Mitigation of Data Bias through Fair Features Selection Methods

Ginel Dorleon

THÈSE

En vue de l'obtention du

DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par: *Université Paul Sabatier Toulouse 3*

Présentée et soutenue le *07/02/2023* **par :**

Ginel DORLEON

Atténuation des Biais de Données par des Méthodes de Sélection de Caractéristiques Équitables

---

JURY

Omar Boussaid	Pr, Université Lumière Lyon 2	Rapporteur
Nicolas Labroche	MCF, HDR, Université de Tours	Rapporteur
Imen Megdiche	MCF, INU Champollion	Examinatrice
Pascal Poncelet	Pr, Université de Montpellier	Examinateur
Nathalie Souf	MCF, Université Paul Sabatier	Co-Directrice de Thèse
Olivier Teste	Pr, Université Jean Jaurès	Directeur de Thèse

École doctorale et spécialité :
MITT : Domaine STIC : Intelligence Artificielle
Unité de Recherche :
Institut de Recherche en Informatique de Toulouse, IRIT (UMR 5505)
Directeur(s) de Thèse :
Olivier TESTE et *Nathalie SOUF*
Rapporteurs :
Omar BOUSSAID et *Nicolas LABROCHE*

Mitigation of Data Bias using Fair Feature Selection Methods

By
Ginel DORLEON

February 7, 2023

"We can't solve problems by using the same kind of thinking we used when we created them."
- Albert Einstein[a]

---

[a] The Journal of Transpersonal Psychology Transpersonal Institute, 1969, 1-4, pp 124

ACKNOWLEDGMENTS

This thesis would not have been possible without the help and support of many people. They brought me their knowledge, their encouragement, or simply their presence, which were so many opportunities to help me move forward and, sometimes, to get back on the road.

First of all, my sincere thanks to my thesis directors Nathalie SOUF and Olivier TESTE for their advice throughout this research journey; also Imen MEGDICHE who was part of the team. Your advice and comments were essential to the success of this project, thanks to you all.

Secondly, a special thanks also to the rapporteurs Omar BOUSSAID and Nicolas LABROCHE for having taken the time to review this work, to Pascal PONCELET for agreeing to be part of the jury.

In addition, a big thanks to "La Région Occitanie", "Le Syndicat Mixte" and "l'IUT de Castres" for financially supporting this research project. I would also like to thank "ISIS: École d'Ingénieurs de Castres" where I spent most of my three years of thesis.

Last but not least, I would like to thank my family and friends who always support me.

Ginel Dorleon
February, 2023

Abstract

The availability and collection of increasingly numerous and heterogeneous data, combined with the development of AI work on machine learning in big data, leads to questions about the impacts of using AI systems to support human decisions. In the context of machine learning, data is the main resource to guide decision-making. However, when bias exists in the data, this can significantly affect the decision-making process and could have far-reaching consequences. By bias we mean any systematic distortion of an evaluation or of a statistical sample chosen in a defective way.

Thus, through this thesis, our research focuses on the qualification of data and bias as well as their applications decision-making systems. **The main goal is to explore the means of informing about input data impacts on decision-making systems results by proposing a qualification of the input data importance and biases induced by the data used. The work carried out during this thesis addresses the entire decision-making process in AI with the aim of understanding the different sources of bias, detecting them and mitigating their effects on the results produced for specific applications.**

During this thesis, we have developed, through several contributions, approaches and methods that make it possible to identify, correct biases and improve fairness in decision-making systems. First, we established a taxonomy of biases and the area where they are likely to occur in the data science process. This first work allowed us to reveal in a second step how feature selection can contribute to induce biases in the decision-making process when features considered to be sensitive (protected) or their redundant are selected. Based on that, we have proposed a first method which consists in evaluating redundancy between features to avoid, in search of fairness, deleting too many features, which would lead to a significant performance loss. Then, we proposed a second approach where we avoid using protected features in the decision-making model but instead their redundant by establishing a trade-off strategy between the model's performance and its fairness. And finally, to compensate the limits of this previous approach on the fact that it did not use any protected features, we opted for an approach of group-balancing and non-deletion of data. Through this last approach, we have proposed a method which aims to divide the input data into subgroups which later will be balanced with regard to the protected features. Then fair local decision-making models are built on these balanced subgroups. Finally, using a learning ensemble strategy, we obtain a final model that is fair without removing any protected features.

We have evaluated and experimentally validated the effectiveness of each of these contributions, which have proven to be very relevant in view of our issue of bias.

