

Chicco, D., & Jurman, G. (2020). Machine learning can predict survival of patients with heart failure from serum creatinine and ejection fraction alone. *BMC Medical Informatics and Decision Making, 20*(1), 16. doi:10.1186/s12911-020-1023-5

Chouldechova, A. J. B. d. (2017). Fair prediction with disparate impact: A study of bias in recidivism prediction instruments. *5*(2), 153-163.

Chowriappa, P., Dua, S., & Todorov, Y. (2014). Introduction to Machine Learning in Healthcare Informatics. In S. Dua, U. R. Acharya, & P. Dua (Eds.), *Machine Learning in Healthcare Informatics* (pp. 1-23). Berlin, Heidelberg: Springer Berlin Heidelberg.

COVID-19 National Survey Dashboard. Retrieved from https://www.camh.ca/en/health-info/mental-health-and-covid-19/covid-19-national-survey

d'Alessandro, B., O'Neil, C., & LaGatta, T. (2017). Conscientious classification: A data scientist's guide to discrimination-aware classification. *Big data, 5*(2), 120-134.

Diabetes 130-US hospitals for years 1999-2008 Data Set. (2014). *UCI Machine Learning Repository.* Retrieved from https://archive.ics.uci.edu/ml/datasets/diabetes+130-us+hospitals+for+years+1999-2008

Dozat, T. (2015). Incorporating Nesterov momentum into Adam Technical Report. In: Stanford, CA: Stanford University.

Dua, D., & Graff, C. (2019). UCI Machine Learning Repository. https://archive.ics.uci.edu/ml/datasets/statlog+(german+credit+data)

Duchi, J., Hazan, E., & Singer, Y. (2011). Adaptive subgradient methods for online learning and stochastic optimization. *Journal of machine learning research, 12*(7).

Duda, R. O., Hart, P. E., & Stork, D. (2001). Pattern Classification. John Wiley & Sons*, 2*.

Erb, R. J. (1993). Introduction to Backpropagation Neural Network Computation. *Pharmaceutical Research, 10*(2), 165-170. doi:10.1023/A:1018966222807

Esteva, A., Kuprel, B., Novoa, R. A., Ko, J., Swetter, S. M., Blau, H. M., & Thrun, S. (2017). Dermatologist-level classification of skin cancer with deep neural networks. *nature, 542*(7639), 115-118.

Feldman, M., Friedler, S. A., Moeller, J., Scheidegger, C., & Venkatasubramanian, S. (2015). *Certifying and Removing Disparate Impact*. Paper presented at the Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Sydney, NSW, Australia. https://doi.org/10.1145/2783258.2783311

Friedler, S. A., Scheidegger, C., & Venkatasubramanian, S. (2016). On the (im) possibility of fairness. *arXiv preprint arXiv:.07236*.

Géron, A. (2017). *Hands-On Machine Learning with Scikit-Learn and TensorFlow: Concepts. Tools, Techniques to build intelligent systems*.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., . . . Bengio, Y. (2014). *Generative adversarial nets.* Paper presented at the Advances in neural information processing systems.

Hall, M. A. (1999). Correlation-based feature selection for machine learning.

Hardt, M., Price, E., & Srebro, N. (2016a). *Equality of opportunity in supervised learning.* Paper presented at the Advances in neural information processing systems.

Hardt, M., Price, E., & Srebro, N. (2016b). Equality of opportunity in supervised learning. *Advances in neural information processing systems*, 3315-3323.

Hashimoto, T. B., Srivastava, M., Namkoong, H., & Liang, P. (2018). Fairness without demographics in repeated loss minimization. *arXiv e-prints*, arXiv:1806.08010.

Heart failure clinical records Data Set. (2020). *UCI Machine Learning Repository.* Retrieved from https://archive.ics.uci.edu/ml/datasets/Heart+failure+clinical+records#:~:text=UCI%20Machine%20Learning%20Repository%3A%20Heart%20failure%20clinical%20records%20Data%20Set&text=Abstract%3A%20This%20dataset%20contains%20the,profile%20has%2013%20clinical%20features.

HireVue: Video Interview Software & Recruitment Platform. (2020). Retrieved from https://www.hirevue.com/

Holstein, K., McLaren, B. M., & Aleven, V. (2018). *Student Learning Benefits of a Mixed-Reality Teacher Awareness Tool in AI-Enhanced Classrooms*, Cham.

