FACULTY OF SCIENCE AND TECHNOLOGY

MASTER THESIS

Study programme / specialisation:
Computer Science - Data Science

The spring semester, 2022

Open

Author: Bjørn Christian Weinbach

(signature author)

Course coordinator: Tom Ryen

Supervisor(s): Vinay Jayarama Setty

Thesis title: Fairness and Interpretability in Machine Learning Models

Credits (ECTS): 30

Keywords: AI, Fairness, Data Science, Discrimination

Pages: 58

+ appendix: 65

Stavanger, 2022-07-07

Faculty of Science and Technology
Department of Electrical Engineering and Computer Science

Fairness and Interpretability in Machine Learning Models

Master's Thesis in Computer Science
by

Weinbach Bjørn Christian

Internal Supervisors

Vinay Setty

External Supervisors

Rajendra Akerkar

July 7, 2022

"Prediction is very difficult, especially about the future."

Danish Proverb

Abstract

Machine Learning has become more and more prominent in our daily lives as the Information Age and Fourth industrial revolution progresses. Many of these machine learning systems are evaluated in terms of how accurately they are able to predict the correct outcome that are present in existing historical datasets. In the last years we have observed how evaluating machine learning systems in this way has allowed decision making systems to treat certain groups unfairly. Some authors have proposed methods to overcome this. These methods include new metrics which incorporate measures of unfairly treating individuals based on group affiliation, probabilistic graphical models that assume dataset labels are inherently unfair and use dataset to infer the true fair labels as well as tree based methods that introduce new splitting criterions for fairness. We have evaluated these methods on datasets used in fairness research and evaluated if the results claimed by the authors are reproducible. Additionally, we have implemented new interpretability methods on top of the proposed methods to more explicitly explain their behaviour. We have found that some of the models do not achieve their claimed results and do not learn behaviour to achieve fairness while other models do achieve better predictions in terms of fairness by affirmative actions. This thesis show that machine learning interpretability and new machine learning models and approaches are necessary to achieve more fair decision making systems.

Acknowledgements

I would like to thank my supervisor for his fantastic enthusiasm and help with writing this thesis. He has been available on a weekly basis and has shown a keen interest in following up this thesis. The thesis would not have taken the direction that it has without his contributions.

Huge thanks to the Western Norway Research Institute and its Teknoløft Sogn og Fjordane initiative for their support throughout the spring of 2022. They have been very helpful in allowing me to focus on the thesis while also employing me during this spring. In the forefront of the institute i would like to especially mention Rajendra Akerkar for his patience and support. I am very happy and proud to have the Western Norway Research Institute as my employer.

And lastly, I would like to thank family and friends for their support through a spring which has been demanding for me in regards to different private matters. Of utmost importance is my partner, Brita Lie Lysne, without her support I would most certainly not have been able to complete my exchange studies in Sweden. She has through her love, support and patience helped ensure that this thesis would be completed in due time.

To all of you, you have my deepest gratitude.

Abbreviations

Bayesnet	**Bayesian Network Module in Forseti**
COMPAS	**C**orrectional **O**ffender **M**anagement **P**rofiling for **A**lternative **S**anctions
FairBN	**Fair B**ayesian **N**etwork
FRFC	**F**air **R**andom **F**orest **C**lassifier
IncomeBN	**Income B**ayesian **N**etwork
NB	**N**aïve **B**ayes
NBSens	**N**aïve **B**ayes **Sens**itive
NSGA-II	**N**ondominated **S**orting **G**enetic **A**lgorithm
pgmpy	**P**robabilistic **G**raphical **M**odels **Py**thon
RQ	**R**esearch **Q**uestion
SCAFF	**S**plitting-**C**riterion **f**or **Fairness**

Contents

Chapter 1

Introduction

1.1 Background and Motivation

Presently, we are undergoing both the Information Age and the fourth industrial revolution [1, 2]. The information age has been characterised by the commercialisation of computer power resulting from technological advances in transistor technology and global communication technologies [1, p. 30]. The fourth industrial revolution is marked by growing connectivity and intelligent automation [3]. Modern smart technology, large-scale machine-to-machine communication (M2M), and the internet of things (IoT) are causing fundamental shifts in the global production and supply network as old manufacturing and industrial methods continue to be automated.

In this broader technological advancement, machine learning has become only one of several fields. In particular, machine learning is replacing manual labour through automation and robotics, as well as higher-level decision-making by quantifying large-scale data and applying this information to provide insight to human decision-makers [2].

The way we do business is changing rapidly. Machine learning is becoming more embedded into our lives, working behind the scenes in diverse scenarios, from optimising production yield to recommending products and more. This shift has enhanced awareness about the implications of using machine learning in numerous processes, as well create more demand to make machine learning powered decisions more fair and interpretable.

These transformations are required to meet many of the UN-defined sustainability goals, such as affordable and clean energy, decent work and economic growth, and industry, innovation, and infrastructure (goals 7, 8, and 9, respectively) [3]. In addition to these goals, the United Nations has outlined two others: gender equality and reduced disparities,

goals 5 and 10, respectively [4]. In machine learning research, equality and fairness are consequently receiving a great deal more focus than in the past [5].

Discrimination is when people are treated unfairly because of the groups, classes, or other categories to which they belong or are seen to belong. In terms of machine learning, there are many distinct definitions of discrimination [6] and the fairness that one wants to attain to avoid this prejudice [7].

According to Dressel and Farid [8], a real world example of discrimination in terms of machine learning is COMPAS (Correctional Offender Management Profiling for Alternative Sanctions). A frequently used technique for determining criminal risk. Since its inception in 1998, it has been used to examine over 1 million offenders. COMPAS uses 137 factors about a person and their criminal history to forecast whether they would commit a misdemeanour or felony within two years of being assessed.

One would think that ignoring sensitive groups, classes, or other categories would be an easy method to avoid discrimination. Counterintuitively, omitting sensitive attributes is insufficient to eliminate prejudice. The discriminatory decision rule is learned indirectly by the machine learning model from qualities that correlate to the sensitive one [8, 9]. This process is called redlining. The sensitive attribute must be included to penalise the machine learning model when it discriminates.

The source of bias and discrimination often comes from the data that the machine learning algorithm is trained on. According to Mehrabi et al. [5] some prominent sources of bias are

- **Historical Bias**: Historical bias is the already existing bias and sociotechnical issues in the world. This affect the data generation process. Imagine trying to make a decision-making system for accepting people to a certain education institution. This system looks at data on previous admissions. From this data, it seems that men dominates studies like engineering. While this data is correct and reflects the current reality, the question remains on whether the system should reflect this reality in its decision-making.
- **Representation Bias**: Lack of representation of certain groups in datasets skew the dataset from the real-world distribution. This bias arises when the sample does not reflect the subgroups in the population that we make inference on. A well known example of this is in image classification, where men, white people and people from the western world have dominated image datasets. This means that other ethnicities, especially black women, suffer from discrimination from systems using image data for training [10].

- **Simpsons Paradox**: Bias originating from the analysis of heterogeneous data that is composed of subgroups or individuals with different behaviours. The best known example of this bias is from the University of California, Berkeley. Examination of aggregate data on graduate admissions to the University of California, Berkeley, for fall 1973 shows a clear but misleading pattern of bias against female applicants. The problem was that the analysis did not take into account that women tended to apply for departments that were very competitive and men applied for departments that were less competitive. When disaggregating the data this relationship disappears and a small favour towards women was shown [11].

Data, especially big data, is often heterogeneous, generated by subgroups with their own characteristics and behaviours. The heterogeneity can bias the data. A model learned on biased data may lead to unfair and inaccurate predictions.

There are currently several challenges in the field of fair machine learning to face. Since ignoring the sensitive attributes does not help to mitigate bias, one must discover how to best employ these sensitive attributes to achieve fairness. In the literature, there exists several statistical and mathematical definitions of fairness. Rather concerning is the fact that some of these definitions are mathematically impossible to satisfy simultaneously [12].

As of the time of writing this thesis, there is no gold standard on how to train fairness-aware machine learning algorithms, and there exists several approaches on how to achieve this [5]. The goal of this thesis is to explore some of these approaches and see how they compared to others.

1.2 Objectives

The goal of this thesis is to explore certain models and approaches to achieve fairness-aware machine learning systems. Specifically, the thesis has the following objectives

- Discover how probabilistic machine learning and graphical models can be used to model the discrimination process.
- Discover how probabilistic machine learning and graphical models can be used to quantify uncertainty in the model and its fairness.
- Explore what definitions of fairness are most appropriate for probabilistic machine learning.

- How does the probabilistic approach compare to traditional machine learning methods?
- Are the fair machine learning models explainable?

1.2.1 RQ1: What probabilistic graphical model is most appropriate to model the discrimination process?

Probabilistic machine learning models come in many different forms. In this thesis we want to use probabilistic machine learning to model the discrimination and data generation process. This is explained in further detail in Section 2. At the end, a specific probabilistic graphical model should be provided.

1.2.2 RQ2: Are the proposed models explainable?

Training a machine learning model to be able to learn decision rules that minimise some mathematical definition of fairness is one thing, but how does the model use the sensitive attributes in its decisions? The proposed fairness-aware models should be evaluated in terms of fairness and explainability as well.

1.2.3 RQ3: Are probabilistic machine learning models cost-effective?

The probabilistic model proposed in this thesis should be compared to baseline methods to see if there are any benefits in terms of performance, accuracy, fairness as well as adding the Bayesian perspective to the problem. There should be a statistically significant increase in accuracy and fairness to justify the additional complexity in computation to be considered cost-effective. At the end, simulations should be presented with both real world datasets and synthetic datasets and their respective performance metrics.

1.3 Approach and Contributions

In this thesis, different algorithms and methods have been explored. These have all been made into a python package for ease of use. This python package has been named Forseti, named after the Norse god of justice and reconciliation. This python package has several modules with implemented algorithms. The code and relevant documentation is available in the following github repository [1], as well as attached to this thesis when submitted.

[1] https://github.com/bcwein/Forseti

Later, the machine learning models are investigated whether they are interpretable. Feature Importance is calculated and Individual Conditional Expectation Plots are performed on the models to explain how the models are using the sensitive attributes in their decision making. This way we hope to gain better insight into how the fair models change their decision compared to their traditional counterparts.

We find that while models satisfy some definition of fairness, when one looks into the models decision making, one quickly realise if this increase in fairness scores is due to reducing model accuracy and predicting randomly or if the model actively uses the data in its decision making and is trying to learn fairer predictions.

1.4 Outline

In this section we have discussed the broader picture of machine learning and the current technological and economic developments in the world. Especially how fairness is becoming evermore important in society as a whole. Through this we have defined some research questions (RQs) that we want to explore.

In chapter 2 we go into more detail on the related works and previous methods already developed. We go into detail on how they work and what described the mechanisms behind them. Chapter 2 serves to give you the necessary background knowledge to understand the workings in later chapters.

Chapter 3 describes the approach that is used in this thesis. We describe the software developed and how it is organised, how the experiments have been set up and how you can set this up on your own system. This chapter gives you the necessary insight to follow the approach yourself and understand exactly how the work in this thesis has been done.

Next in chapter 4 we show the results from the approach described in chapter 3. This includes data exploration, experimental results, presentation of hypothesis tests and discussion of these results.

Finally in chapter 5 we draw the final remarks. Including what we have found out, what has been good about the approach and our results and what are the limitations of this study. And finally, we conclude and propose possible future work.

Chapter 2

Background and Related Work

One of the most cited and comprehensive articles out there for getting to know the field of fairness in machine learning is the survey paper by Mehrabi et al. [5]. This paper elaborates numerous concerns about the fairness of the models' outputs. It serves as a gateway to a lot of the current research in the field, a lot of which will be summarised in this section of the thesis.

2.1 Algorithmic Fairness

Many definitions of discrimination exist, and while there is no gold-standard, it is often defined as an absence of any prejudice or favouritism towards individuals or groups based on some intrinsic traits [5, 13]. These definitions are core in the many mathematical definitions of fairness. Some of these are summarised below

Equalized Odds: According to Mehrabi et al. [5], Hardt et al. [14] A predictor $\hat{Y}$ satisfies equalised odds with respect to a sensitive attribute A and outcome Y if $\hat{Y}$ and A are conditionally independent on Y. i.e.

$$P(\hat{Y}|A = 0, Y = y) = P(\hat{Y}|A = 1, Y = y), y \in 0, 1$$

Equal Opportunity: According to Mehrabi et al. [5], Hardt et al. [14] A binary predictor $\hat{Y}$ satisfies equal opportunity with respect to A and Y if

$$P(\hat{Y} = 1|A = 0, Y = 1) = P(\hat{Y} = 1|A = 1, Y = 1)$$

Demographic Parity: According to Mehrabi et al. [5], Dwork et al. [15] A predictor $\hat{Y}$ satisfies demographic parity if

$$P(\hat{Y}|A = 0) = P(\hat{Y}|A = 1) \tag{2.1}$$

And many more metrics exist. A challenge is that according to Mehrabi et al. [5], Kleinberg et al. [12] it is impossible to satisfy some of these fairness constraints. One should therefore be considerate when using a certain metric. Synthesising these definitions to one gold-standard remains an open RQ. For this thesis. Demographic parity is especially important as it is fundamental to the model described in Section 2.6.

Strong Demographic Parity: According to Barata and Veenman [16], this is a parity score that extends demographic parity by considering fairness throughout the entire range of possible decision thresholds. It was proposed by Jiang et al. [17]. When learning a fair classifier to satisfy strong demographic parity, the predictor $\hat{Y}$ must satisfy demographic parity for any threshold t and sensitive attribute A

$$\forall t \in \hat{Y} : P(\hat{Y} \geq t|A = 0) = P(\hat{\geq} t|A = 1)$$

This assumes that the model output $\hat{Y}$ is a probability or score of belonging to the class of interest and t is a selected threshold for classifying.

2.2 Methods for Fair Machine Learning

Machine Learning is a large domain, encompassing many subdomains. These include *Classification, Regression, PCA, Clustering, Deep Learning* and many more. In this thesis, the focus will be on fair classification.

Fair Classification: Some of the most important works in the field of fair classification, summarised by [5]. Some important previous work for this thesis is the naïve Bayes approach for fair classification by Calders and Verwer [9] In this work the authors investigated how to modify the naïve Bayes classifier in order to perform classification that is independent of the sensitive attributes. In this paper they measure discrimination by *discrimination score* which is defined as

$$P(C = +|S_+) - P(C = +|S_-)$$

Which assumes that a classifier is fair if the outcome is independent of the sensitive attribute. i.e., *Demographic Parity* as described above. The main limitation of this paper is that the classifier on the non-sensitive attributes is a naïve Bayes classifier. This means it assumes that all the features are independent. This makes it one of the simplest Bayesian networks out there as well as scalable since the number of parameters scales linearly with the number of features. We address this limitation further in Section 2.3.

Other important works are the works of [18] and Dwork et al. [19]. Zafar et al. [18] introduced new notions on how to define fairness, arguing that the traditional parity based notion is quite stringent, limiting the overall decision-making accuracy. They tie in elements from envy-freeness literature in economics and game theory and propose preference-based notions of fairness.

Dwork et al. [19] provide a simple and efficient decoupling technique, which can be added on top of any black-box machine learning algorithm, to learn different classifiers for different groups. Using transfer learning to mitigate the problem of having too little data on any one group.

Important to this thesis is the work of Choi et al. [20] Which is a follow-up paper to [9]. They have generalised the limitation of the first paper, where a naïve Bayes classifier was necessary. Their framework can be generalised to any local probabilistic network. This will be described in more detail in section 3.5. The work of Barata and Veenman [16] and their development of a fair tree based classifier using strong demographic parity is also important for this thesis and is described in more detail in section 2.7.

2.3 Probabilistic Machine Learning

In this thesis, we will focus on probabilistic machine learning, therefore a brief introduction to this field is in place. According to Murphy [21], machine learning is usually divided into two main types. In **Predictive** or **Supervised learning** approach, the goal is to learn a mapping from inputs x to outputs y given a labelled set of input-output pairs [21, p. 2]

$$\mathcal{D} = (x_i, y_i)_{i=1}^{N}$$

The second type is the **descriptive** or **unsupervised learning** where we are only given the data itself without labels

$$\mathcal{D} = x_{i i=1}^{N}$$

here the goal is to find interesting patterns in the data that are inherent to the data itself without the need for labels. This problem is not as well-defined as the predictive case, and there is no obvious error metric. The third type is **reinforcement learning**, where you let an agent explore a space and reward desired behaviour through a performance or reward metric.