Keywords: Artificial intelligence; machine learning; bias; feature selection; decision-making system; protected feature; imbalanced data; redundant feature; fairness

Résumé

La disponibilité et le recueil de données qui sont de plus en plus nombreuses mais hétérogènes, alliés au développement de travaux en intelligence artificielle (IA) basée sur l'apprentissage machine, conduit à se questionner sur les impacts de l'utilisation de ces systèmes d'IA pour accompagner des décisions humaines. Dans le contexte de l'apprentissage machine, les données constituent la principale ressource pour guider les prises de décisions. Cependant, lorsque des biais existent dans les données, cela peut affecter de façon significative l'interprétation des décisions et pourrait avoir des conséquences considérables. Par biais, nous entendons toute déformation systématique d'une évaluation ou d'un échantillon statistique choisi de manière défectueuse.

Ainsi, à travers cette thèse, notre recherche se base sur la qualification des données et des biais ainsi que leurs applications dans les systèmes décisionnels. **L'objectif est d'étudier les moyens d'informer sur les impacts des données d'entrées sur les résultats des systèmes décisionnels en proposant une qualification de l'importance des données et de leur biais. Le travail réalisé au cours de cette thèse aborde l'ensemble du processus décisionnel en IA dans le but de comprendre les différentes sources de biais, de les détecter et d'atténuer leurs effets sur les résultats produits pour des applications spécifiques.**

Au cours de la thèse, nous avons élaboré des approches et des méthodes qui permettent d'identifier, corriger les biais et d'améliorer l'équité dans les systèmes décisionnels. Nous avons établi en premier lieu une taxonomie de biais ainsi que les étapes où ils sont susceptibles d'arriver lors du développement d'un système d'IA dirigé par les données. Ce premier travail nous a permis par la suite de montrer comment la sélection des caractéristiques d'entrées peut induire des biais lorsque des attributs jugés protégés (ou leurs redondants) sont sélectionnés. Nous avons proposé une première méthode qui consiste à évaluer la redondance entre les caractéristiques pour éviter, à trop vouloir être équitable, de supprimer trop d'attributs, ce qui conduirait à une perte considérable en terme de performance. Dans un second travail, nous avons proposé une approche qui vise la non-prise en compte des attributs protégés dans la construction du modèle décisionnel mais plutôt leurs redondants par un compromis entre la performance et l'équité. Et finalement, pour compenser les limites de cette approche sur la non-prise en compte des attributs protégés dans la construction du modèle, nous avons opté pour une approche d'équilibrage et de non-suppression de données. A travers cette dernière approche, nous avons proposé une méthode qui vise à découper les données d'entrées en sous-groupes ("clusters") qui sont équilibrés au regard des attributs protégés. Ensuite des modèles décisionnels locaux équitables sont construits sur ces sous-groupes. Puis à l'aide d'une stratégie ensembliste, un modèle global équitable est obtenu en conservant les attributs protégés.

Nous avons évalué et validé expérimentalement l'efficacité de chacune de ces contributions qui se sont avérées très pertinentes au vu de notre problématique sur les

biais.

Mots-clés: Intelligence artificielle; apprentissage automatique; biais; sélection de caractéristiques; système décisionnel; attribut protégé; déséquilibre des données; attribut redondant; équité.

Table of contents

List of Figures

List of Tables

Chapter 1

Introduction

Contents

1.1 Context and Motivation of the Thesis

The accessibility of numerous and heterogeneous data, coupled with the development of work in artificial intelligence (AI) and machine learning on big data, has raised concerns about the impacts using AI systems to support human decisions. These concerns are clearly exposed in the Villany report [1] which highlights the need for transparency in autonomous systems. This need for transparency is further carried by he DARFA (Defence Advanced Research Project Agency) which later defines the notion of Explainable Artificial Intelligence (XAI) [2] to introduce explainability into decision-making systems [Younsi *et al.*, 2019]. In the context of health or justice applications for example, the notion of explainability is particularly significant for offering systems that people can put their trust. The need for explainability was identified from the first applications of AI [Shortliffe *et al.*, 1979] however, with the current trends where big data abounds and for which a plethora of black box-type solutions are used, this need is crucially amplified.