Kamiran, F., & Calders, T. (2009, 17-18 Feb. 2009). *Classifying without discriminating.* Paper presented at the 2009 2nd International Conference on Computer, Control and Communication.

Kamiran, F., & Calders, T. (2012). Data preprocessing techniques for classification without discrimination. *Knowledge and Information Systems, 33*(1), 1-33. doi:10.1007/s10115-011-0463-8

Kamiran, F., Karim, A., & Zhang, X. (2012, 10-13 Dec. 2012). *Decision Theory for Discrimination-Aware Classification.* Paper presented at the 2012 IEEE 12th International Conference on Data Mining.
Kass, G. V. (1980). An exploratory technique for investigating large quantities of categorical data. *Journal of the Royal Statistical Society: Series C, 29*(2), 119-127.
Kingma, D. P., & Ba, J. L. (2014). Adam: A method for stochastic optimization. *arXiv e-prints*, arXiv:1412.6980.
Kleinberg, J., Mullainathan, S., & Raghavan, M. (2016). Inherent trade-offs in the fair determination of risk scores. *arXiv preprint arXiv:.05807*.
Kusner, M. J., Loftus, J., Russell, C., & Silva, R. (2017). Counterfactual Fairness. *Advances in Neural Information Processing Systems 30 (NIPS 2017)*, 4066--4076.
Lambrecht, A., & Tucker, C. (2019). Algorithmic bias? An empirical study of apparent gender-based discrimination in the display of STEM career ads. *Management Science, 65*(7), 2966-2981.
Loh, W.-Y., & Shih, Y.-S. (1997). Split selection methods for classification trees. *Statistica sinica*, 815-840.
Louizos, C., Swersky, K., Li, Y., Welling, M., & Zemel, R. (2016). The variational fair autoencoder. *arXiv e-prints*, arXiv:1511.00830.
Lum, K., & Isaac, W. (2016). To predict and serve? *Significance, 13*(5), 14-19.
Lum, K., & Johndrow, J. (2016). A statistical framework for fair predictive algorithms. *arXiv e-prints*, arXiv:1610.08077.
Machine learning. *Wikipedia.* Retrieved from https://en.wikipedia.org/wiki/Machine_learning#Types_of_learning_algorithms
McCulloch, W. S., & Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics, 5*(4), 115-133.
Mehrabi, N., Morstatter, F., Saxena, N., Lerman, K., & Galstyan, A. (2019). A survey on bias and fairness in machine learning. *arXiv e-prints*.
Mitchell, T. (1997). Machine Learning, McGraw-Hill Higher Education. *New York*.
Narayanan, A. (2018). *Translation tutorial: 21 fairness definitions and their politics.* Paper presented at the Proc. Conf. Fairness Accountability Transp., New York, USA.
Ngo, T. (2011). Data mining: practical machine learning tools and technique, third edition by Ian H. Witten, Eibe Frank, Mark A. Hell. *36*(5 %J SIGSOFT Softw. Eng. Notes), 51–52. doi:10.1145/2020976.2021004
Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., . . . Dubourg, V. (2011). Scikit-learn: Machine learning in Python. *the Journal of machine Learning research, 12*, 2825-2830.
Polyak, B. (1964). Some methods of speeding up the convergence of iteration methods. *Ussr Computational Mathematics and Mathematical Physics, 4*, 1-17. doi:10.1016/0041-5553(64)90137-5
Quinlan, J. R. (1986). Induction of decision trees. In *Machine Learning* (Vol. 1, pp. 81-106).
Rajkomar, A., Hardt, M., Howell, M. D., Corrado, G., & Chin, M. H. (2018). Ensuring fairness in machine learning to advance health equity. *Annals of internal medicine, 169*(12), 866-872.
Reback, J., McKinney, W., jbrockmendel, Bossche, J. V. d., Augspurger, T., Cloud, P., . . . Mehyar, M. (2020). pandas-dev/pandas: Pandas 1.0. 5: Zenodo. Retrieved from https://doi.org/10.5281/zenodo.3715232
Rokach, L., & Maimon, O. (2005). Top-down induction of decision trees classifiers-a survey. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on, 35*(4), 476-487.
Salzberg, S. L. (1994). C4. 5: Programs for machine learning by j. ross quinlan. morgan kaufmann publishers, inc., 1993. In: Springer.