A common way to perform supervised learning is to treat y as a random variable and estimate a mapping

$$f : x \to y$$

One example of this is Linear Regression. Which maps input vectors x to outputs y using the following mapping [21, p. 19]

$$f : y = \mathbf{w}^T x + \epsilon = \sum_{j=1}^{N} w_j x_j + \epsilon$$

and often ϵ is assumed to be Gaussian and the model can be rewritten as

$$p(y|x, \theta) = \mathcal{N}(y|\mu(x), \sigma^2(x))$$

One common way to estimate the parameters of a statistical model is to calculate the maximum likelihood estimate of the model parameters [21, p. 217]

$$\hat{\theta} = \arg\max_{\theta} \log p(\mathcal{D}|\theta)$$

and for linear regression, minimising the sum of squared errors has an explicit solution [21, p. 220]

$$\hat{\mathbf{w}}_{\text{OLS}} = (\mathbf{X^T X})^{-1} \mathbf{X^T y}$$

This estimate gives us a point estimate of the model parameters. This is traditionally what many machine learning algorithms do, take a dataset and calculate the most likely point estimate of the model parameters. It is reasonable to assume that the model

parameters that are returned from one dataset are different to the true model parameters, and it would in many cases be beneficial to know how uncertain the model parameters are. This is where the **probabilistic** approach comes in.

In a probabilistic approach, we treat the input data and labels as random variables, but also the model parameters. After training, we will have a distribution of model parameters which we can sample from and simulate different realisations of our models. One example of this is **Bayesian Linear Regression**

In Bayesian linear regression, the likelihood of y is given by [21, p. 232]

$$p(\mathbf{y}|\mathbf{X}, \mathbf{w}, \mu, \sigma^2) = \mathcal{N}(\mathbf{y}|\mu + \mathbf{X}\mathbf{w}, \sigma^2\mathbf{I_N})$$

and using a Gaussian prior distribution since it is a conjugate prior, the posterior becomes [21, p. 232]

$$p(\mathbf{w}|\mathbf{X}, \mathbf{y}, \sigma^2) \propto \mathcal{N}(\mathbf{w}|\mathbf{w_0}, \mathbf{V_0})\mathcal{N}(\mathbf{y}|\mathbf{X}\mathbf{w}, \sigma^2\mathbf{I})$$

And we have a full distribution of model parameters, which gives us insight into the uncertainty of the model. This property is desired for assessing fairness and uncertainty later in this thesis.

2.4 Graphical Models

A graphical model is a way to represent a joint probability distribution. Nodes represent random variables and the edges between the random variables represent dependencies, and the lack of edges means the random variables are conditionally independent [21, p. 308]. There are many graphical models, and all of them tie probability theory and graph theory together comprehensively. We describe some different models in this section.

2.5 Bayesian Networks

Graphical models give us a graphical way to represent the joint PDF. It models the conditional dependencies between random variables. From this graph, we see that the joint probability distribution of this classifier is

$$p(y, x_1, x_2, x_3, x_4) = p(y)p(x_1|y)p(x_2|y)p(x_3|y)p(x_4|y)$$

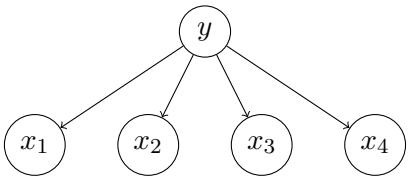


Figure 2.1: Bayesian network with 4 features representing the naïve Bayes classifier

2.5.1 Naïve Bayes Classifier: Baseline Method

The naïve Bayes classifier is the simplest Bayesian network, assuming that all features are independent conditionally on the class variable, hence the name naïve Bayes. This naïve assumption gives the model few parameters to learn, and it requires little training data to achieve good performance. From the work by Zhang [22] we know that naïve Bayes classifiers perform well despite the assumption of independence among features due to the dependencies cancelling each other out and dependencies distributing evenly among classes.

According to Ankan and Panda [23, p. 217], assume that we have a dataset $X = (x_1, x_2, \ldots, x_N)$ with N independent features and k classes C_k that we want to classify the data to. naïve Bayes does this by modelling the posterior distribution in terms of the joint probability

$$P(C_k|X) \propto P(C_k, X) = P(C_k) \prod_{i=1}^{n} P(x_i|C_k)$$

Where the naïve assumption is that $P(x_i|X \setminus x_i, C_k) = P(x_i|C_k)$ i.e., the features are mutually independent conditioned on C_k. Different naïve Bayes methods exist for the assumptions on the priors and likelihoods, i.e. if they are Gaussian, binomial, categorical etc. The model classifies the data to the class that has the highest posterior probability.

2.5.2 pgmpy: naïve Bayes

We have used the naïve Bayes classifier implemented in pgmpy and is used as a baseline method in the experiment described in section **??**. The implementation in pgmpy implements a naïve Bayes method and assumes categorical distributions on all parameters. We calculated the probabilities as conditional probability tables (CPD Tables) using the MLE estimate from data. I.e. calculating probabilities from the dataset by counting occurrences conditioned on the class label. For more detail on how this is implemented, see pgmpy's documentation [1].

[1] https://pgmpy.org/index.html

2.5.3 Structure Learning: Hill Climb Search

According to Koller and Friedman [24, p. 811] finding a maximum-score graphical network evaluated under any decomposable scoring function is NP-hard. Thus, we have to resort to a heuristic algorithm that attempt to find the best network, but are not guaranteed to do so. We have used Hill-Climb search which try to find the best graph G_{best} by selecting an initial network $G_{\emptyset}$, which in pgmpy is a network with no edges. Then we search through all possible search operations O to the network (delete, add, reverse in pgmpy) and score them. We perform the change that gives the best score until convergence or the maximum number of iterations is reached. The algorithm is shown below:

Algorithm 2.1 Hill Climb Searched with Data Perturbation

Input: $G_{\emptyset}$ = Initial Network, D = fully observed dataset, score = scoring function, O = search operations, search = search procedure, t_0 = initial perturbation size, γ = Reduction in perturbation size.
Output: G_{best} = Best network structure found.

```
G ← Search(G_∅, D, Score, O)
G_best ← G
t ← t_0
for i ∈ {1, ..., until convergence} do
    D' ← Perturb(D, t)
    G ← Search(G, D', Score, O)
    if Score(G : D) > Score(G_best : D) then
        G_best ← G
    end if
    t ← γ · t
end for
```

For more information about the algorithm and Hill Climb Search, see [24, p. 816–819] and the implementation in pgmpy which also adds some parameters like red-listed edges and non-changable edges.[2]

2.5.4 Parameter Learning: Expectation Maximization

After we have learned the model structure, we will have to learn the parameters of the model given its structure. Since we will use a model with latent variables, Expectation Maximisation is the algorithm of choice. The expectation maximisation algorithm in general as described by Murphy [21], Bishop and Nasrabadi [25] is as follows:

Consider a probabilistic model in which we collectively denote all the observed variables by $\boldsymbol{X}$ and the latent variables $\boldsymbol{Z}$. The joint distribution $p(\boldsymbol{X}, \boldsymbol{Z}|\boldsymbol{\theta})$ is governed by the parameters $\boldsymbol{\theta}$. We want to maximise the likelihood given by

[2] https:/Z/pgmpy.org/_modules/pgmpy/estimators/HillClimbSearch.html#HillClimbSearch.estimate

$$p(\boldsymbol{X}|\boldsymbol{\theta}) = \sum_{\boldsymbol{X}} p(\boldsymbol{X}, \boldsymbol{Z}|\boldsymbol{\theta})$$

Which is not computable, as $\boldsymbol{Z}$ is unknown. Therefore the expectation step is introduced which is to compute Q

$$Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{\text{old}}) = E[\log L(\boldsymbol{\theta}; \boldsymbol{X}, \boldsymbol{Z})]$$

In the M step we optimise Q w.r.t. $\boldsymbol{\theta}$

$$\boldsymbol{\theta}^{\text{new}} = \arg\max_{\boldsymbol{\theta}} Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{\text{old}})$$

this is done iteratively until a certain threshold is achieved. For more details on how exactly this is implemented in pgmpy, see the documentation.[3]

2.6 Modelling Discrimination Process

Now that probabilistic machine learning and graphical models have been introduced, it is time to introduce the work of Choi et al. [20] in more detail. As discussed previously in this section. There are many sources of bias in data and it is reasonable to assume that almost all datasets out there is biased Choi et al. [20]. describes a way of learning fair probability distributions from biased data by explicitly modelling a latent variable that represents a hidden, unbiased label. In particular, they aim to achieve demographic parity by enforcing certain independencies in the learned model.

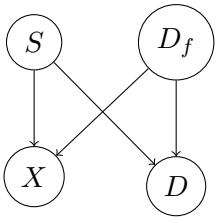


Figure 2.2: Bayesian network structures that represent the proposed fair latent variable approach from [20]

In other words, they model the process on how biased datasets are generated. The biased labels present in the dataset are dependent on the sensitive attributes S and the true fair labels D_f. The latent variable D_f is used for decision making on future instances by inferring $P(D_f|e)$ given some evidence.

[3] https://pgmpy.org/_modules/pgmpy/estimators/EM.html#ExpectationMaximization

The paper states that any probabilistic model can be used but that this model needs to satisfy the independence assumptions in the Bayesian network.

2.7 Fair Tree Classifier

The paper by Barata and Veenman [16] has been implemented in this thesis and the related python module. They introduce a new splitting criterion that evaluates splits in terms of the Area under curve (AUC) w.r.t. the predicted value and the sensitive attribute. Assume that we want to learn a classifier f that learns a mapping from features X and predictor $\hat{Y}$ which outputs a probability of belonging to the predicted class

$$f : X \to \hat{Y}$$

The AUC for this predictor w.r.t. the true labels Y can be calculated as

$$AUC_Y(\hat{Y}, Y) = \frac{\sum_{t_0 \in Y_-} \sum_{t_1 \in Y_+} \mathbf{1}[\hat{Y}_{t_0} < \hat{Y}_{t_1}]}{|Y_-| \cdot |Y_+|}$$

where Y_- and Y_+ are the set of indexes for negative and positive instances in the true labels. $\mathbf{1}$ denotes the indicator function. The authors calculate the AUC score for the predicted labels using scikit-learn [26] and the method called roc_auc_score [27]. When calculating the AUC w.r.t. the sensitive attribute, denoted AUC_s the authors have derived the following formula

$$AUC(\hat{Y}, S) = \max(1 - AUC(\hat{Y}, S), AUC(\hat{Y}, S)) \tag{2.2}$$

The max operator maps the bounds to the range $[0.5, 1]$. The authors then introduce the splitting criterion used in their tree algorithm, Splitting Criterion AUC for Fairness (SCAFF). Which is calculated as

$$SCAFF(\hat{Y}, Y, S, \Theta) = (1 - \Theta) \cdot AUC_Y(\hat{Y}, Y) - \Theta \cdot AUC_S(\hat{Y}, S)$$

Θ is here a hyperparameter of the tree classifier. When $\Theta = 1$ splits are only evaluated in terms of fairness and vice versa. As is typical with tree learning, the architecture is learned by evaluating splits at each depth and selecting the split that maximises the splitting criterion. Other hyperparameters as maximum depth, number of bins etc are also used.

2.8 Interpretable Machine Learning

Up until this point in the thesis, we have discussed fair machine learning models and sources of bias in the data. Another important field in machine learning that is important regarding fairness is Interpretable Machine Learning. Many methods for fairness rely on either pre-processing of the datasets to make the datasets more fair or in-processing methods that change the machine learning algorithm to reduce discrimination during training [5].

Interpretable Machine Learning methods instead focus on understanding the mechanisms behind the decision that the machine learning model makes. This way, we can investigate how the machine learning model uses the data to make a prediction. According to Miller [28] interpretability is how well a human could understand the decisions in the given context. The notion of interpretability is domain-specific and depends on the purpose of the interpretable component in the first place. The higher the interpretability of a machine learning model, the easier it is for someone to comprehend why certain decisions or predictions have been made [29].

2.8.1 Example of interpretability

Examples of interpretable machine learning models are Linear Regression, Logistic Regression and Decision Trees among many others. Linear Regression is especially interpretable, as we shall describe in this section. According to Molnar [29], linear models can be used to model the dependence of a regression target $\boldsymbol{y}$ on some features $\boldsymbol{x}$. The learned relationships are linear and can be written for a single instance i as follows

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\beta} + \epsilon$$

Where β_1 is the weight associated with feature X_1. These weights can be interpreted in several ways depending on the nature of the feature.

- Numerical Features: Increasing the feature by a unit of one increases the estimated outcome by β_i.
- Binary Feature: Changing the binary feature form the reference category to other category increases the outcome by β_i
- Categorical Feature: Create dummy variables. Interpretation of each dummy variable is the same as for binary features.

As you see, linear regression models are structured in such a way that it is easy to explicitly describe how the predictions of the models are made. Other models are also interpretable, which is described in more detail by Molnar [29].

Chapter 3

Approach

3.1 Overall Approach

To answer the three RQs mentioned in section 1.2 we want to compare some models find in literature. Mainly the fair Bayesian network and the fair tree classifiers. These should be evaluated on datasets used in machine learning research. We then will perform experiments evaluating the performance metrics of the models, as well as new performance metrics reflecting fairness. After collecting the data on the performance of the models, hypothesis testing will be done to evaluate whether the differences are significant.

3.2 Python Code Repository: Forseti

We manage the code through GitHub, a python module named Forseti as well as some Jupyter notebooks. In figure 3.1 the structure of the module is shown. It has the following modules:

- Bayesnet: This module contains classes of Bayesian networks.
- Datasets: This module contains functions for generating synthetic datasets.
- Datproc: Data processing module.
- Fairness: Fairness metrics and fairness reports.
- Tree: Tree based methods and classifiers.

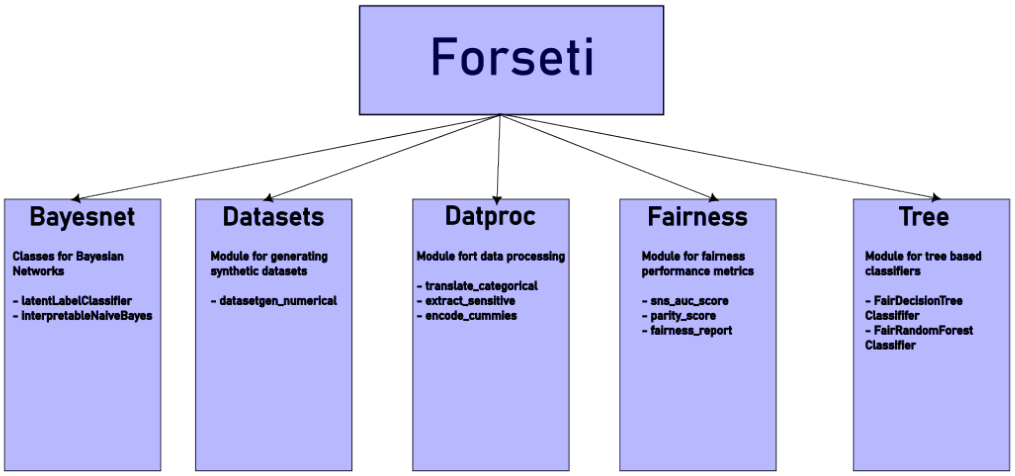


Figure 3.1: Setup of Forseti Python Module

The code and its documentation is available at GitHub[1]. The aim of organising the code in such a way was to make it as easy as possible to let others run the code on their own systems. How to do this is described in the next section.

3.2.1 Setup

The environment used in Forseti is available in the file *environment.yml* and one can setup the environment using Anaconda. See the documentation [2] on how to do this.

If one prefers to not use anaconda, the necessary packages are as follows:

- Python
- Pytest
- Black
- Jupyter

[1] https://github.com/bcwein/forseti
[2] https://docs.conda.io/projects/conda/en/latest/user-guide/tasks/manage-environments.html#creating-an-environment-from-an-environment-yml-file

- ipykernel
- Pandas
- Seaborn
- pip
- pgmpy

3.3 Data Exploration and Selection

For this thesis, exploration of approaches to achieve fairness and interpretability in machine learning algorithms is the goal. The Adult dataset [3] was selected for this topic. Before beginning on implementing machine learning methods and experiments, some preliminary data exploration is in order.

3.3.1 Adult Dataset

To select the features of interest and as an initial data exploration. We will investigate the correlation between the features and the dependent variable *income*. Since almost all the attributes are categorical, we calculate correlation using dummy variables. Which means, transforming categorical attributes to columns of binary attributes. We then explore which dummy variables have the highest absolute correlations. We interpret the weight of each variable as the absolute sum of its dummies.