Machine learning methods are based either on explicit models which in general make it possible to obtain an explicable result (such as decision trees), or on black box-type models (such as deep neural networks) which proof of their effectiveness in terms of results but hiding the reason for obtaining such results. However, explaining, interpreting or understanding the decision-making mechanisms are unavoidable challenges to integrate such tools in decision-making systems of critical area such as justice, finance as exposed by [Samek et Müller, 2019] and [Gunning et Aha, 2019], on the needs for an explainable Artificial Intelligence.

Like many other scientific fields, data processing has greatly benefited from the advancement of artificial intelligence and machine learning. The increase in the amount of data produced on a daily basis plays an important role in data science work. With artificial intelligence and machine learning, we have witnessed interesting advances in many areas such as image recognition, prediction systems, recommendation systems, targeted marketing, etc. The basis of these advances are implemented by algorithms performing calculation beyond humans [Cortes *et al.*, 2001]. Thus, algorithms are increasingly part of our daily lives. However, despite of how efficient and intelligent they may be, these algorithms only analyze data that is provided. These algorithms are criticized for being black boxes and sometimes for producing controversial results qualified as biased [Breiman et Wald Lecture, 2002, Ye *et al.*, 2021, Cao *et al.*, 2021]. Algorithms produce outputs from what have been provided to them as inputs. Biases are likely to be linked to the inputs, hence the importance of qualifying and identifying which input features to be considered. An importance that is discussed throughout the first part of our work.

In this thesis, we are mainly focused on two issues related to the problematic of the subject:

[1] https://www.aiforhumanity.fr/pdfs/9782111457089_Rapport_Villani_accessible.pdf

[2] https://www.darpa.mil/program/explainable-artificial-intelligence

- **a) Tackling inputs feature importance and bias related.**
  For the first issue of the thesis, we are interested in how bias occurs in machine learning algorithms. For this, we investigate and qualify the characteristics of the given inputs. We do so by addressing the notion of feature selection.

- **b) Mitigating bias in machine learning models.**
  For the second issue of the thesis, we are interested in mitigating bias in machine learning algorithms. To do this, we look at the main causes of bias that we have identified in the first issue in terms of imbalanced data and protected features.

We extend our work to the notion of class imbalance and their impact on an output model. We do this by looking at the effect of different classes' imbalance ratio and the impact on the obtained model.

1.2 Contributions Overview

Throughout the course of this PhD, our research has focused the entire AI decision-making process with the aim of understanding how and where biases occur, how to prevent them or mitigate their effects on specific applications. The main contributions of this thesis that we later detail in the individual chapters have been published in different international venues. We summarize the contributions of our thesis in three parts detailed below:

1. During the first part of the thesis, we have published a taxonomy and definition of biases using a data-science process in which we have identified the different areas where different types of biases are likely to occur. This contribution, named as **Qualification of data bias in the data science process**, has been published in EGC'21[Dorleon *et al.*, 2021b].

2. In the second part of the thesis, the taxonomy defined in part 1 allowed us to identify and focus on a specific type of bias: the one related to feature selection. Thus, we have proposed an approach consisting of removing protected features while using their redundant. This method named as **Features Selection Under Fairness Constraints** is a trade-off between fairness and performance. A poster of this contribution has been published in the 37th ACM/SIGAPP Symposium On Applied Computing [Dorleon *et al.*, 2022b]. The full paper has been published in the 24th International Conference on Big Data Analytics and Knowledge Discovery (DaWaK) [Dorleon *et al.*, 2022a]. Beside this, we have also proposed a redundancy analysis method that aims to handle redundancy when dealing with protected features. This redundancy based method, **Absolute Redundancy Analysis Based on Features Selection**, has been published in the 4th International Conference on

Data Mining and Big Data (DMBD 2021), Shangai, China ACM, New York, NY, USA [Dorleon *et al.*, 2021a].

3. In the third part of the thesis, we have proposed another approach which attempts to overcome limits of our second approach by not deleting data relating to protected attributes. Thus, this method named **FAPFID: A Fairness-aware Approach for Protected Features and Imbalanced Data** is a fairness-aware strategy based on the use of balanced and stable clusters for dealing with protected features and data imbalance. It has been accepted in TLDKS (Transactions on Large-Scale Data and Knowledge-Centered Systems) 2023.

1.3 Outline of the Thesis

The rest of the manuscript is organized as follows: in chapter 2, we provide a wide state-of-the-art of our main research interests in terms of bias, fairness and feature selection. Furthermore, we investigate how bias occurs in AI decision-system, the different causes and sources in terms of protected/redundant features and imbalanced data. We also give more details on all fairness metrics that we have used in our experiments. In chapters 3, 4 and 5, we present our main contributions, their experimental process and results.