Sammut, C., & Webb, G. I. (2017). *Encyclopedia of machine learning and data mining*: Springer.
Saxena, N. A., Huang, K., DeFilippis, E., Radanovic, G., Parkes, D. C., & Liu, Y. (2019). *How Do Fairness Definitions Fare? Examining Public Attitudes Towards Algorithmic Definitions of Fairness*. Paper presented at the Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society, Honolulu, HI, USA. https://doi.org/10.1145/3306618.3314248
Schmidt, M., Le Roux, N., & Bach, F. (2017). Minimizing finite sums with the stochastic average gradient. *Mathematical Programming, 162*(1-2), 83-112.
Singh, Y., & Chauhan, A. S. (2009). NEURAL NETWORKS IN DATA MINING. *Journal of Theoretical and Applied Information Technology, 5*(1).
Sokolova, M., Lapalme, G. J. I. p., & management. (2009). A systematic analysis of performance measures for classification tasks. *45*(4), 427-437.
Speicher, T., Heidari, H., Grgic-Hlaca, N., Gummadi, K. P., Singla, A., Weller, A., & Zafar, M. B. (2018). *A Unified Approach to Quantifying Algorithmic Unfairness: Measuring Individual &Group Unfairness via Inequality Indices*. Paper presented at the Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, London, United Kingdom. https://doi.org/10.1145/3219819.3220046
Spitzer, R. L., Kroenke, K., Williams, J. B. W., & Löwe, B. (2006). A Brief Measure for Assessing Generalized Anxiety Disorder: The GAD-7. *Archives of Internal Medicine, 166*(10), 1092-1097. doi:10.1001/archinte.166.10.1092 %J Archives of Internal Medicine
Steinwart, I., & Christmann, A. (2008). *Support vector machines*: Springer Science & Business Media.
Strack, B., DeShazo, J. P., Gennings, C., Olmo, J. L., Ventura, S., Cios, K. J., & Clore, J. N. (2014). Impact of HbA1c measurement on hospital readmission rates: analysis of 70,000 clinical database patient records. *BioMed research international, 2014*.
Sutskever, I., Martens, J., Dahl, G., & Hinton, G. (2013). *On the importance of initialization and momentum in deep learning*. Paper presented at the Proceedings of the 30th International Conference on Machine Learning, Proceedings of Machine Learning Research. http://proceedings.mlr.press
Talent Matching Platform. *pymetrics.* Retrieved from https://www.pymetrics.ai/
Tieleman, T., & Hinton, G. (2012). Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural networks for machine learning*

4(2), 26-31.
Verma, S., & Rubin, J. (2018, 29-29 May 2018). *Fairness Definitions Explained.* Paper presented at the 2018 IEEE/ACM International Workshop on Software Fairness (FairWare).
Witten, I. H., Frank, E., Hall, M. A., & Pal, C. J. (2016). *Data Mining: Practical machine learning tools and techniques*: Morgan Kaufmann.
Yu, H.-F., Huang, F.-L., & Lin, C.-J. (2011). Dual coordinate descent methods for logistic regression and maximum entropy models. *Machine Learning, 85*(1-2), 41-75.
Zhang, B. H., Lemoine, B., & Mitchell, M. (2018). *Mitigating unwanted biases with adversarial learning.* Paper presented at the Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society, New York, NY, USA.
Žliobaite, I., Kamiran, F., & Calders, T. (2011, 11-14 Dec. 2011). *Handling Conditional Discrimination.* Paper presented at the 2011 IEEE 11th International Conference on Data Mining.

Appendix A

Table 30) Explanation of features of the Heart Failure Dataset (Chicco & Jurman, 2020)

Feature	Explanation	Measurement	Range
Age	Age of the patient	Years	[40, ..., 95]
Anemia	Decrease of red blood cells or hemoglobin	Boolean	0, 1
High blood pressure	If a patient has hypertension	Boolean	0, 1
Creatinine phosphokinase (CPK)	Level of the CPK enzyme in the blood	mcg/L	[23, ..., 7861]
Diabetes	If the patient has diabetes	Boolean	0, 1
Ejection fraction	Percentage of blood leaving the heart at each contraction	Percentage	[14, ..., 80]
Sex	Woman or man	Binary	0, 1
Platelets	Platelets in the blood	kiloplatelets/mL	[25.01, ..., 850.00]
Serum creatinine	Level of creatinine in the blood	mg/dL	[0.50, ..., 9.40]
Serum sodium	Level of sodium in the blood	mEq/L	[114, ..., 148]
Smoking	If the patient smokes	Boolean	0, 1
Time	Follow-up period	Days	[4,...,285]
(target) death event	If the patient died during the follow-up period	Boolean	0, 1