3.3.2 COMPAS Dataset

The COMPAS dataset contains records for defendants from Broward County indicating their jail and prison times, demographics, criminal histories and COMPAS risk scores from 2013 to 2014 [5]. This dataset is high dimensional with a mix of categorical, numerical and date time columns. For this thesis, the focus has not been to implement the best model out there, but rather compare the fairness between models. Therefore, we have limited ourselves to the following attributes when training our algorithms

- Sex: Gender of individual (binary)
- Age: Age of individual (positive integer)

[3] https://archive.ics.uci.edu/ml/datasets/adult

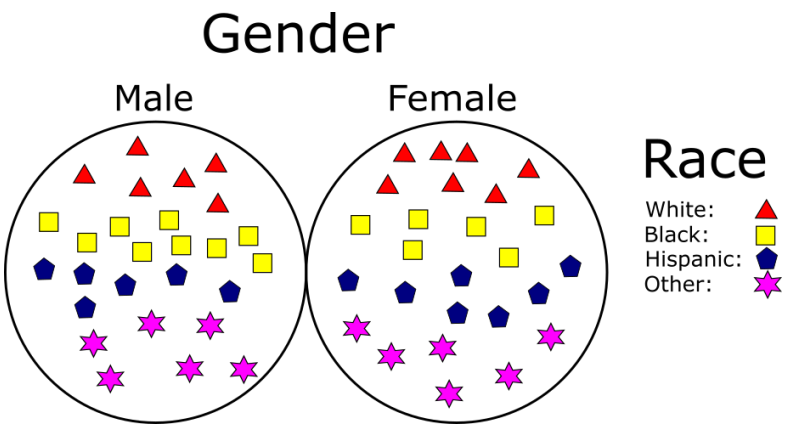


Figure 3.2: Visualisation of overlapping groups.

- Race: Ethnicity of individual (6 categories)
- Priors Count: Prior Crimes (positive integer)
- Juvenile felony count: Felonies as a juvenile (positive integer)
- Juvenile misdemeanour count: Misdemeanours as a juvenile (positive integer)
- Juvenile other count: Other charges as a juvenile (positive integer)
- Charge degree: Degree of current charge (binary)
- Two year recidivism: Whether individual reoffended within two years (binary)

We selected these attributes mainly due to these being the only attributes in the dataset that are not recorded after rearrest and were not in a date time format. The point is not to make the best performing classifier, but to compare classifiers with regard to fairness.

3.4 Metric for fair machine learning

Using demographic parity as described in equation 2.1 is the metric of choice for evaluating the model fairness in this thesis. The main reason for this is that the metric does not assume that the dataset labels are fair. Demographic parity is appropriate to use when we want our predictions to be more in line with a state of nature that we want to see in the world and when we are aware of historical biases that affect the data. [4]. In the case of the adult dataset described in section 4.1, we have the binary sensitive attributes *gender* and the categorical sensitive attribute *race*, both of which are known to experience discrimination in income.

[4] https://bit.ly/3Ko10sL

A challenge when working with multiple sensitive attributes and multivariate sensitive attributes is that you get overlapping groups. See figure 3.2 for an example. There we see that we have the binary attribute *Gender* and the categorical *Race* we have several subgroups, i.e., Black Women, White Male etc. An algorithm can be *independently group fair* when you calculate fairness for each sensitive attribute independently, and/or *intersectional group fair* when you calculate fairness for all subgroups [30]. In this thesis, we use both approaches to calculate parity.

3.4.1 Scoring Function: Demographic Parity Score

We defined a new scoring function based on demographic parity in equation 2.1, where $\hat{Y}$ is the predictor and S the sensitive attribute. This equation can be generalised to the case where we have a categorical sensitive attribute with K classes.

$$P(\hat{Y}|S_i) = P(\hat{Y}|S_j) \qquad i, j \in \{0, \ldots, K-1\}, \qquad i \neq j$$

When calculating the probabilities, we want to condense these probabilities to a single metric between 0 and 1. I.e, when we have the likelihood of a positive outcome for the different classes of a sensitive attribute in a list of probabilities L like so

$$L = \{P(\hat{Y} = 1|S = 0), \ldots, P(\hat{Y} = 1|S = K-1)\}$$

We want a function f that takes such a list and maps it to a real number between 0 and 1

$$f : L \to [0, 1]$$

It is important that this function works for a list of likelihoods of arbitrary length, since we want to evaluate demographic parity for both binary and categorical sensitive attributes as well as all intersections of the sensitive attributes. Given the definition of demographic parity, we want the likelihoods to be as equal as possible. The scoring function derived is shown below

$$f = 1 - 2\sigma(L)$$

where σ denotes the standard deviation. We plotted the scoring function in figure 3.3. The scoring function receives a list of two probabilities which slowly diverges from being

L	f
[0.5, 0.5]	1
[0.6, 0.7]	0.9
[0.6, 0.8, 0.4]	0.67
[0, 1, 0, 1]	0

Table 3.1: Example of values of f given different lists of likelihoods.

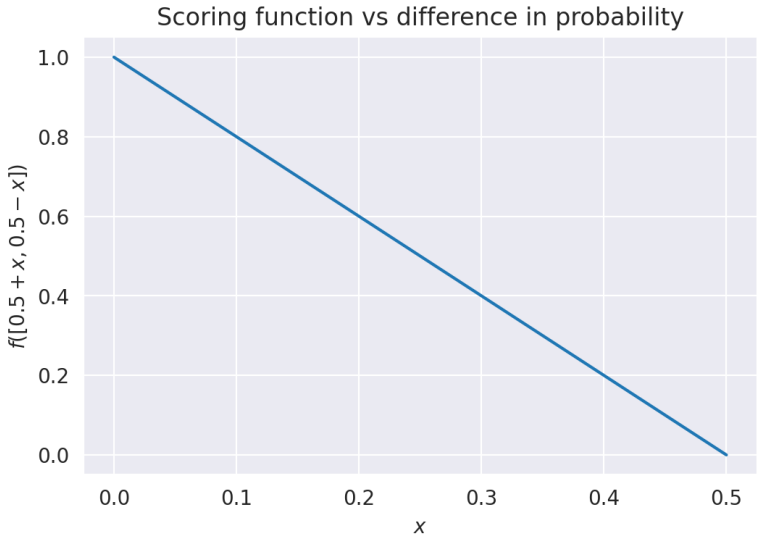


Figure 3.3: Scoring function for a list of two diverging probabilities

equal at 0.5 to the complete opposite, i.e $[0, 1]$. When the probabilities are equal, the score is 0 and if they are very different, the score is 0. See the examples of scores in table 3.1

3.5 Fair Bayesian Network

Based on the fair Bayesian network described by Choi et al. [20], the following algorithm was derived.

This method uses two methods implemented in pgmpy [31]. These are Hill Climb Search for learning the structure which is described in section 2.5.3 and Expectation Maximisation which is described in section 2.5.4. The resulting Bayesian network on the adult dataset is shown below in figure 3.4.

Inference has been performed by evaluating $P(F|A, E, R_e, C, M, W, H, O)$ on a test dataset of unobserved labels.

Algorithm 3.1 Latent Label Classifier Training

Input: D = Training Dataset, S = Sensitive Attributes, L = attribute to predict. t = tolerance for expectation maximisation.
Output: M = Trained Bayesian network.
Set of blacklisted nodes: $B = \{(x, y) \quad \forall x \in D.\text{columns}, \quad \forall y \in S\}$
M.structure = HillClimbSearch(D.columns $\setminus L$, B)
M.structure$\cup\{(x, L) \quad \forall x \in$S$\}$
initialise fair node F
M.structure$\cup\{(F, x) \quad \forall x \in D.\text{columns}\}$
M.structure$\cup\{(F, L)\}$
M.parameters = ExpectationMaximisation(M.structure, D)
return M

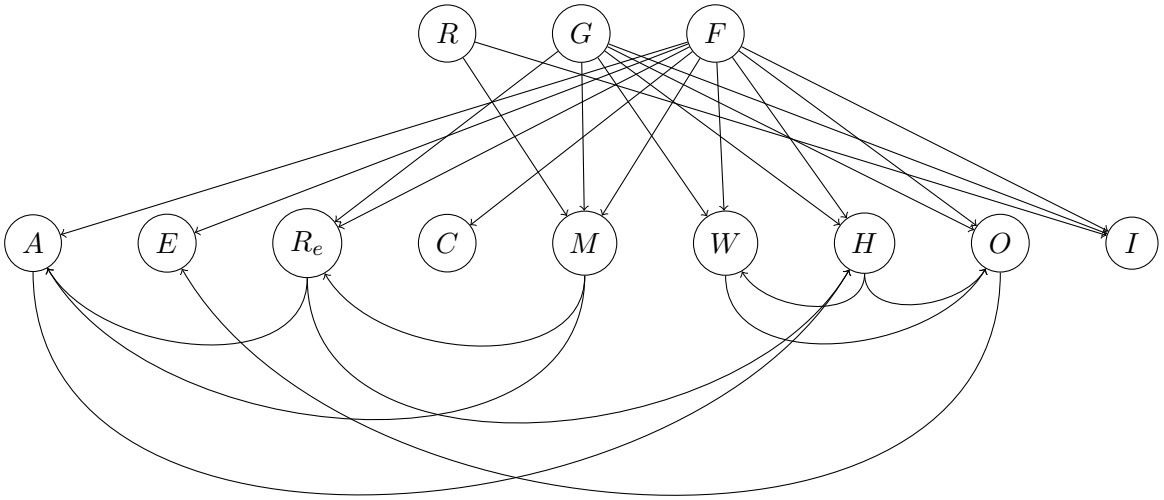


Figure 3.4: latentFairClassifier trained on Adult Dataset. The nodes are R: race, G: gender, F: latent fair labels, A: Age, E: Education, R_e: Relationship, C: Capital gain, M: Marital Status, W: Work class, H: Hours-per-week, O: Occupation and I: Income.

3.6 Fair Tree Classifier

The fair tree classifier, described in detail in section 2.7. Barata and Veenman [16] kindly provided their code on GitHub, which made implementing this over to Forseti a bit easier. The code is available in their repository[5]. We added their decision tree classifier and fair random forest classifier classes to Forseti and train and test the models on the same datasets as the fair tree classifier.

[5] https://bit.ly/3x937Nr

3.7 Experiment 1: Test Models on COMPAS and Adult Dataset

After implementing the Fair Bayesian Network and the Fair Tree Classifier. We wanted to evaluate the models on the Adult Dataset and COMPAS Dataset. To evaluate the models we had to choose some performance metrics. A combination of traditional performance metrics and a fairness score was desired. The fairness score used is the one described in section 3.4.1. For the traditional metrics there were several pros and cons for each of them. We will go through each one of them and explain the motivation for using that particular scoring method.

3.7.1 Accuracy

We use the accuracy score as implemented in Scikit-Learn. Which follows the following formula

$$A(y, \hat{y}) = \frac{1}{n_{\text{samples}}} \sum_{i=0}^{n_{\text{samples}}-1} 1(\hat{y}_i = y_i)$$

Accuracy is a very intuitive coring function for predictions, as it can be interpreted as the ratio of correct classifications. The problem with accuracy as a scoring function is when datasets are imbalanced, the scoring function also suffers and are biased toward the most prominent class.

3.7.2 Balanced Accuracy

Balanced Accuracy is calculated as follows

$$BA(y, \hat{y}) = \frac{1}{\sum \hat{w}_i} \sum_i 1(\hat{y}_i = y_i)\hat{w}_i$$

where $\hat{w}_i$ is sample weight of the i-th sample and the weight is adjusted to

$$\hat{w}_i = \frac{w_i}{\sum_j 1(y_j = y_i)w_j}$$

Balanced Accuracy is equal to the arithmetic mean of the sensitivity and specificity in the binary case. Since this accuracy score scales with imbalanced datasets it gives a more clear picture of how well the classifier performs and adjusts if the classifier takes advantage of the imbalance.

3.7.3 F1 Score

The F1 score is defined as the harmonic mean of precision and recall and thus reflect. We chose to add F1 score as it is a commonly used score in machine learning and the score reflects how well the model is able to maximise precision and recall and not have a huge disparity between them. One challenge with F1 score is that it ignores true negatives.

3.7.4 Specificity

And lastly, we calculate the specificity (true negative rate). Which is calculated from the confusion matrix as follows

$$TNR(\hat{y}, y) = \frac{TN}{N}$$

We mainly chose this metric since F1 score ignores true negatives and by having this in our fairness report we have some information with regard to true negatives.

3.7.5 ROC Curve

We also want to calculate the ROC curve and plot it to see the threshold independent performance of the model. This requires us to also get the predicted probability of positive outcome for the models. When we have the predicted probabilities and the true class labels. We use the different probability scores as thresholds, sort them and calculate the TPR and FPR for each probability score and plot TPR with FPR to produce the ROC curve. We use the method *plot_roc_curve* in sklearn.

3.7.6 Model Selection

For the first experiment, we also want to compare our fair models with some baseline models. Additionally, we want to explore the hyperparameter-space of models that have hyperparameters. This led us to the following models to evaluate

- *FairBN*: Training a Fair Bayesian Network and doing inference on the latent fair variable.
- *IncomeBN*: Same model as FairBN. prediction is done on the variable Income instead of the latent fair attribute.

- *NBSens* is a naïve Bayes classifier trained on all of the attributes available.
- *NB* is a naïve Bayes classifier trained on the dataset without the sensitive attributes.
- *FRFC03*: Fair Random Forest classifier with $\Theta = 0.3$
- *FRFC05*: Fair Random Forest classifier with $\Theta = 0.5$
- *FRFC07*: Fair Random Forest classifier with $\Theta = 0.7$

naïve Bayes with and without serves as the baseline method. Removing the sensitive attributes serves as a first attempt at achieving fairness by not allowing the model to explicitly use the sensitive attributes for prediction. We expect to see that the fair machine learning methods performs better in regard to fairness to the naïve Bayes model. Otherwise, there is no reason to use the fair methods. The results of the first experiment are shown in section 4.2.

3.8 Experiment 2: Performance on synthetic dataset and comparison of performance metrics.

3.8.1 Motivation for experiment

The results in the first experiment shed some light on the challenge of calculating fairness. The fair Bayesian network classifier got higher parity scores as expected but for the random forest classifier, unexplained behaviour of the model with respect to the hyperparameter was observed. Since the parity score used in this experiment is a new and self proposed metric of fairness, this should be investigated further to evaluate its validity.

Barata and Veenman [16] claim that they have made a classifier able to give more fair predictions, and using a hyperparameter Θ that give more accurate predictions when $\Theta \to 0$ and more fair predictions when $\Theta \to 1$. We fail to reproduce these results when calculating parity score for the models. The authors themselves do not calculate intersectional parity score, but rather AUC_S with respect to the individual sensitive attributes. Due to these observations, we chose to focus on how to calculate fairness and what are the limitations of the respective fairness metrics for the next iteration of experiments.

The two models that have been implemented in the first round of experiments, namely the fair Bayesian network introduced by and the Fair Tree Classifier introduced by.

These classifiers have two different ways of evaluating fairness that is the motivation behind their design. In the work of Choi et al. [20] the model was designed in terms of demographic parity by modelling a Bayesian network in such a way that the sensitive attributes are independent of the predictions but still used for learning the parameters of the distribution of fair labels. In the work of Barata and Veenman [16] they use strong demographic parity as motivation for their model. They implement a tree based method where splits are evaluated using the threshold-independent measure of AUC with respect to the labels of the dataset and regularised using AUC with respect to the sensitive attributes.

3.8.2 Experiment Design

There are some questions that have arisen from the first round of experiments and some improvements that we wanted to implement. These are

1. Does demographic parity score capture fairness?
2. What other measures of fairness can we introduce that are more well established?
3. How do the different performance metrics used by the different authors compare?
4. Collect enough samples to perform hypothesis testing.