All of the work detailed in these three chapters are contained in the following publications:

1. Ginel Dorleon and Nathalie Bricon-Souf and Imen Megdiche and Olivier Teste Qualification du biais de données dans le processus de la science des données, Revue des Nouvelles Technologies de l'Information EGC'21 [Dorleon *et al.*, 2021b].

2. Ginel Dorleon, Nathalie Bricon-Souf, Imen Megdiche, and Olivier Teste Absolute Redundancy Analysis Based on Features Selection. In DSIT '21: 4th International Conference on Data Mining and Big Data (DMBD 2021), Shangai, China. ACM, New York, USA [Dorleon *et al.*, 2021a].

3. Ginel Dorleon, Imen Megdiche, Nathalie Bricon-Souf, and Olivier Teste. 2022 Feature Selection Under Fairness Constraints [POSTER]. In Proceedings of ACM SAC Conference (SAC'22), ACM, New York, NY, USA [Dorleon *et al.*, 2022b].

4. Ginel Dorleon, Imen Megdiche, Nathalie Bricon-Souf, and Olivier Teste. 2022 Feature Selection Under Fairness and Performance Constraints. The 24th International Conference on Big Data Analytics and Knowledge Discovery (DaWaK) Vienna, Austria [Dorleon *et al.*, 2022a].

5. Ginel Dorleon, Imen Megdiche, Nathalie Bricon-Souf, and Olivier Teste. 2022 FAPFID: A Fairness-aware Approach for Protected Feature and Imbalanced Data [Accepted in TLDKS].

In chapter 6, we present a summary of our work, conclusions and future perspectives. In the annex A, we introduce all the necessary basics settings and background on machine learning techniques and algorithms that are used in different chapters of this manuscript. This chapter aims to give more explanations and facilitate the understanding of the basics of machine learning algorithms used in this thesis.

Chapter 2

SOTA: Bias, Fairness and Feature Selection

Contents

2.1 Introduction

Current machine learning approaches are used to support decision-making process by enabling the discovery and prediction of complex hidden scheme in big data. These approaches are used in the decision-making process of many applications such as transportation [Paparrizos *et al.*, 2011], healthcare [Farahani *et al.*, 2019], recruitment or employment screening [Qin et Tang, 2019], finance [Amarasinghe *et al.*, 2018], engineering [Lejeune *et al.*, 2020], business intelligence recommendation [Drushku *et al.*, 2019], news feed updates prediction [Belkacem *et al.*, 2020] and many more. Any error made during the decision-making process can cause significant damage to the organization and more generally to human life [Osoba et IV, 2017]. It is a common belief that using an automated algorithm makes decisions more objective [Baeza-Yates, 2018]. However, this is unfortunately not the case since artificial intelligence (AI) algorithms are not always as objective as we would expect to have a reliable model.

To be used in real-world, decision-making systems are evaluated based on their performance which is mostly depends on data. Whether it is guiding businesses decisions, offering new services using machine learning algorithms, data is the primary resource for improving decision-making's performance and is being used in all of the stages of the process. Consequently, any gap in the data or in its use during any stage of the decision-making process may result in significant performance losses at an economic and human level. One among the errors that can hinder obtaining reliable decision-making systems is the use of biased data, i.e data containing bias [Pessach et Shmueli, 2020, Baeza-Yates, 2018]. In the context of decision-making systems, bias is referred to the problems related to the gathering or processing of data that might result in prejudiced decisions on the bases of inherent or acquired characteristics such as race, sex, and so forth or the use of dataset with an imbalance between classes [Ntoutsi *et al.*, 2020]. When undetected, bias in data can significantly affect the interpretation of results and have devastating consequences on the use of AI in areas such as justice or health [Agarwal *et al.*, 2019, Osoba et IV, 2017].

Recently, it as been discovered that machine learning algorithms [Yeom *et al.*, 2018] may lead to unfair decisions against certain groups defined by these inherent or acquired characteristics. Thus, fairness [Barocas *et al.*, 2017, Oneto et Chiappa, 2020] is another concern in using automated decision-making based on machine learning algorithms. In the context of decision-making, fairness is the absence of any discrimination or favoritism towards an individual or a group based on their inherent or acquired characteristics [Mitchell *et al.*, 2021]. Since the use of these systems now affects many aspects of people's life, it is crucial to focus on the development of decision models that can help mitigating bias in data [Chouldechova *et al.*, 2018] and guarantee fairness in automated decision-making based on machine learning algorithms.