Note. From Chicco, D., & Jurman, G. (2020). Machine learning can predict survival of patients with heart failure from serum creatinine and ejection fraction alone. BMC Medical Informatics and Decision Making, 20(1), 16. doi:10.1186/s12911-020-1023-5

Table 31) Explanation of features of the Diabetes Dataset (Strack et al., 2014)

Feature name	Type	Description and values	% missing
Encounter ID	Numeric	Unique identifier of an encounter	0%
Patient number	Numeric	Unique identifier of a patient	0%
Race	Nominal	Values: Caucasian, Asian, African American, Hispanic, and other	2%
Gender	Nominal	Values: male, female, and unknown/invalid	0%
Age	Nominal	Grouped in 10-year intervals: [0, 10), [10, 20), .. ., [90, 100)	0%
Weight	Numeric	Weight in pounds.	97%
Admission type	Nominal	Integer identifier corresponding to 9 distinct value	0%
Discharge disposition	Nominal	Integer identifier corresponding to 29 distinct values	0%
Admission source	Nominal	Integer identifier corresponding to 21 distinct values	0%
Time in hospital	Numeric	Integer number of days between admission and discharge	0%
Payer code	Nominal	Integer identifier corresponding to 23 distinct values	52%

Medical specialty	Nominal	Integer identifier of a specialty of the admitting physician surgeon	53%
Number of lab procedures	Numeric	Number of lab tests performed during the encounter	0%
Number of procedures	Numeric	Number of procedures (other than lab tests) performed during the encounter	0%
Number of medications	Numeric	Number of distinct generic names administered during the encounter	0%
Number of outpatient visits	Numeric	Number of outpatient visits of the patient in the year preceding the encounter	0%
Number of emergency visits	Numeric	Number of emergency visits of the patient in the year preceding the encounter	0%
Number of inpatient visits	Numeric	Number of inpatient visits of the patient in the year preceding the encounter	0%
Diagnosis 1	Nominal	The primary diagnosis ; 848 distinct values	0%
Diagnosis 2	Nominal	Secondary diagnosis; 923 distinct values	0%
Diagnosis 3	Nominal	Additional secondary diagnosis; 954 distinct	1%
Number of diagnoses	Numeric	Number of diagnoses entered to the system	0%
Glucose serum test result	Nominal	Indicates the range of the result or if the test was not taken	0%
A1c test result	Nominal	Indicates the range of the result or if the test was not taken	0%
Change of medications	Nominal	Indicates if there was a change in diabetic medications (either dosage or generic name)	0%
Diabetes medications	Nominal	Indicates if there was any diabetic medication prescribed. Values: “yes” and “no”	0%
24 features for medications	Nominal	feature indicates whether the drug was prescribed or there was a change in the dosage	0%
Readmitted	Nominal	Days to inpatient readmission	0%

Note. From track, B., DeShazo, J. P., Gennings, C., Olmo, J. L., Ventura, S., Cios, K. J., & Clore, J. N. (2014). Impact of HbA1c measurement on hospital readmission rates: analysis of 70,000 clinical database patient records. BioMed research international, 2014

Table 32) Features and their description of the CAMH dataset

Feature	Description
wave number	data collection period (from 1 to 4)
region	Province or territory:
age group	18-39
gender	Male
Q4-Have you or those close to you tested positive for COVID-19	I or someone close to me has tested positive for COVID-19
	I or someone close to me has had symptoms of COVID-19 but has not been tested

or are at high risk of COVID-19?	I or someone close to me has been tested for COVID-19 but it was negative
	I or someone close to me is elderly and/or has a health condition that increases the risk of serious illness from COVID-19
	I have a job that exposes me to high risk of getting COVID-19
	Someone close to me has a job that exposes them to high risk of getting COVID-19
	None of the above
Q5-How worried are you about the impact of COVID-19 on your personal financial situation?	Very worried
	Somewhat worried
	Not very worried
	Not at all worried
Q6-How have physical distancing measures due to the COVID-19 pandemic affected your employment situation?	I have continued working but now I am working from home instead of my usual location
	I am not currently working or I have been laid off/let go due to the pandemic
	I was working from home due to the pandemic but now I am back working at my usual location outside the home
	I was previously not working/ laid off/let go due to the pandemic but now I am back at work with the same or a new employer
	No change - I have continued working outside my home as I always did
	No change - I have continued working from home as I always did
	No change - I was not employed prior to the pandemic (e.g. retired student paid leave recently graduated) and I have remained unemployed
	Other
Q6b-On average how has the number of hours you are working for pay been affected by the COVID-19 pandemic?	Increased a lot
	Increased somewhat
	No change
	Decreased somewhat
	Decreased a lot
Q7-How worried are you that you or someone close to you (close relative or friend) will get ill from COVID-19	Very worried
	Somewhat worried
	Not very worried
	Not at all worried
Q8x1-P2W frequency - Feeling nervous anxious or on edge	Not at all
	Several days
	Over half the days
	Nearly every day
Q8x2-P2W frequency - Not being able to stop or control worrying	Not at all
	Several days
	Over half the days
	Nearly every day
	Not at all