3.8.3 Generating Synthetic Data

To address question 1, we wanted to generate a synthetic dataset where we are in control of the bias in the data to see how the fairness measures fares. We implemented an algorithm for generating synthetic datasets with numerical (Gaussian) features as well as two sensitive features, gender, and race. The dataset can be either informative or non-informative. If the dataset is informative, the features are dependent on the sensitive attributes. I.e., there is bias in the data with respect to the sensitive attributes. We generate the dataset in the following way. The first attribute, gender G_s, is assumed to follow a Bernoulli distribution

$$G_s \sim B(n, p)$$

where $n = 1$ and $p = 0.5$. I.e. we assume that it is equally likely to be a male or female. The second attribute, Race R_s is assumed to follow a categorical distribution

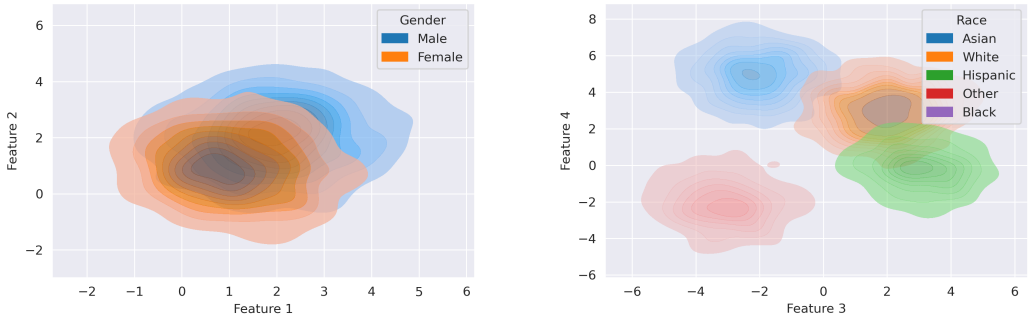


Figure 3.5: KDE of the bivariate features of the synthetic dataset..

$$R_s \sim C(k, \boldsymbol{\theta})$$

Which has PMF

$$f_{R_s}(R_s = i) = \boldsymbol{\theta}_i, \qquad i \in \{1, \ldots, k\}$$

where $\boldsymbol{\theta}$ is a vector of k event probabilities p_i. $(p_i \geq 0, \sum p_i = 1)$. We want to initialise $\boldsymbol{\theta}$ randomly and this is done by sampling k numbers from τ which follows the standard half-normal distribution

$$\tau \sim |N(0, 1)|$$

And $\boldsymbol{\theta}$ is then calculated by arranging the k samples from τ in a vector and transforming them to probabilities

$$\boldsymbol{\theta} = (\tau_1, \ldots, \tau_k) \cdot \frac{1}{\sum \tau}$$

The numerical values of the sensitive attributes are mapped to strings using a dictionary in python. Now that the sensitive attributes are sampled, we will sample the non-sensitive features. To simulate the discrimination process, we sample the features from Gaussian distributions that depend on the sensitive attributes. There are 4 numerical features where the first two depend on the gender and the last two depend on the race. Let's denote these features X_1, X_2, X_3 and X_4.

We define a separator parameter Θ which is used to model the separation between the sensitive classes. We sample X_1 and X_2 the following way conditionally on the sensitive attribute G_s

$$X_1|G_s = \text{Female} \sim N(1,1)$$
$$X_1|G_s = \text{Male} \sim N(1+\Theta,1)$$

And the same for X_2. The same principle is used for X_3 and X_4 but with a slight twist. There are k races in the dataset and a random sign variable R following the discrete uniform distribution over the set $\{-1,1\}$.

$$X_3|R_s = k \sim N(1+\Theta \cdot R \cdot k, 1)$$

and the same for X_4. Then lastly, to calculate the labels of the dataset we take the sum of all the numerical features $X = \{X_1, X_2, X_3, X_4\}$ and calculate the median of all the sums for each data point. If the sum is above the median, it gets labelled 1, otherwise, 0.

This can be summarised in algorithm 3.2

Algorithm 3.2 Synthetic dataset generation

Input: I = Informative. True or false, Θ = Separability, N = Number of samples.
Output: D = Synthetic Dataset.

Sample N samples from $G_s \sim B(1, 0.5)$
$k = 5$
$\boldsymbol{\tau_k} = \{\tau_1, \ldots, \tau_k\}$ with $\tau_j \sim |N(0,1)|$
$\theta = \boldsymbol{\tau} \cdot \frac{1}{\sum \tau_k}$
Sample N samples from $R_s \sim C(k, \theta)$
Sample N_{Female} samples from $X_1 \sim N(1,1)$
Sample N_{Male} samples from $X_1 \sim N(1+\Theta,1)$
Sample N_{Female} samples from $X_2 \sim N(1,1)$
Sample N_{Male} samples from $X_2 \sim N(1+\Theta,1)$
for all $i \in \{1, \ldots, k\}$ **do**
  Select R uniformly from the set $\{-1,1\}$
  Sample N_i samples from $X_3|R_s = i \sim N(1+\Theta \cdot R \cdot i, 1)$
  Select R uniformly from the set $\{-1,1\}$
  Sample N_i samples from $X_4|R_s = i \sim N(1+\Theta \cdot R \cdot i, 1)$
end for
$t = \sum X$ for all data points in the dataset
if $t \geq \text{Median}(X)$ **then**
  $Y = 1$
else
  $Y = 0$
end if
$D = \{G_s, R_s, X_1, X_2, X_3, X_4, Y\}$
return D

3.8.4 Fairness Performance Measure

The performance metric used for the models up to this point is the parity score metric as described in section. 3.4.1. This score is based on the notion that the likelihood of positive outcome is independent of the sensitive attributes. There is another measure that is similar to this. That is Kullback-Leibler divergence. Which according to MacKay et al. [32] is calculated as

$$D_K L(P||Q) = \sum_x \log \frac{P(x)}{Q(x)}$$

Kullback-Leibler divergence is a measure of how different two distributions P and Q are. It is not strictly a metric as it is not symmetric and is instead a divergence. While metrics are symmetric and linear in distance, divergences are asymmetric and generalise square distance. For our work, we want to use Kullback-Leibler divergence as a measure of unfairness. The general idea is that we want P and Q to denote different distributions conditional on sensitive attributes. Assume that we have a dataset with model predictions Y and a binary sensitive attribute S. Then we define

$$P \sim Y|S = 0$$

and

$$Q \sim Y|S = 1$$

Under demographic parity, we want $Q \perp\!\!\!\perp P, S$ and thus a Kullback-Leibler Divergence of 0. Kullback-Leibler Divergence is limited to two distributions, though there are generalisations. For this round of experiments, we limit ourselves to the binary case and calculate the divergence in the binary cases and compare them to the other metrics.

3.8.5 Hypothesis Tests

We want to perform hypothesis testing on the performance metrics of different models. Assume that we have samples of an unspecified performance metric for two different models trained on the same dataset. Let us denote the two distributions X and Y.

We have chosen to use non-parametric hypothesis tests, as we do not want to make assumptions on the kind of distribution the samples have. Since the samples are of model

Hypothesis	H_0	H_1	α
FairBN (Y) has a better intersectional parity score than naïve Bayes (X) without sensitive attributes	$X = Y$	$X < Y$	0.01
FRFC (Y) has a better intersectional parity score than naïve Bayes (X) without sensitive attributes	$X = Y$	$X < Y$	0.01
FairBN (Y) has a better intersectional parity score than FRFC (X)	$X = Y$	$X < Y$	0.01
FairBN (Y) has a better KLD w.r.t. Gender than FRFC (X)	$X = Y$	$X > Y$	0.05

Table 3.2: List of hypothesis

performance of two machine learning models trained on the same dataset, the samples will be dependent on each other. We therefore select the Wilcoxon signed-rank test as the testing method of choice. This test was introduced and named after Wilcoxon [33]. The Wilcoxon signed-ranked test calculates the differences between the ranked samples and test whether or not the differences are symmetric around zero. We will test the following hypotheses, shown in table 3.2.

3.8.6 Experiment Setup

We ran 100 synthetic dataset generations and trained the models on the synthetic datasets and evaluated them on a test dataset. For each iteration the train-test split is 70% for the training set and 30% for the test dataset. We calculated the same performance metrics as in the first experiment. Additionally, we also introduce Kullback-Leibler Divergence w.r.t. Gender.

After the performance metrics have been calculated and collected. We will visualise the distributions of the performance metrics as well as the correlation between parity score, KL Divergence and AUC score. The results are shown and discussed in section 4.3

3.9 Experiment 3: Interpretable Machine Learning

We have been able to show that the models we have trained are able to satisfy some mathematical notion of fairness in their predictions. But we want to investigate this further. By employing interpretable machine learning methods, we can try to explain how the model uses the sensitive attributes in their predictions and try to explain their behaviour.

3.9.1 Experiment Design

For this experiment, we want to implement the following in an attempt to explain the decisions made by machine learning models

- Train a baseline interpretable model and investigate the decision rules it learns.
- Interpret the models that are interpretable by default.
- Use model agnostic interpretable methods for models that are not interpretable.

In the following sections, we will describe the methods chosen to answer the above points.

3.9.2 Interpreting Decision Trees

To address the first point, we want to train a baseline interpretable model. Our choice landed on Decision Trees. According to Molnar [29] the interpretation of decision trees, while quite similar for all algorithms, differ by the kind of algorithm for building the tree is used. The interpretation described here is for the CART algorithm, which is the one implemented by Scikit-learn [26].

The interpretation of decision trees works as follows: Starting from the root node, you go to the next nodes and the edges tell you which subsets you are looking at. Once you reach the leaf node, the node tells you the predicted outcome [29]. To calculate the feature importance. We calculate the reduction in the split. In our case, we have used entropy and information gain to evaluate splits which is defined as

$$IG(T) = H(T) - \sum_{i=1}^{2} \frac{n_i}{n} H(C_i)$$

Where T is the node being split and C_i is child node i. H denotes Shannon entropy introduced by Shannon [34]. Go through all the splits for which the feature was used and measure how much it has reduced the information gain. Scale all the sums to 1, then you can calculate the share of a features importance as an percentage.

We will do this for the datasets that have been used in the previous experiments to uncover what attributes are used in the model prediction.

3.9.3 Interpreting naïve Bayes

As described in section **??**, the naïve Bayes classifier assumes that all features are mutually independent, conditioned on the class labels. We can therefore interpret the contributions of the different attributes through the conditional probabilities that the naïve Bayes classifier has learned [29, p. 142].

The way we have chosen to do this is like so: Assume that a feature X_i in a dataset $\boldsymbol{X}$ is informative. Then we would expect that the likelihood on X_i is very different given the class Y

$$P(X_i|Y=0) \neq P(X_i|Y=1)$$

In the case that X_i is categorical with k categories. We have a conditional dependency table on the form

$$\begin{bmatrix} P(X_0|Y=0) & P(X_0|Y=1) \\ P(X_1|Y=0) & P(X_1|Y=1) \\ \vdots & \vdots \\ P(X_k|Y=0) & P(X_k|Y=0) \end{bmatrix}$$

Then we denote the first column $P = P(X_i|Y=0)$ and the second column $Q = P(X_i|Y=1)$. We assume that the more informative X_i is as a feature, the difference in these distributions should increase. To calculate the feature weight we define the weight as

$$D_{KL}(P||Q)$$

This should also be validated using permutation importance to compare the weights. Introduced by 3.3 is used and is based on the one used by Scikit-learn [26]

This algorithm returns a dataset of importances, as we want to see the distribution of importances for each feature.

3.9.4 Interpreting Bayesian Networks

Bayesian Networks are to some extent interpretable machine learning models, dependent on the complexity of the conditional dependencies. Naïve Bayes is an interpretable machine learning model and is the simplest Bayesian network. It is very intuitive

Algorithm 3.3 Permutation Importance Algorithm

Input: C = Classifier, D = Test Dataset. K = No of permutations.
Output: I = Dataset of importances.

```
Compute reference score S on the test dataset D.
for all columns i ∈ D do
  for j ∈ {1, ..., K} do
    Permute (shuffle) column D_i and denote the corrupted dataset D̃_j
    Compute score s_ij on test dataset D̃_j
    Set I_ij = S − s_ij
  end for
end for
return I
```

for humans to explain how much each attribute contributes, as they are conditionally independent given the class label. As the complexity of the conditional dependencies increase, they are more demanding to understand.

In our case, from the fair Bayesian network as shown in figure 2.2. We see that the sensitive attributes are independent of the true latent dataset labels. Meaning that the sensitive attributes should not be used explicitly in the prediction. To investigate this, we want to use the permutation importance algorithm to see how important the features are for getting accurate predictions. We will also apply counterfactual generation described in section 3.9.6 on the fair Bayesian network for further interpretability.

3.9.5 Interpreting Fair Random Forest

Random Forests are not interpretable as the many different trees used for predictions are not immediately intuitive to explain, and in many cases so many that the interpretation is incomprehensible in human terms. Therefore, one has to resort to model agnostic interpretability methods if one wants to gain some insight. We have chosen to use feature importance here as well in an effort to understand the model.

There is one challenge with the approach by Barata and Veenman [16], and that is the fact that the sensitive attributes are only used during training of the model. Meaning that one cannot infer how the model uses the sensitive attributes neither using the permutation importance method nor counterfactual generation. We will resort to investigating how the model uses the non-sensitive attributes in their prediction to investigate how the model achieves fairness.

3.9.6 Counterfactuals

According to Molnar [29], a counterfactual explanation should describe the smallest change to the feature values that changes the prediction to a predefined output. In our case for the Adult and the COMPAS dataset, we want to generate the smallest changes to get a positive outcome (being classified as high income or not likely to reoffend). Counterfactuals should follow these criteria

- Counterfactuals should be as similar as possible to the instance regarding feature values.
- Counterfactual instances should have feature values that are likely.
- Change as few features as possible.

Various methods exist for generating counterfactuals, while we will get inspiration from the method proposed by Dandl et al. [35], specifically the NSGA-II algorithm introduced by Deb et al. [36]. We want to generate counterfactuals that satisfy the following objectives

$$o_1(\hat{f}(\boldsymbol{x}), Y') = \begin{cases} 0 & \text{if} \hat{f}(\boldsymbol{x}) \in Y' \\ \inf\limits_{y' \in Y'} |\hat{f}(\boldsymbol{x}) - y'0|, & \text{else} \end{cases}$$

Which is the distance between the desired prediction y' and the predicted value $\hat{f}(\boldsymbol{x})$. The second objetive o_2 reflect that counterfactuals should be as equal to the instance $\boldsymbol{x}$ that we want to flip the prediction for

$$o_2(\boldsymbol{x}, \boldsymbol{x'}) = \frac{1}{p} \sum_{j=1}^{p} \mathbb{I}_{x_j \neq x'_j}$$

Which uses the indicator function, as the models implemented in pgmpy uses categorical data. Numerical values are also categorical as they are discretised. Next, we introduce o_3 which measures the amount of features that have been changes.

$$o_3(\boldsymbol{x}, \boldsymbol{x'}) = ||\boldsymbol{x} - \boldsymbol{x'}||_0 = \sum_{j=1}^{p} \mathbb{I}_{x_j \neq x'_j}$$

Lastly, we want the counterfactual to have attributes that are likely to occur. We measure this by searching for the closest sample in the training dataset and calculate the distance between the candidate and the closest point.

$$o_4(x', X^{\text{obs}}) = \frac{1}{p}\sum_{j=1}^{p} \mathbb{I}_{x_j \neq x_j^{[1]}}$$

And we try to minimise all these objective functions at once. To do this we will have to resort to a genetic algorithm as we do not want to collapse the four objectives into one but optimise all at the same time.

3.9.7 Nondominated Sorting Genetic Algorithm: NSGA-II

The NSGA-II algorithm introduce by Deb et al. [36] works as a four step iterative algorithm. The steps are as follows:

Initially, we generate a parent population that consists of N number of mutated copies of $\boldsymbol{x}$, the data point we want to change the outcome for. The mutations in the beginning are quite extensive so that we explore the optimisation space thoroughly. Then, we use fast-non-dominated-sort algorithm [36, p. 184] to rank the population into frontiers.

Then, we add each frontier in decreasing order into the new population P_{new} until there is no room to add the next frontier. Then we apply the crowding-distance-assignment method to add the remaining number of slots from the last frontier into P_{new} [36, p. 185]

We then have a new population $P = P_{\text{new}}$ and we create a new generation R which consists of mutated samples from P. We then rank $R \cup P$ using nondominated sorting and reiterate the steps above until the specified amount of iterations is completed.

The code for NSGA-II is added to the classes *interpretableNaiveBayes* and *latentLabel-Classifier* as a method in Forseti.

Chapter 4

Experimental Evaluation

4.1 Adult Dataset Data Exploration

- Age: The age of the individual (Positive integer)
- Work class: The sector the individual works in (8 Categories)
- fnlgwgt: A weight determined by the census bureau (Positive integer)
- Education: Highest educational degree (16 categories)
- Educational-num: Enumerated education (16 categories)
- Marital Status: Marital status of individual (7 Categories)
- Occupation: General type of occupation (15 categories)
- Relationship: What kind of relationship the individual is to others (6 categories)
- Race: What race the individual belongs to (6 Categories)
- Gender: Biological sex of the individual (2 Categories)
- Capital gain: Capital gain of individual (Positive integer)
- Capital loss: Capital loss of the individual (Positive integer)
- Native country: Native country of the individual (42 categories)
- Income: Whether individual makes more than 50K or not (2 Categories)

This dataset consists of mostly categorical attributes which are not ordinal. This makes analysis quite challenging. Many models assume Gaussian distributions, which is not

Attribute	Correlation
marital-status.Never-married	-0.318782
relationship.Own-child	-0.225691
relationship.Not-in-family	-0.190372
occupation.Other-service	-0.155254
relationship.Unmarried	-0.143642
education.HS-grad	-0.130706
race.Black	-0.090448
education.11th	-0.086728
occupation.Adm-clerical	-0.086475
relationship.Other-relative	-0.085601

Table 4.1: Features that are negatively correlated with income.