The use of these inherent or acquired characteristics in decision-making systems based on machine learning can be amplified by the application of certain feature selection

methods [Jensen et Neville, 2002]. Feature selection methods aim to select a set of relevant features for a learning task. However, in some cases where inherent or acquired characteristics referred to as protected features are selected, this can significantly leads to bias in the decision-making [Singhi et Liu, 2006].

In the sections of this chapter, we will address the issues of Bias, Fairness and Feature Selection. The rest of the chapter is organized as follows: in section 2.2 we present a state-of-the-art (sota) of the different categories and types of biases that have been mentioned in the literature, with some real-world examples of known biased systems. In section 2.3, we introduce a general data science process in which we identify areas on the process where different types of bias are likely to occur. In section 2.4, we present a state-of-the-art of fairness and the related literature of the most frequently used fairness metrics and processing techniques that are used in the literature to mitigate bias. In section 2.5 we review the different methods of feature selection whose applications can also lead to bias.

2.2 Bias

One of the main causes of poor decision-making in automated machine learning systems is what we most often referred to as ***bias*** [Baeza-Yates, 2018]. Machine learning is used in many real life applications [Chouldechova *et al.*, 2018, Guégan et Hassani, 2018] such as courts to assess the likelihood of a recidivism, in different medical fields, in child protection systems and in autonomous vehicles. All of these applications have a direct effect on people's lives and can harm society if not modeled and designed properly between different groups with different characteristics. Thus, a biased model is one whose decisions are biased in favor of a particular group.

2.2.1 Bias Definition

In machine learning, the term ***bias*** was introduced by [Mitchell, 1980] to mean any basis for choosing one hypothesis over another, other than strict consistency with the observed training instances. Authors in [Baeza-Yates, 2018] have defined bias as the interference in the research results of any fault whose presence can distort the results and their interpretation or influence them in a certain direction. Specifically related to decision-making systems, [Ntoutsi *et al.*, 2020] defined bias as the problem related to the gathering or processing of data that might result in prejudiced decisions on the bases of demographic features such as race, sex, and so forth.

Below, we present some examples of world-known systems where biases were detected. Those systems were biased mostly towards certain groups on the bases of inherent or demographic features.

- The Amazon Prime Bias: That Amazon's prime AI algorithm's task was to decide which areas of a city in the United Sates (US) is eligible for receiving advanced services. However, [Ingold et Soper, 2016] found that areas mostly inhabited by black people were ignored even though the algorithm did not consider race as a feature.

- The Compass Software: Compass is software used in the criminal justice system in the US. However, recent findings have shown that the algorithm used in that software incorrectly predicted future crime among African Americans at twice the rate it predicted for whites [Amini *et al.*, 2019, Angwin et Kirchner, 2016].

- The Amazon Hiring System: In 2014, it was revealed that Amazon discovered that its hiring system discriminates against female applicants, especially for software development and technical positions [Wang et Wang, 2014]. It is suspected that the reason is that most of the historical data recorded was of software developers who were male [Backurs *et al.*, 2019].

- The Google's Ads Algorithm: [Datta *et al.*, 2014] and [Baeza-Yates, 2018] pointed out what is referred to as gender-bias in the Google's ads targeting algorithm. In fact, they discovered that this algorithm has been shown to offer higher-paying senior executive jobs to men than women.

- The Biased Face Recognition Camera: [Alipourfard *et al.*, 2018] identify a facial recognition software in digital cameras that overestimates Asians blinking.

- The AI Beauty System: Another example of biased AI is that system [Lloyd, 2018] that judges beauty pageant winners but biased against darker-skinned contestants [Osoba et IV, 2017]. When the results where released, out of the 44 people that the algorithms judged to be the most "attractive," all of the finalists were white except for six who were Asian. Only one finalist had visibly dark skin.

2.2.2 Bias Categories

Several categories of bias have been mentioned in the literature by [Bellamy *et al.*, 2018], [Dobbe *et al.*, 2018], [Amini *et al.*, 2019] and also by [Osoba et IV, 2017]. Specially, three categories of bias have been presented by [Barocas et Selbst, 2016]:

1. **Humans Bias:** systematic error in thinking that occurs when people are processing and interpreting information in the world around them and affects the decisions and judgments that they make. In other words, it is a systematic pattern of deviation from norm and/or rationality in judgment.