Q8x3-P2W frequency - Worrying too much about different things	Several days
	Over half the days
	Nearly every day
Q8x4-P2W frequency - Trouble relaxing	Not at all
	Several days
	Over half the days
	Nearly every day
Q8x5-P2W frequency - Being so restless that it's hard to sit still	Not at all
	Several days
	Over half the days
	Nearly every day
Q8x6-P2W frequency - Becoming easily annoyed or irritable	Not at all
	Several days
	Over half the days
	Nearly every day
Q8x7-P2W frequency - Feeling afraid as if something awful might happen	Not at all
	Several days
	Over half the days
	Nearly every day
Q15-During the PAST 7 DAYS on how many days did you drink ALCOHOL?	Numbers from 0 to 7
Q16-On how many of the PAST 7 DAYS did you drink [4 (if woman) or 5 (if man) or 5 (if other gender)] or more drinks on one occasion?	Numbers from 0 to 7
Q17-In the PAST 7 DAYS did you drink more ALCOHOL about the same or less alcohol overall than you did before the COVID-19 pandemic started?	Drink much more alcohol
	Drink slightly more alcohol
	No change
	Drink slightly less alcohol
	Drink much less alcohol
	Prefer not to answer
Q18-During the PAST 7 DAYS on how many days did you use CANNABIS	Numbers from 0 to 7
Q19-In the PAST 7 DAYS did you use CANNABIS more often about the same or	Much more
	Slightly more
	No change
	Much less

less often overall than you did before the COVID-19 pandemic started?	Slightly less
	Prefer not to answer
Q20x1-In the PAST 7 DAYS how often have you felt depressed?	Rarely or none of the time (less than 1 day)
	Some or a little of the time (1-2 days)
	Occasionally or a moderate amount of the time (3-4 days)
	Most or all of the time (5-7 days)
Q20x2-In the PAST 7 DAYS, how often have you felt lonely?	Rarely or none of the time (less than 1 day)
	Some or a little of the time (1-2 days)
	Occasionally or a moderate amount of the time (3-4 days)
	Most or all of the time (5-7 days)
Q20x3-In the PAST 7 DAYS how often have you felt hopeful about the future?	Rarely or none of the time (less than 1 day)
	Some or a little of the time (1-2 days)
	Occasionally or a moderate amount of the time (3-4 days)
	Most or all of the time (5-7 days)
Household	Live alone
	Live with others
	Prefer not to answer
Children	Kids
	No Kids
What is the highest level of education you have completed?	Did not graduate from high school
	Completed high school
	Some post-high school education (college technical university etc.)
	College diploma / degree
	University diploma / degree
	Prefer not to answer
What is your current marital status?	Married
	Living with a partner
	Widowed
	Divorced
	Separated
	Never married
	Prefer not to answer
Which of the following best describes your racial or ethnic group?	Asian – East (e.g. Chinese Japanese Korean)
	Asian – South (e.g. Indian Pakistani Sri Lankan)
	Asian – South East (e.g. Malaysian Filipino Vietnamese)
	Black (Africa Caribbean North American)
	Indignons (First Nations Inuit Métis)
	Latin American (e.g. Argentinean Chilean Salvadoran)
	Middle Eastern (e.g. Egyptian Iranian Lebanese)
	White (European North American)

	Mixed heritage (e.g. Black – North American & White – North American)
	Other
	Not sure
	Prefer not to answer
What is the total household income?	Less than 40k
	40-79k
	80-119k
	120k+
Do you consider yourself to be living in	Urban area
	Suburban area
	Rural area

Note. Adapted from COVID-19 National Survey Dashboard. Retrieved from https://www.camh.ca/en/health-info/mental-health-and-covid-19/covid-19-national-survey