Attribute	Correlation
education.Masters	0.174184
education.Bachelors	0.180371
occupation.Prof-specialty	0.188793
occupation.Exec-managerial	0.210938
gender.Male	0.214628
capital-gain	0.223013
hours-per-week	0.227687
age	0.230369
marital-status.Married-civ-spouse	0.445853

Table 4.2: Features that are positively correlated with income.

present in the dataset. In this dataset, there are also some sensitive attributes, most notably *Gender* and *Race*. One could also argue that *Marital Status* and *Relationship* could also be sensitive attributes.

In a fair machine learning system, we would expect that the outcome in terms of income does not depend on the race, gender, marital status or relationship or at the very least that the decision by the model is independent of the sensitive attributes.

4.1.1 Attributes correlated with income

We see in that there are some categories in the following attributes that are correlated with income

- Marital Status
- Age
- Hours per week

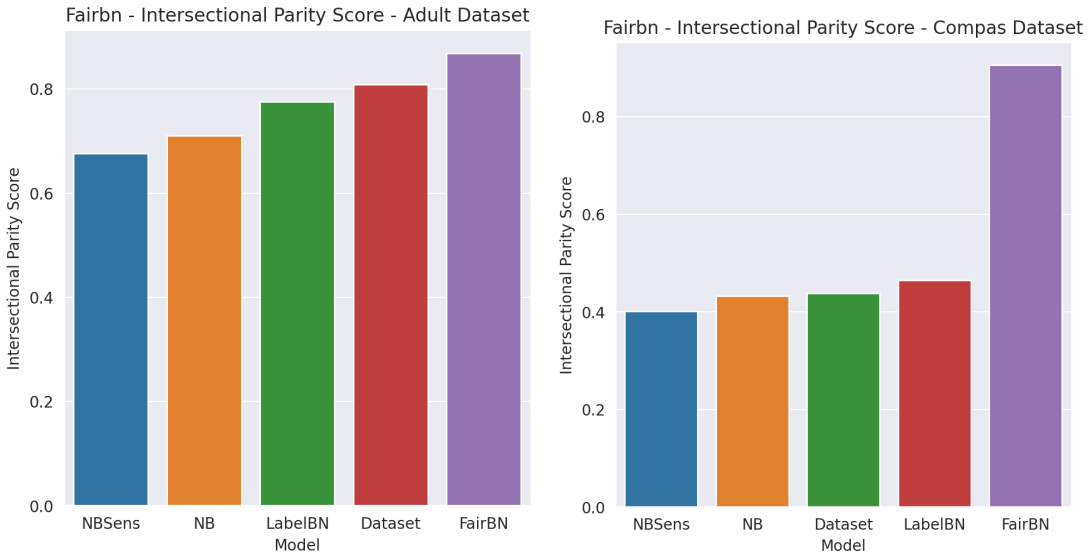


Figure 4.1: Intersectional Parity Score for fair Bayesian network vs naïve bays.

- Capital Gain
- Occupation
- Relationship
- Education
- Gender
- Race

We observe that our identified sensitive attributes are correlated with income. The challenge now is that we have to learn a model that does not treat individuals belonging to different classes in the sensitive attribute unfairly.

4.2 Experiment 1: FairBN, FairTreeClassifier vs NB

Below we will go through the different results of the first round of experiments. To see the detailed results, these are available in the appendix. See A.1

4.2.1 Fair Bayesian Network

After training the fair Bayesian network classifier on the adult and COMPAS dataset, we get an intersectional parity score of 0.87 and 0.91 respectively. This is better than

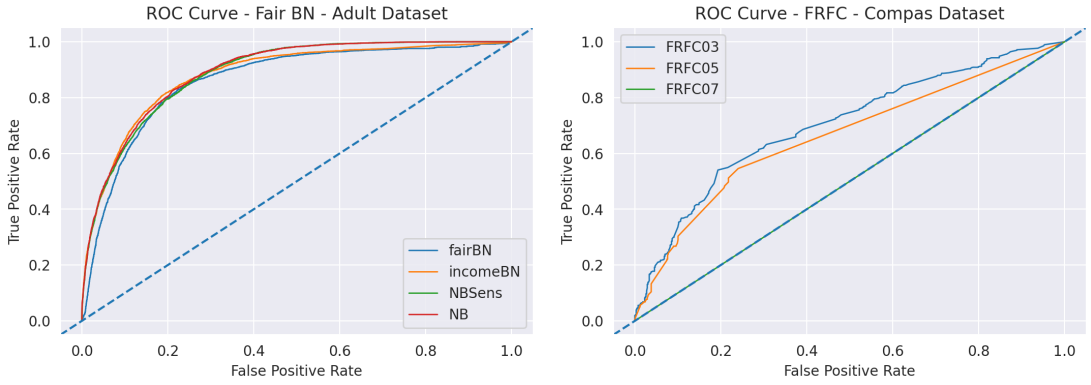


Figure 4.2: ROC curve for fair Bayesian network vs naïve Bayes.

the inherent parity score in the dataset labels, which is 0.81 and 0.51 respectively. These results and how the different methods compare to one another is shown in figure 4.1.

In terms of accuracy and traditional performance of the model, we observe that there is a tradeoff between performance and fairness. This is best shown in the ROC curve shown in figure 4.2. There is a slight drop in the fair Bayesian network compared to the naïve Bayes method.

We also observe a quite significant performance difference between the adult dataset and the COMPAS dataset. Why this is is not explored further in this thesis, as we are interested in seeing differences in performance with respect to fairness.

4.2.2 Fair Random Forest Classifier

We ran the same experiments using their classifier on the adult dataset and COMPAS dataset. Rather interestingly, we do not observe any improvement in intersectional parity in the adult dataset for any of the methods, with the inherent intersectional parity for the dataset being 0.810 and the best fair random forest classifier with $\Theta = 0.3$ having an intersectional parity of 0.78, which is counterintuitive to the claimed meaning behind the hyperparameter Θ stated by the authors.

For the COMPAS dataset, things are looking better with the models with $\Theta \in \{0.3, 0.7\}$ having higher intersectional parity scores than is inherent in the dataset labels. Still, the results given the stated meaning behind Θ is counterintuitive. The parity scores are shown in figure 4.3. The ROC curve for the different datasets is shown in figure 4.4

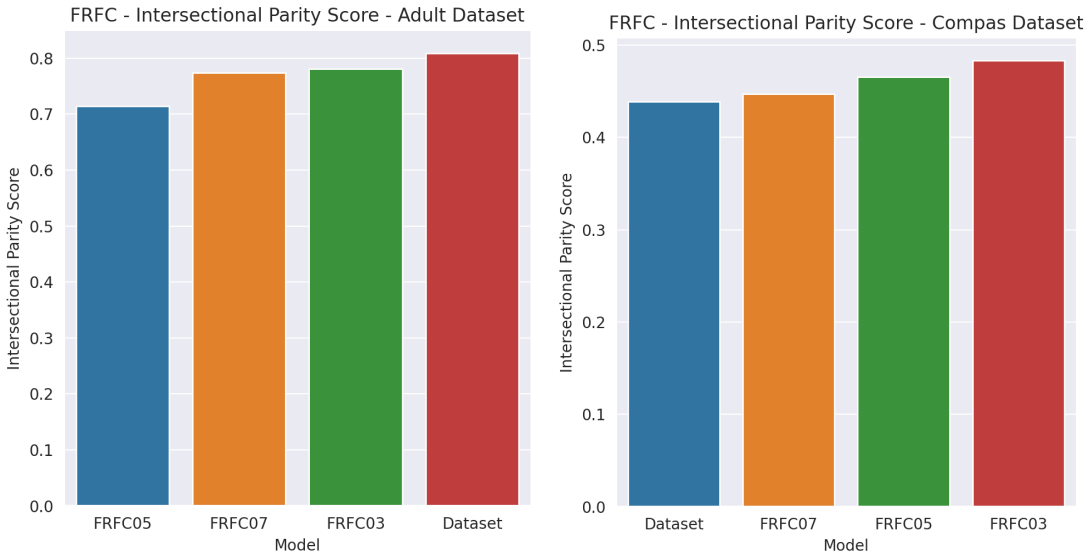


Figure 4.3: Parity score for fair random forest classifier.

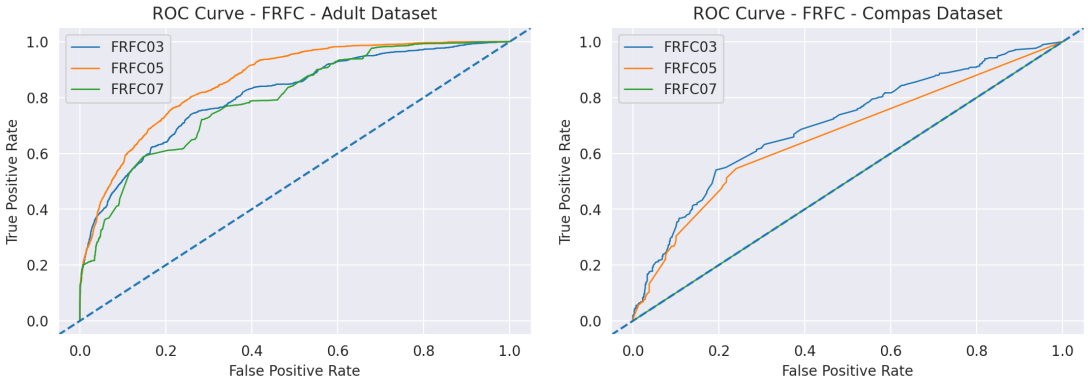


Figure 4.4: ROC curve for fair random forest classifier.

4.3 Experiment 2: Results

4.3.1 F1 Score

When evaluating the models in terms of F1 Score, we observe that the fair methods have lower but acceptable F1 Scores. For the fair Bayesian network classifier 95% of the F1 scores is at 0.4 or higher with mean at 0.6. For the fair random forest classifier, these values are 0.0 and 0.63. We see from the plot in figure 4.5 that the F1 score of the fair Bayesian network is on average lower than for fair random forest classifier but with lower variance.

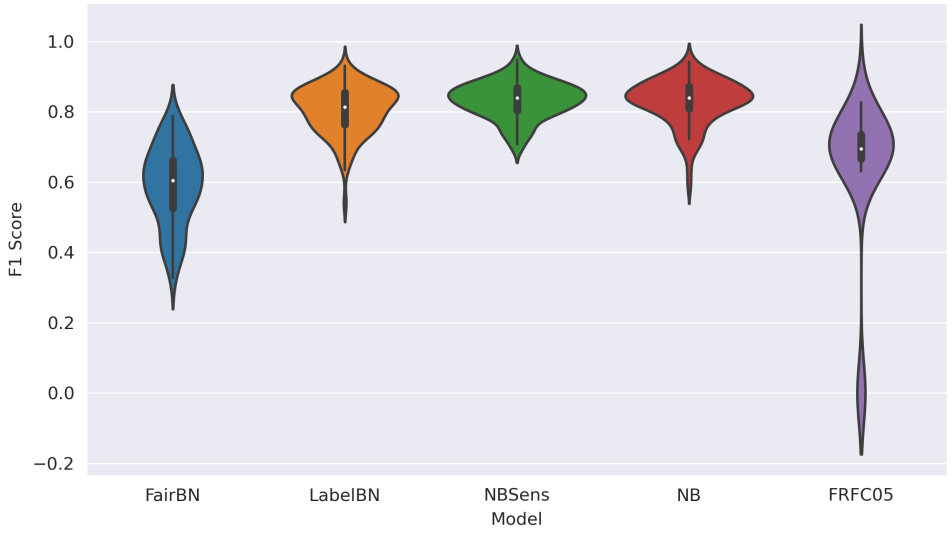


Figure 4.5: F1 Scores of the different models on 100 synthetic datasets. Higher is better.

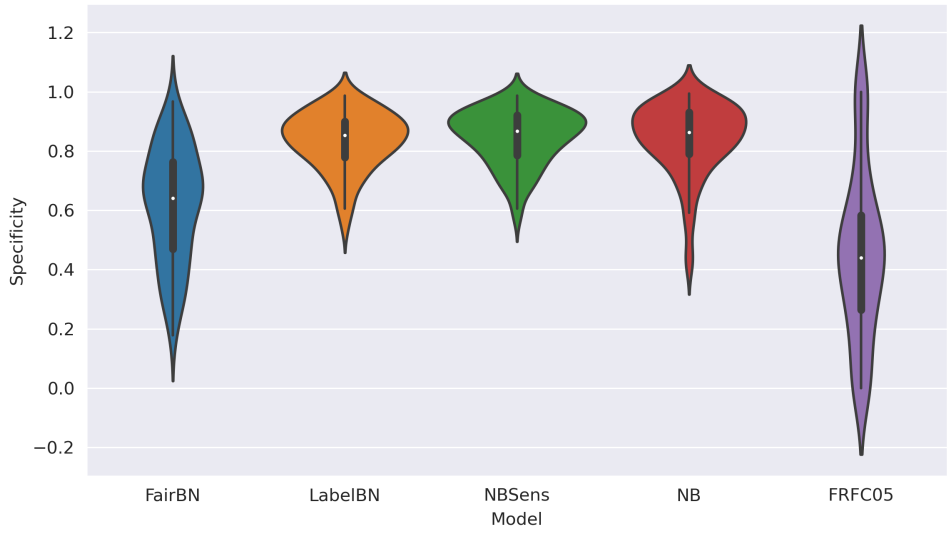


Figure 4.6: Specificity of the different models on 100 synthetic datasets. Higher is better

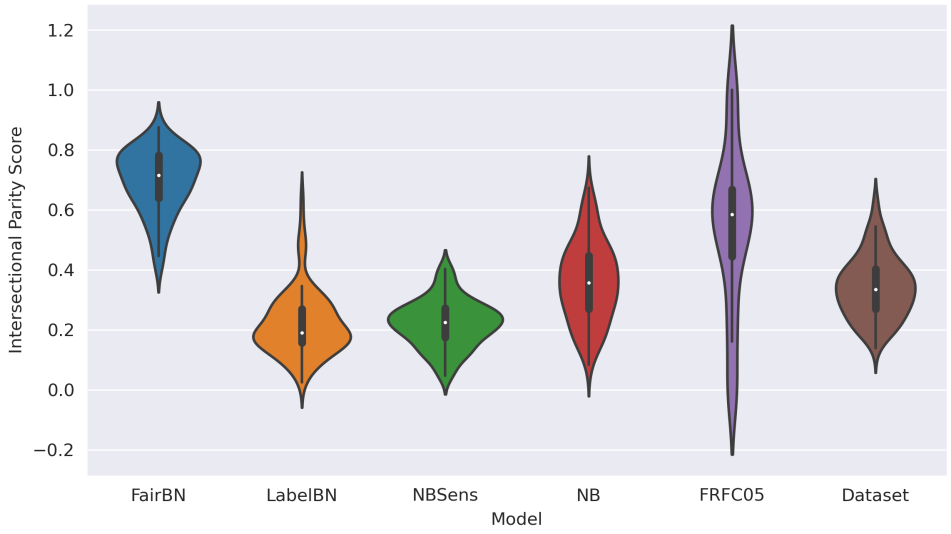


Figure 4.7: Intersectional Parity Score of the different models on 100 synthetic datasets. Higher is better.

4.3.2 Specificity

Since F1 Score is biased with respect to true negatives, we calculate the Specificity of the models as well. The distribution of specificity for the different models are shown in figure 4.6. We observe that the fair methods has a higher variance in their specificity as compared to the naïve Bayesian methods as well as the unfair predictions of the fair Bayesian network. 95% of the specificity values are at 0.3 or higher with mean at 0.61. For fair random forest classifier, these values are 0.0 and 0.44 respectively.

4.3.3 Intersectional parity score

In terms of intersectional parity score, i.e. average demographic parity across all subgroups of sensitive attributes, we see that the fair Bayesian network is able to consistently have more fair decisions. See figure 4.7. The mean parity score for the FairBN classifier is 0.70 with the 5th percentile at 0.50. For the fair random forest classifier these values are 0.54 and 0.0 respectively.

4.3.4 AUC Gender

In the paper by Barata and Veenman [16] they propose use AUC w.r.t. gender to evaluate splits which are more fair. Interestingly, while the fair random forest method is based on selecting splits that minimise the AUC w.r.t. gender, the model fares quite similarly

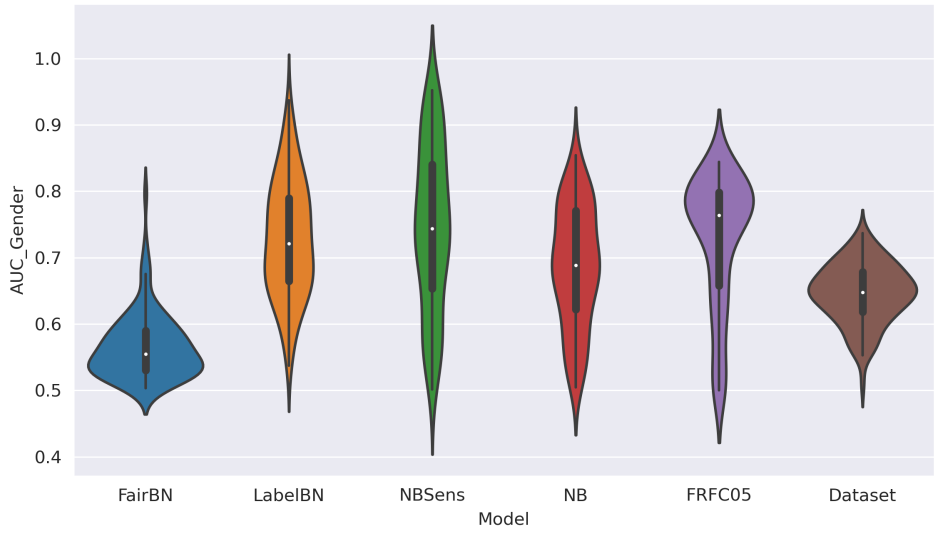


Figure 4.8: AUC w.r.t. Gender of the different models on 100 synthetic datasets. Lower is better.

to the Naïve Bayes models and incomeBN. See figure 4.8 The AUC is even higher than what is inherent in the labels of the dataset itself and predictions are in this sense more discriminatory than what is present in the true labels. The only model to minimise AUC more than the inherent dataset labels is FairBN. FairBN has a mean AUC of 0.56 and a 5th percentile of 0.51. While for the fair random forest classifier, these values are 0.72 and 0.51 respectively.

4.3.5 Kullback-Leibler Divergence

Lastly, we evaluate the models in terms of Kullback-Leibler Divergence. Here we see a significant difference between the methods. See figure 4.9The best performing model here is FairBN. With a 5th percentile value of 4.67×10^{-5} and mean 0.025. For the fair random forest classifier this is 0 and 0.21 respectively

4.3.6 Correlation between scoring methods

Demographic Parity v KL Divergence

From figure 4.10 we see that demographic parity and KL divergence are inversely related. As the parity score increases the divergence decreases. This gives us further confidence that both measure fairness in terms of demographic parity.

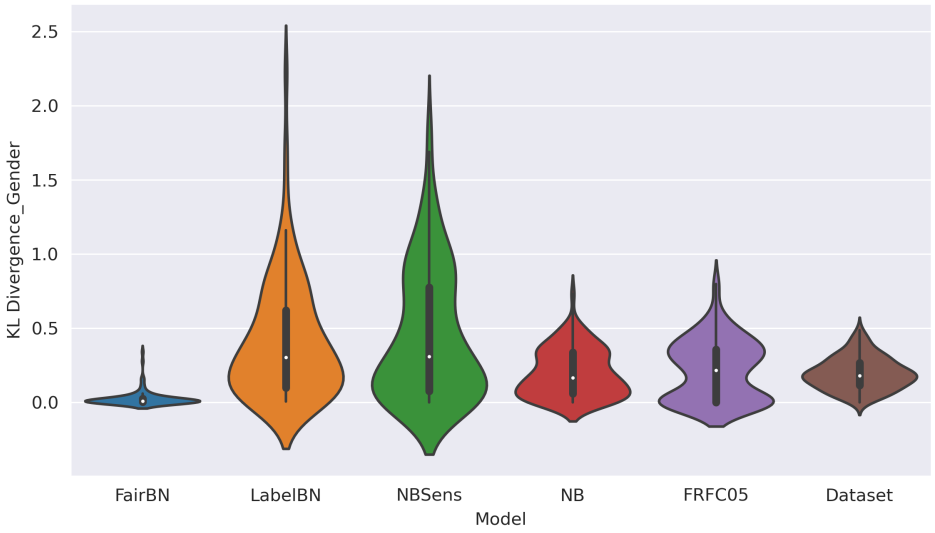


Figure 4.9: Kullback-Leibler Divergence of the different models on 100 synthetic datasets. Lower is better.

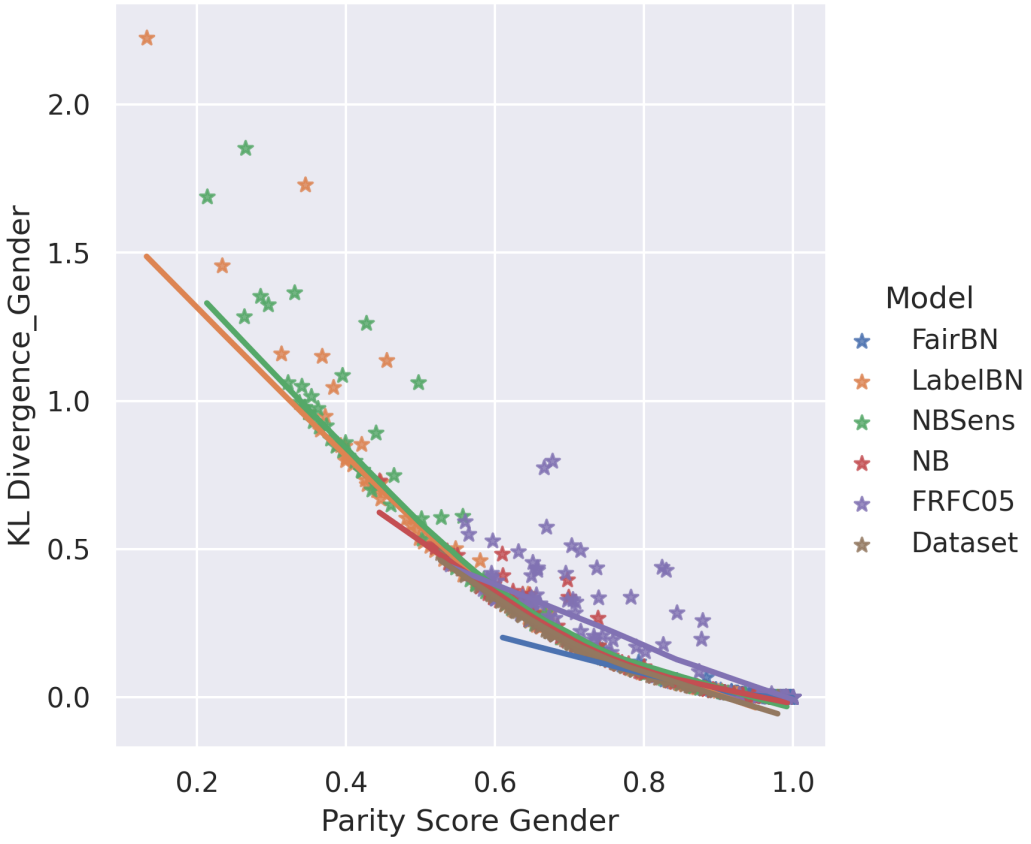


Figure 4.10: Demographic Parity and KL Divergence

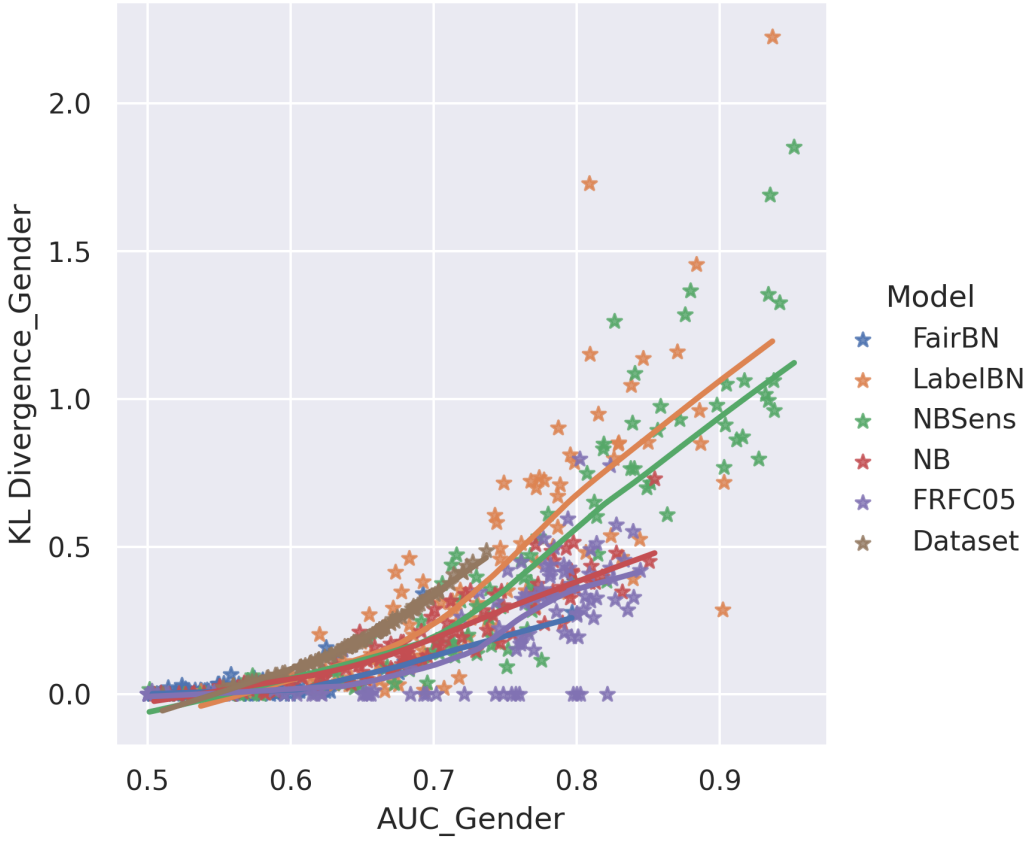


Figure 4.11: AUC and KL Divergence w.r.t. Gender

4.3.7 AUC and KL Divergence

From figure 4.11 we see that these scores are also correlated with one another, though not as strongly as with Parity Score and KL Divergence. This gives some validity to the use of AUC to train a fair random forest classifier.

4.3.8 Hypothesis Tests

As we see from the table above, we reject the null-hypothesis in all cases. This implies that both fair machine learning models have more fair predictions than the baseline method of using naïve-Bayes without the sensitive attributes. Additionally, we find that the fair Bayesian network has a more consistent and statistically significantly better score than the fair random forest classifier in terms of both parity score and KL Divergence.

Hypothesis	p	Reject H_0	T
FairBN (Y) has a better intersectional parity score than naïve Bayes (X) without sensitive attributes	4.13×10^{-18}	Yes	25
FRFC (Y) has a better intersectional parity score than naïve Bayes (X) without sensitive attributes	1.93×10^{-9}	Yes	812
FairBN (Y) has a better intersectional parity score than FRFC (X)	1.41×10^{-6}	Yes	1163
FairBN (Y) has a better KLD w.r.t. Gender than FRFC (X)	2.13×10^{-10}	Yes	4341

Table 4.3: Result of hypothesis tests

Attribute	Importance
marital-status_Married-civ-spouse	0.529924
capital-gain	0.347748
education_Bachelors	0.041893
hours-per-week	0.040666
age	0.038778
education_Preschool	0.000992

Table 4.4: Decision Tree: Adult Dataset - Feature Importance

4.4 Experiment 3: Results

4.4.1 Decision Tree: Adult Dataset

To gain insight into how the decision tree uses the attributes in its decisions, we calculate the feature importance of all attributes and show the ones that have positive feature importance. Feature importance is calculated as the normalised amount of reduction in entropy brought on by that attribute in the model. After training the decision tree classifier in Scikit-learn on the adult dataset. We got the following feature importance shown in table 4.4.

On the adult dataset we see that the decision rules considers whether or not someone has a civilian spouse as the most important variable to consider when classifying someone as high income or not. No sensitive attributes are considered important and when investigating the tree structure, no nodes employ the sensitive attributes.

4.4.2 Decision Tree: COMPAS Dataset

On the other hand, when training the decision tree classifier on the COMPAS dataset, we see that whether someone is male or considered to belong in the *Other* race category

	Attribute	Importance
0	priors_count	0.524409
1	age	0.417728
2	sex_Male	0.030951
3	juv_misd_count	0.011672
4	c_charge_degree_M	0.006723
5	race_Other	0.005893
6	juv_fel_count	0.002624

Table 4.5: Decision Tree: COMPAS Dataset - Feature Importance

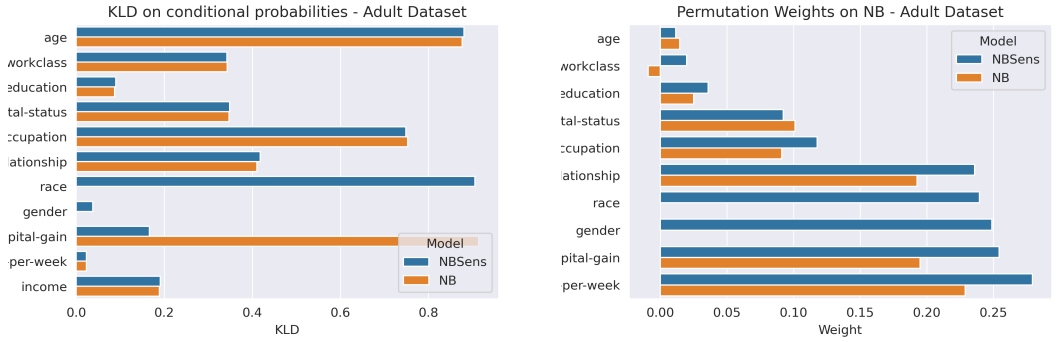


Figure 4.12: KLD and Permutation weights of naïve Bayes with and without sensitive attributes. Adult Dataset.

is used in prediction. This implies that there is a significant gender and racial bias in the data.

4.4.3 Naive Bayes: Feature Importance

To interpret the naïve Bayes models we look at the conditional distributions for the different attributes and compare calculate the KLD between the distribution $P = p(X|Y = 0)$ and $Q = p(X|Y = 1)$. We also calculate the permutation feature importance by calculating the loss in accuracy by permuting a specific column. The resulting weights are shown in figure 4.12 for the adult dataset and figure 4.13.

4.4.4 Fair Tree Classifier: Feature Importance

We also calculate the feature importance of the fair tree classifier. We see a similar relationship in the feature weights as in Naive Bayes but age and work class is completely neglected. Also, as Θ increases, overall weight of the attributes decreases.

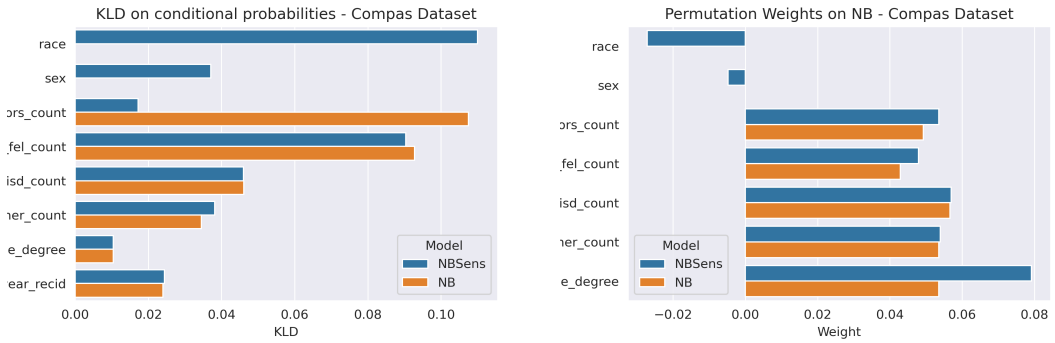


Figure 4.13: KLD and Permutation weights of naïve Bayes with and without sensitive attributes. COMPAS Dataset.

4.4.5 Individual Conditional Expectation

We have implemented ICE calculations for the classifiers Naive Bayes and Fair Tree Classifiers. As the sensitive attributes are not numerical but categorical, we do not resort to line plots of the predicted probability but rather categorical plots. Therefore, the distribution of predicted probability given the certain values is shown for the different models and datasets. Figure **??** shows how gender is treated by the different models in the Adult and COMPAS dataset. Figure 4.16 show how race is treated by the different models.

4.4.6 Counterfactuals

naïve Bayes With Sensitive Attributes

For the following data point:

	age	workclass	education	marital-status	occupation	relationship	race	gender	capital-gain	hours-per-week
46343	(31.6, 46.2]	Private	Assoc-voc	Divorced	Tech-support	Unmarried	Black	Female	(-4460.355, 16515.0]	(20.6, 40.2]

Got the following counterfactuals:

	age	workclass	education	marital-status	occupation	relationship	race	gender	capital-gain	hours-per-week	O1	O2	O3	O4
155	(31.6, 46.2]	Private	Assoc-voc	Married-AF-spouse	Tech-support	Unmarried	Black	Male	(79128.0, 99999.0]	(20.6, 40.2]	0.000000	0.7	3	0.0
162	(31.6, 46.2]	Private	Assoc-voc	Divorced	Adm-clerical	Unmarried	Black	Male	(79128.0, 99999.0]	(20.6, 40.2]	0.000000	0.7	3	0.0
100	(31.6, 46.2]	Private	Assoc-voc	Married-AF-spouse	Adm-clerical	Unmarried	Black	Male	(58257.0, 79128.0]	(20.6, 40.2]	0.000000	0.6	4	0.0
167	(31.6, 46.2]	Private	Assoc-voc	Married-AF-spouse	Adm-clerical	Unmarried	Black	Male	(79128.0, 99999.0]	(20.6, 40.2]	0.000000	0.6	4	0.0
176	(31.6, 46.2]	Private	Assoc-voc	Married-AF-spouse	Adm-clerical	Unmarried	Black	Male	(58257.0, 79128.0]	(20.6, 40.2]	0.000000	0.6	4	0.0
2	(60.8, 75.4]	Local-gov	Doctorate	Divorced	?	Not-in-family	Amer-Indian-Eskimo	Female	(79128.0, 99999.0]	(59.8, 79.4]	0.000000	0.2	8	0.0
177	(31.6, 46.2]	Private	Assoc-voc	Married-AF-spouse	Tech-support	Unmarried	Black	Female	(16515.0, 37386.0]	(20.6, 40.2]	0.419303	0.8	2	0.0

naïve Bayes Without Sensitive Attributes

For the following data point:

	age	workclass	education	marital-status	occupation	relationship	race	gender	capital-gain	hours-per-week
23356	(16.927, 31.6]	?	HS-grad	Separated	?	Unmarried	Black	Female	(-4460.355, 16515.0]	(20.6, 40.2]

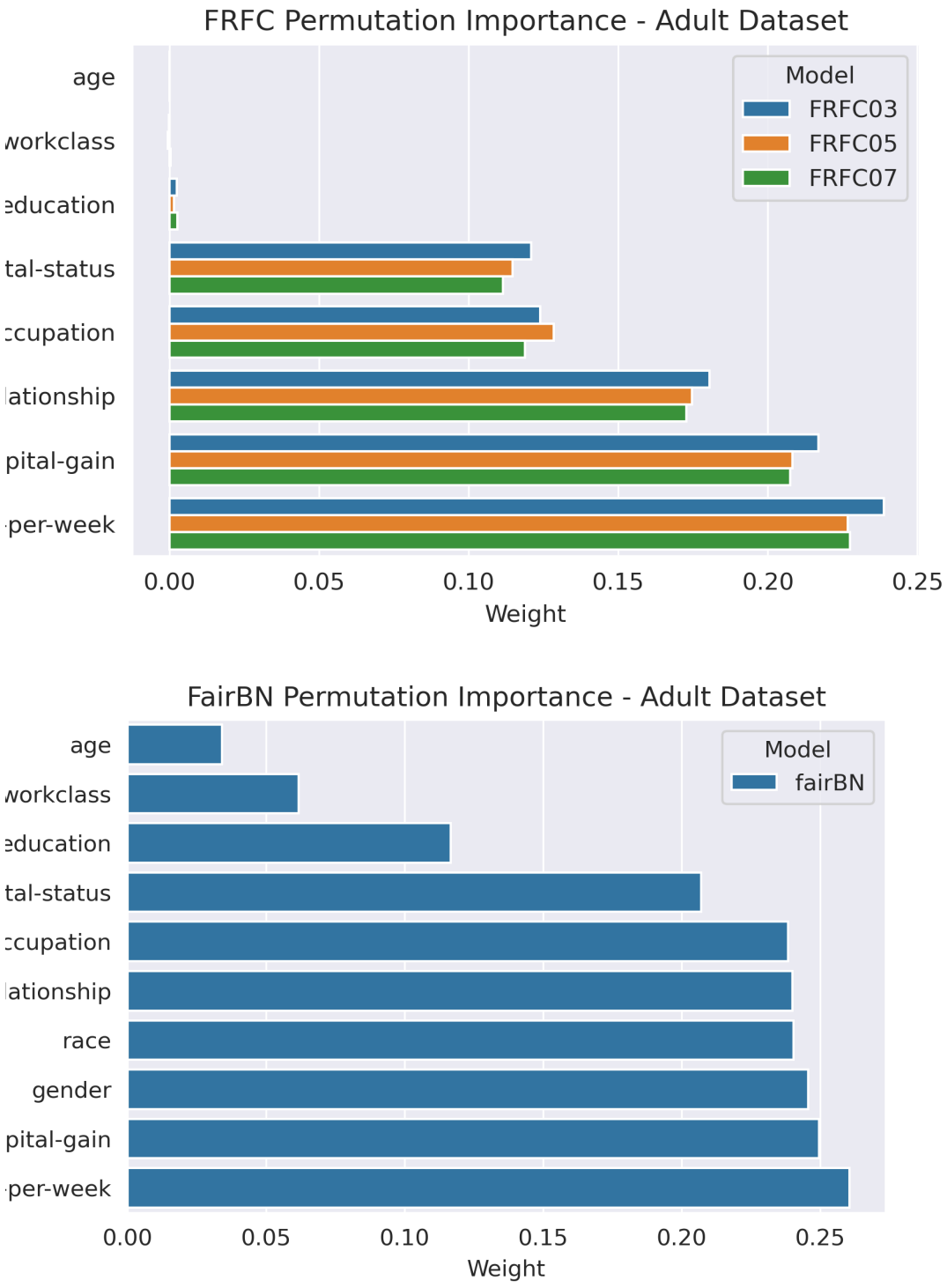


Figure 4.14: Caption

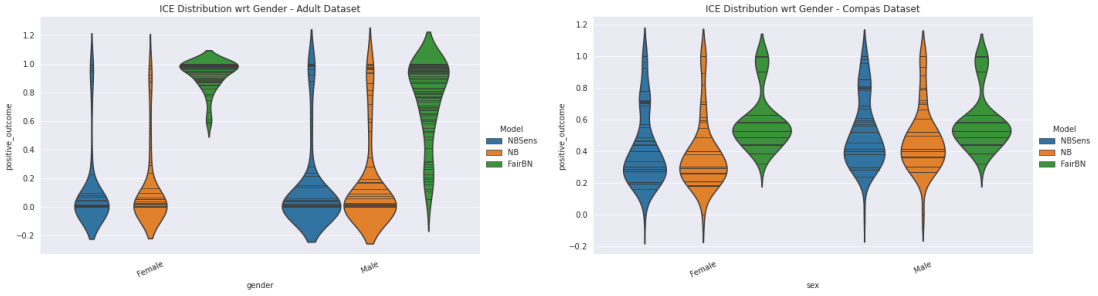


Figure 4.15: ICE distribution for Gender in the different datasets.

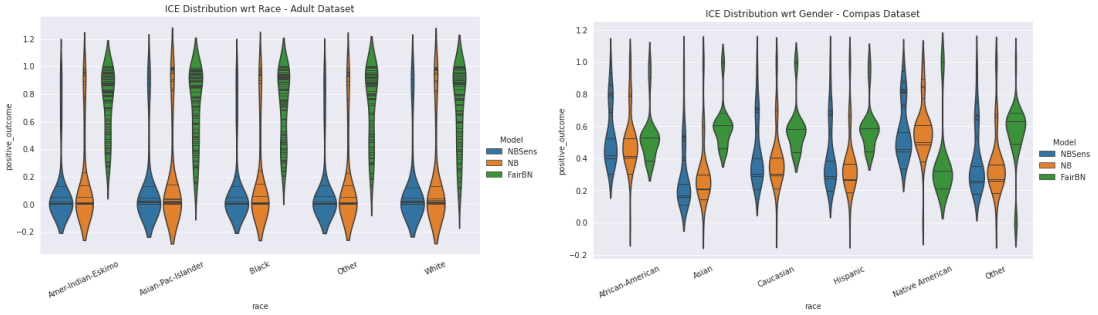


Figure 4.16: ICE distribution for Race in the different datasets

We get the following counterfactuals:

	age	workclass	education	marital-status	occupation	relationship	race	gender	capital-gain	hours-per-week	O1	O2	O3	O4
128	(16.927, 31.6]	State-gov	HS-grad	Separated	?	Unmarried	Black	Male	(79128.0, 99999.0]	(20.6, 40.2]	0.000000	0.7	3	0.0
97	(31.6, 46.2]	?	HS-grad	Separated	?	Not-in-family	Black	Male	(79128.0, 99999.0]	(20.6, 40.2]	0.000000	0.6	4	0.0
147	(16.927, 31.6]	State-gov	Doctorate	Separated	?	Unmarried	Black	Male	(79128.0, 99999.0]	(20.6, 40.2]	0.000000	0.6	4	0.0
136	(31.6, 46.2]	State-gov	Doctorate	Married-AF-spouse	?	Husband	Amer-Indian-Eskimo	Female	(-4460.355, 16515.0]	(20.6, 40.2]	0.165877	0.4	6	0.1

Fair Bayesian Network

For the following counterfactual

	age	workclass	education	marital-status	occupation	relationship	race	gender	capital-gain	hours-per-week
23356	(16.927, 31.6]	?	HS-grad	Separated	?	Unmarried	Black	Female	(-4460.355, 16515.0]	(20.6, 40.2]

We get the following counterfactuals

	age	workclass	education	marital-status	occupation	relationship	race	gender	capital-gain	hours-per-week	O1	O2	O3	O4
111	(16.927, 31.6]	?	11th	Separated	?	Unmarried	Black	Female	(79128.0, 99999.0]	(79.4, 99.0]	0.000000e+00	0.7	3	0.0
114	(16.927, 31.6]	?	HS-grad	Separated	?	Unmarried	White	Female	(79128.0, 99999.0]	(79.4, 99.0]	0.000000e+00	0.7	3	0.0
162	(16.927, 31.6]	?	11th	Separated	?	Unmarried	White	Female	(79128.0, 99999.0]	(20.6, 40.2]	0.000000e+00	0.7	3	0.1
58	(16.927, 31.6]	Never-worked	HS-grad	Separated	?	Unmarried	Asian-Pac-Islander	Female	(58257.0, 79128.0]	(79.4, 99.0]	0.000000e+00	0.6	4	0.0
118	(16.927, 31.6]	?	Assoc-acdm	Separated	?	Unmarried	Black	Male	(79128.0, 99999.0]	(79.4, 99.0]	0.000000e+00	0.6	4	0.0
172	(16.927, 31.6]	Never-worked	11th	Separated	?	Unmarried	White	Female	(58257.0, 79128.0]	(20.6, 40.2]	0.000000e+00	0.6	4	0.1

Chapter 5

Conclusions

5.1 Summary of the thesis

There is no question that fairness is important to incorporate in machine learning and decision-making systems if one wants to have the benefits of automation while at the same time achieve equality and sustainability for a better world. Machine learning model fairness and interpretability are vital for data scientists, researchers and developers to explain their models and understand the value and accuracy of their findings. Interpretability is also important to debug machine learning models and make informed decisions about how to improve them. While there exists many methods of achieving fairer systems, the field is still new and no state of the art exists. In this thesis, we have sought to investigate some current proposed methods for fair machine learning proposed in literature and try to shed some light on fairness in machine learning and what methods look promising or not. We have mainly focused on two approaches. The probabilistic Bayesian network approach and the fair tree classifier approach, which utilises two different methods for achieving fairness and also makes different assumptions.

5.1.1 Fair Bayesian Network

The fair Bayesian network, proposed by Choi et al. [20], which we have implemented in python using pgmpy in Forseti, makes the following assumptions

- Training Data is biased
- The true fair class affiliation is a hidden attribute and must be inferred rather than measured.

- Demographic Parity is the definition of choice for fairness.
- Sensitive attributes are important in the decision-making process as they contain necessary information of context.

5.1.2 Fair Tree Classifiers

While the fair tree classifier makes the following assumptions

- Training data is used as the true state of nature.
- Strong Demographic Parity is the definition of choice for fairness.
- Sensitive attributes should not be used during inference and used to evaluate splits during the training phase. Penalising the split if the sensitive attribute depends on the outcome.

5.1.3 Approach

We have trained these models extensively on the Adult dataset and COMPAS dataset which are established datasets in fairness research. We have proposed new performance metrics that incorporate fairness and evaluated models in terms of this. To validate these new metrics, we have proposed new algorithms for generating datasets where we are in control of the bias in the data to further evaluate these metrics.

In addition to training models and evaluating them in terms is new proposed fairness measures, we have tried to explicitly explain and understand the behaviour of the models to validate that models satisfying fairness measures actually truly achieves fair behaviour. In addition we have implemented an genetic algorithm approach to generate counterfactual data points that highlight what attribute changes flip the outcome of an individual. Which gives a quite intuitive overview of the inner workings of the models.

5.2 Findings

We have observed quite different behaviour from the two approaches investigated, while both claim to achieve fairness in their predictions, we have only been able to replicate these results with the approach proposed by Choi et al. [20]. In the approach proposed by Barata and Veenman [16] we have not been able to replicate the results presented in their paper.

From our work, we have shown that the fair Bayesian network is able to achieve fairer predictions while not suffering too much from the fairness-accuracy tradeoff. There is still a tradeoff, and the model does not achieve the same performance in terms of traditional performance measures as the naïve Bayes model. This is reasonable and expected from the model given that the model assumes that the dataset labels are biased and incorrect and only used to infer the true state of nature.

By employing machine learning interpretability methods, we have also shown that the fair Bayesian network achieves its fairness through affirmative actions. This is done by increasing the predictive probability of a fairer outcome, while giving neglected groups a higher chance of a positive outcome than expected from their representation in the training data.

From the same approach when looking at the fair tree classifier, we see a very high variation in the performance of the model in terms of fairness while traditional performance metrics are not high enough to confidently determine that the distribution of predictions better than predicting at random. Suggesting that the model achieves its fairness performance by predicting randomly.

This shows that Demographic Parity as an performance metric might at first seem intuitive and well defined, but as with any optimisation problem, it is very hard defining good objective function that makes a model achieve what you really want. The easiest way to achieve demographic parity is to reject any model and just sample predictions at random, giving predictions that are independent of the sensitive attributes but in no sense an informed prediction.

5.3 Research Questions

5.3.1 RQ1: What probabilistic graphical model is most appropriate to model the discrimination process?

We believe that we have demonstrated the power of graphical models through their intuitive design and graphical representation of their conditional probabilities and dependencies between variables to make them fit for use for fair machine learning. Inferring latent class affiliations from datasets that are assumed to be biased is probably a quite reasonable assumption, and a necessary one to achieve fairness. Probabilistic Graphical Models may have a future as the model of choice when it comes to fair machine learning.

5.3.2 RQ2: Are the proposed models explainable?

The proposed models are not fully interpretable. The fair random forest is not interpretable and given that the model does not use the sensitive attributes in it's predictions, it is hard to infer the inner workings and decision making of the model through model-agnostic interpretability methods. The fair Bayesian network is to a more extent interpretable, dependent on the complexity of the Bayesian network. In the case that the model is an naïve Bayes classifier it is quite intuitive for a human to understand how each attribute contributes to the outcome. As the complexity of the Bayesian network increases, this becomes more difficult.

5.3.3 RQ3: Are probabilistic machine learning models cost-effecitive?

While we have not discussed this in detail, the biggest challenge to the probabilistic graphical model approach is that inference is a NP-Hard problem and requires quite a lot of computing resources. The inference is thus very slow. For some real-world cases out there this could make such models infeasible. This is the biggest drawback to these models.

5.4 Future Directions

For future work, exploring new graphical models should be encouraged. The number of possible graphical models out there is vast and given the performance these can achieve this could lead to some interesting new models. Additionally, while inference in the graphical models used here is NP-Hard, sum-product networks do not suffer with this problem having very fast inference and probabilistic calculations. Implementing fair graphical models as sum-product networks and developing libraries for this could improve the ease of implementation as well as increase the cost-effectiveness of such models.

From our results we also see that demographic parity alone as a definition of fairness can lead to models having undesired behaviour. Investigating further definitions of fairness for machine learning should be in focus. If one would find a performance metric that truly achieve fairness, a lot of models could very quickly be adapted to incorporate fairness.

We have also shown that until a state of the art fair performance metric exist, machine learning interpretability will be necessary to evaluate models in terms of fairness and to avoid accidentally deploying bad models even though they achieve good performances. Developing a framework and methods for incorporating fair machine learning methods together with machine learning interpretability should be explored further.

Appendix A

Experimental results, figures and poster

A.1 Experiment 1 Results

Accuracy	**BA**	**F1 Score**	**Specificity**	**PS Race**	**PS gender**	**Int PS**	**Model**	**Dataset**
0.42000	0.61000	0.45000	0.24000	0.88000	0.80000	0.77000	FRFC07	Adult
0.54000	0.65000	0.47000	0.44000	0.88000	0.79000	0.78000	FRFC03	Adult
0.66000	0.75000	0.57000	0.58000	0.86000	0.67000	0.71000	FRFC05	Adult
0.81000	0.80000	0.67000	0.83000	0.82000	0.66000	0.71000	NBSens	Adult
0.81000	0.80000	0.67000	0.83000	0.82000	0.66000	0.71000	NBSens	Adult
0.82000	0.76000	0.63000	0.89000	0.88000	0.94000	0.87000	FairBN	Adult
0.83000	0.72000	0.59000	0.94000	0.85000	0.75000	0.78000	IncomeBN	Adult
1.00000	1.00000	1.00000	1.00000	0.85000	0.80000	0.81000	Dataset	Adult
0.60000	0.56000	0.28000	0.96000	0.94000	0.90000	0.91000	FairBN	Compas
0.60000	0.57000	0.28000	0.96000	0.95000	0.66000	0.45000	LabelBN	Compas
0.64000	0.62000	0.52000	0.81000	0.70000	0.56000	0.38000	NBSensitive	Compas
0.64000	0.64000	0.61000	0.64000	0.91000	0.54000	0.49000	NB	Compas
1.00000	1.00000	1.00000	1.00000	0.94000	0.54000	0.50000	Dataset	Compas
0.53000	0.56000	0.64000	0.20000	0.99000	0.82000	0.82000	FRFC03	Compas
0.61000	0.59000	0.39000	0.91000	0.90000	0.66000	0.45000	FRFC05	Compas
0.46000	0.50000	0.63000	0.00000	1.00000	1.00000	1.00000	FRFC07	Compas

A.2 Poster

Fairness and Interpretability in Machine Learning Models

Introduction and Problem Statement

Problem:
Careless use of machine learning models can do more harm than good. The canonical example being COMPAS.

- Decisions might explicity depend on group membership.
- Decisions might be biased but decisions are hidden.
- Datasets are biased.

Approaches:

- Introduce new loss functions.
- Probabilistic inference.
- Explaining model decision.

Goal: Explore some fairness aware classifiers, evaluate them on datasets in fairness research and use interpretable machine learning methods to explain how the models achieve fairness.

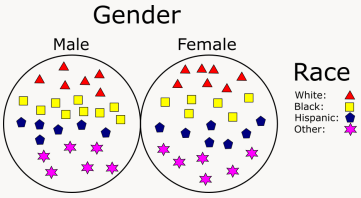
Method

The thesis has focused on implementing three models:

- Fair Bayesian Network with Latent Fair Decisions proposed by Choi et al. [1]
- Fair Tree Classifier proposed by Barata and Veenman [2]
- Naive Bayes trained with and without sensitive attributes.

Datasets:

- Adult dataset
- Compas dataset
- Synthetic dataset



Interpretable Machine Learning:

- Global Model Agnostic Methods
  - Feature Importance.
- Local Model Agnostic Methods
  - Individual Conditional Expectation Plots
  - Counterfactual Explanation Generation

Conclusion

While there exists many models out there that claim to achieve fairness when classifying individuals. This might not always be the case.

- Methods rely on predictions being independent of sensitive groups.
- Just predicting randomly achieves this.
- The model might learn a good prediction function but keeps predictions independent of the sensitive attributes (with a fairness-accuracy tradeoff)
- The might learn a prediction function that just predict randomly the outcome (leading to predictions that are independent of sensitive attributes)
- When a model is evaluated we would like to know which of the above cases are present.

How the models performed:

- Fair Bayesian Network with Latent Fair Decision achieves fairness by affirmative actions.
- The Fair Tree Classifiers achieves fairness by predicting randomly.

How to achieve true fairness:

- Explain the decision that the model makes using interpretability methods.
- Alternatively, researching loss functions for machine learning algorithms that reflect fairness.

Results

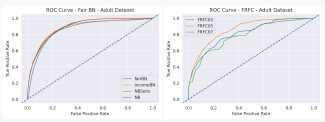


Figure: Roc-Curve of Fair Bayesian Network and Fair Tree Classifier on the Adult Dataset

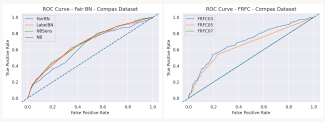


Figure: Roc-Curve of Fair Bayesian Network and Fair Tree Classifier on the Compas Dataset

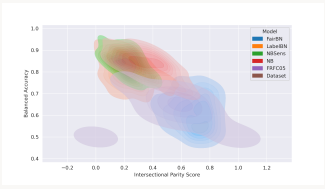


Figure: Balanced Accuracy vs Intersectional Parity Score

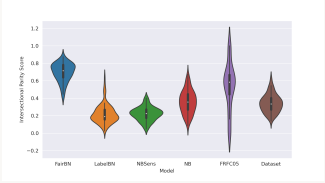


Figure: Intersectional Parity Score for models.

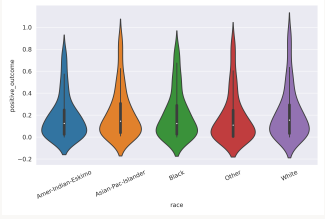


Figure: ICE Plot of Fair Bayesian Network

Naive Bayes

As a baseline method, we train Naive Bayes models both with and without the sensitive attributes in the datasets.

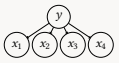


Figure: Bayesian network with 4 features representing the Naive Bayes classifier

Fair Bayesian Network with Latent Fair Decisions

Figure: Bayesian network structures that represent the proposed fair latent variable approach from [1]

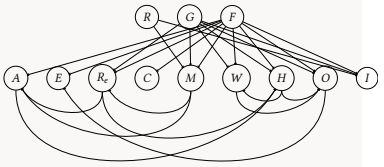


Figure: latentFairClassifier trained on Adult Dataset. The nodes are *R*: race, *G*: gender, *F*: latent fair labels, *A*: Age, *E*: Education, R_e: Relationship, *C*: Capital gain, *M*: Marital Status, *W*: Work class, *H*: Hours-per-week, *O*: Occupation and *I*: Income.

The Fair Bayesian Network proposed by Choi et al. [1] tries to simulate the discrimination process. It assumes that the labels D in the dataset are biased and generated through some distribution dependent on the sensitive attributes S and the latent true and fair labels D_f

$$P(D, D_f, S) = P(D|S, D_f)P(S)P(D_f)$$

While the non-sensitive features X are generated through some distribution

$$P(X, D_f, S) = P(X|S, D_f)P(S)P(D_f)$$

Fair Tree Classifier

The fair tree classifier proposed by Barata and Veenman [2]. They introduce a new splitting criterion that evaluates splits in terms of the Area under curve (AUC) wrt the predicted value and the sensitive attribute.
The AUC wrt the true labels Y can be calculated as

$$AUC_Y(\hat{Y}, Y) = \frac{\sum_{t_0 \in Y_-} \sum_{t_1 \in Y_+} \mathbf{1}[\hat{Y}_{t_0} < \hat{Y}_{t_1}]}{|Y_-| \cdot |Y_+|}$$

where Y_- and Y_+ are the set of indexes for negative and positive instances in the true labels. When calculating the AUC wrt the sensitive attribute, denoted AUC_S, the authors have derived the following formula

$$AUC(\hat{Y}, S) = \max(1 - AUC(\hat{Y}, S), AUC(\hat{Y}, S)) \quad (1)$$

The max operator maps the bounds to the range $[0.5, 1]$. The authors then introduce the splitting criterion used in their tree algorihm, Splitting Criterion AUC for Fairness (SCAFF). Which is calculated as

$$SCAFF(\hat{Y}, Y, S, \Theta) = (1 - \Theta) \cdot AUC_Y(\hat{Y}, Y) - \Theta \cdot AUC_S(\hat{Y}, S)$$

Where Θ is a parameter for balancing accuracy and fairness.

References

[1] YooJung Choi, Meihua Dang, and Guy Van den Broeck. Group fairness by probabilistic modeling with latent fair decisions. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(13):12051–12059, 2021.

[2] António Pereira Barata and Cor J. Veenman. Fair tree learning, 2021.

Bjørn Christian Weinbach
bc.weinbach@stud.uis.no
University of Stavanger

Appendix B

Instructions to Compile and Run System

B.1 Installation Instructions

The code used in this thesis is organised in Jupyter notebooks and it is programmed using python. A range of libraries is used and these are managed using Anaconda[1]. To make it as easy as possible to recreate the environment, we suggest that Anaconda is installed so the environment can be recreated easily. Installing anaconda is explained in detail in the documentation[2].

B.1.1 The Python Environment

It is not mandatory to install anaconda to run the code. The libraries that are necessary are as follows:

- Python
- Pytest
- Flake8 (Used for linting)
- Black (Used for linting and formatting)
- jupyter
- ipykernel

[1] https://www.anaconda.com/
[2] https://docs.anaconda.com/anaconda/install/

- pandas
- seaborn
- pip
- installed using pip: pgmpy

As long as these libraries are installed the code should work.

B.1.2 Setting up the environment using Anaconda

After installing Anaconda. The environment is set up as follows.

1. Navigate to the root folder of the code repository. There you should find a file named *environment.yml*.
2. Run the command: *conda env create -f environment.yml*
3. When the previous command is complete. Run *conda acitvate forseti* to activate the environment.

Now the environment is active and running and ready to execute the code.

B.1.3 Notebooks

We will go through each of the notebooks and their purpose so you know which notebook to use if you want to reproduce results.

- Bayesian-net-adult: Notebook used to train the fair bayesian network and naive bayes classifiers as well as saving predictions on the adult dataset.
- Counterfactuals: Runs the *generateCounterfactuals* method on the trained naive bayes and fair bayesian networks and export the results to latex
- data_exploration: Early notebook for exploring the adult dataset. Calculate correlation for dummy variables and plot the results.
- experiment2-visualisation: Generates a synthetic dataset, traind models on the synthetic dataset and visualises their scores.

- fairtree: Training the fair tree classifier on the adult dataset and COMPAS dataset and store its predictions.
- Interpretability: Training of simple interpretable models, and calculating feature importance for implemented models.
- Local-agnostic: Visualise ICE plots for implemented models.
- Model-evaluation: Calculate fairness scores for adult dataset on the implemented models.

Bibliography

[1] Manuel Castells. *The Rise of the network society Volume 1 With a new preface*. Wiley Online Library, 2009.

[2] Allam Hamdan, Aboul Ella Hassanien, Anjum Razzaque, and Bahaaeddin Alareeni. *The Fourth Industrial Revolution: Implementation of Artificial Intelligence for Growing Business Success*. Springer Nature, 2021.

[3] Chunguang Bai, Patrick Dallasega, Guido Orzes, and Joseph Sarkis. Industry 4.0 technologies assessment: A sustainability perspective. *International Journal of Production Economics*, 229:107776, 2020.

[4] The United Nations. Transforming our world: the 2030 agenda for sustainable development, 2015. UN Resolution A/RES/70/1.

[5] Ninareh Mehrabi, Fred Morstatter, Nripsuta Saxena, Kristina Lerman, and Aram Galstyan. A survey on bias and fairness in machine learning. *ACM Computing Surveys (CSUR)*, 54(6):1–35, 2021.

[6] Andrew Altman. Discrimination, 2011. URL https://plato.stanford.edu/entries/discrimination/. Accessed: 2022-02-09.

[7] Reuben Binns. Fairness in machine learning: Lessons from political philosophy. In *Proceedings of the 1st Conference on Fairness, Accountability and Transparency*, PMLR '18, pages 149–159, 2018.

[8] Julia Dressel and Hany Farid. The accuracy, fairness, and limits of predicting recidivism. *Science Advances*, 4(1):eaao5580, 2018.

[9] Toon Calders and Sicco Verwer. Three naive bayes approaches for discrimination-free classification. *Data mining and knowledge discovery*, 21(2):277–292, 2010.

[10] Toby Walsh. The troubling future for facial recognition software. *Commun. ACM*, 65:35–36, 2022.

[11] P. J. Bickel, E. A. Hammel, and J. W. O'Connell. Sex bias in graduate admissions: Data from berkeley. *Science*, 187:398–404, 1975.

[12] Jon Kleinberg, Sendhil Mullainathan, and Manish Raghavan. Inherent Trade-Offs in the Fair Determination of Risk Scores. In *8th Innovations in Theoretical Computer Science Conference (ITCS 2017)*, Leibniz International Proceedings in Informatics (LIPIcs), pages 43:1–43:23, 2017.

[13] Nripsuta Ani Saxena, Karen Huang, Evan DeFilippis, Goran Radanovic, David C. Parkes, and Yang Liu. How do fairness definitions fare? examining public attitudes towards algorithmic definitions of fairness. In *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*, AIES '19, pages 99—-106, 2019.

[14] Moritz Hardt, Eric Price, Eric Price, and Nati Srebro. Equality of opportunity in supervised learning. In *Advances in Neural Information Processing Systems*, volume 29 of *NIPS '19*, pages 3315–3323, 2016.

[15] Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Richard Zemel. Fairness through awareness. In *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference*, ITCS '12, pages 214—-226, 2012.

[16] António Pereira Barata and Cor J. Veenman. Fair tree classifier using strong demographic parity, 2021.

[17] Ray Jiang, Aldo Pacchiano, Tom Stepleton, Heinrich Jiang, and Silvia Chiappa. Wasserstein fair classification. In *Proceedings of The 35th Uncertainty in Artificial Intelligence Conference*, Proceedings of Machine Learning Research, pages 862–872, 2020.

[18] Muhammad Bilal Zafar, Isabel Valera, Manuel Rodriguez, Krishna Gummadi, and Adrian Weller. From parity to preference-based notions of fairness in classification. In *Advances in Neural Information Processing Systems*, volume 30, pages 229–239, 2017.

[19] Cynthia Dwork, Nicole Immorlica, Adam Tauman Kalai, and Max Leiserson. Decoupled classifiers for group-fair and efficient machine learning. In *Proceedings of the 1st Conference on Fairness, Accountability and Transparency*, volume 81 of *PMLR '18*, pages 119–133, 2018.

[20] YooJung Choi, Meihua Dang, and Guy Van den Broeck. Group fairness by probabilistic modeling with latent fair decisions. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(13):12051–12059, 2021.

[21] Kevin P Murphy. *Machine learning: a probabilistic perspective*. MIT press, 2012.

[22] Harry Zhang. The optimality of naive bayes. In *Proceedings of the Nineteenth National Conference on Artificial Intelligence (AAAI 2004)*, number 2 in AIII '04, page 3, 2004.

[23] Ankur Ankan and Abinash Panda. *Mastering probabilistic graphical models using python*. Packt Publishing Ltd, 2015.

[24] Daphne Koller and Nir Friedman. *Probabilistic graphical models: principles and techniques*. MIT press, 2009.

[25] Christopher M Bishop and Nasser M Nasrabadi. *Pattern recognition and machine learning*. Springer, 2006.

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

[27] Lars Buitinck, Gilles Louppe, Mathieu Blondel, Fabian Pedregosa, Andreas Mueller, Olivier Grisel, Vlad Niculae, Peter Prettenhofer, Alexandre Gramfort, Jaques Grobler, Robert Layton, Jake VanderPlas, Arnaud Joly, Brian Holt, and Gaël Varoquaux. API design for machine learning software: experiences from the scikit-learn project. In *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*, pages 108–122, 2013.

[28] Tim Miller. Explanation in artificial intelligence: Insights from the social sciences. *Artificial Intelligence*, 267:1–38, 2019.

[29] Christoph Molnar. *Interpretable machine learning*. Lulu. com, 2020.

[30] Forest Yang, Moustapha Cissé, and Sanmi Koyejo. Fairness with overlapping groups, 2020.

[31] Ankur Ankan and Abinash Panda. pgmpy: Probabilistic graphical models using python. In *Proceedings of the 14th Python in Science Conference (SCIPY 2015)*, SCIPY '15), pages 6–11, 2015.

[32] David JC MacKay, David JC Mac Kay, et al. *Information theory, inference and learning algorithms*. Cambridge university press, 2003.

[33] Frank Wilcoxon. Individual comparisons by ranking methods. *Biometrics Bulletin*, 1:80–83, 1945.

[34] C. E. Shannon. A mathematical theory of communication. *The Bell System Technical Journal*, 27(3):379–423, 1948.

[35] Susanne Dandl, Christoph Molnar, Martin Binder, and Bernd Bischl. Multi-objective counterfactual explanations. In *Parallel Problem Solving from Nature – PPSN XVI*, LNTCS, volume 12269, pages 448–469, 2020.

[36] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE Transactions on Evolutionary Computation*, 6(2): 182–197, 2002.