2. **Algorithm Bias:** this type of bias refers to the bias that is not present in the input data but added, created or generated purely by the algorithm [Baeza-Yates, 2018]. The algorithmic design choices, such as use of certain optimization functions, regularization, choices in applying regression models on the data as a whole or considering subgroups can lead to algorithmic bias [Danks et London, 2017].

3. **Data Bias:** according to [Dee, 2005], bias in data is an error that occurs when certain elements of a dataset are over-weighted or over-represented. Biased datasets don't accurately represent a model's use case, which leads to skewed outcomes or systematic prejudice. According to [Lavalle *et al.*, 2020], bias in data most often results from imbalance of classes.

Later on, this categorization made by [Barocas et Selbst, 2016] has been quoted by [Mehrabi *et al.*, 2021] who have identified two categories of bias:

1. Bias originating from data referred to as **Data Bias**

2. and those originating from algorithms mainly referred to as **Algorithm Bias**.

In this thesis, we were based on the study of bias by [Mehrabi *et al.*, 2021] instead of the categorization made by [Barocas et Selbst, 2016]. The reason is that the former one demonstrated how human bias is considered as a type of data bias. Thus, we were focused on data bias as data represents the starting point of a any decision problem based on machine learning and artificial intelligence.

2.2.3 Data Bias

Data bias is one of the primary causes of unreliable outcomes from automated decision-making system based on machine learning and big data [Žliobaitė, 2017]. Bias can exist in different types and may have unreliable sources. Different types of data bias have been mentioned in the literature, the study in [Mehrabi *et al.*, 2021] highlights the more important ones. In [Suresh et Guttag, 2019], the authors outline a list of different types of biases with their corresponding definitions that exist in different cycles of a data processing.

In the following , we outline the most relevant types of data bias based on the work of [Suresh et Guttag, 2019] and [Mehrabi *et al.*, 2021]:

1. **Historical Bias.** Historical bias is the socio-technical problems that often exist in the environment where the data is collected and which can infiltrate the data generation process [Mehrabi *et al.*, 2021].

   An example of this type of bias can be found in a 2018 image search result on Google where searching for women CEOs[1] ultimately resulted in fewer female CEO images due to the fact that only 5% of Fortune 500 CEOs were

[1] CEO: Chief Executive Officer

woman, which would cause the search results to be biased towards male CEOs [Suresh et Guttag, 2019]. These search results were of course reflecting the reality, but whether or not the search algorithms should reflect this reality is an issue worth considering.

2. **Sampling Bias.** According to [Mehrabi *et al.*, 2021] is due to a non-random sampling of the subgroups. As a result of the sampling bias, the trends estimated for a population may not be generalized to the data collected from a new population.

   To explain this type of bias, let's consider a school with four grades. If we sample the students from this school by using only students from one grade and ignore the others, we will experience what is referred as *sampling bias*.

3. **Representation Bias.** Representation bias, by [Friedman et Nissenbaum, 1996], occurs from the criteria used to sample a population or when certain characteristics of the population are underrepresented.
   Considering the above example in sampling bias, *representation bias* will arrive if, among of the selected sample, one considers only male students over female or vice versa.

4. **Measurement Bias.** According to [Mehrabi *et al.*, 2021], measurement bias comes from the way we choose and measure a particular feature.
   An example of this type of bias was observed in the COMPAS recidivism risk prediction algorithm [Angwin et Kirchner, 2016]. In this algorithm, previous arrests and arrests of friends/families were used as proxy[2] feature to measure the level of risk or criminality by race. Thus, the algorithm was still biased against race because the proxy feature has provided the same amount information as race. This type of bias is also referred as feature selection bias.

5. **Omitted Variable Bias.** Omitted feature bias, as mentioned by [Mehrabi *et al.*, 2021] and [Friedman et Nissenbaum, 1996], is the type of bias that occurs when one or more important features are omitted during the model construction phase. This type of bias is likely a result of a technical default.

6. **Aggregation Bias.** This bias appears when a single model is used for groups with different conditional distributions. Aggregation bias can lead to a model that is not optimal for any group, or a model adapted to the dominant population (if combined with a representation bias) [Mehrabi *et al.*, 2021]. For example, a model trained on a population $H1$ should not be used for a a population $H2$ as they may present different conditional distributions.

[2]By proxy feature we mean a feature by which we can find information about another one, for example age and date of birth.

7. **Evaluation Bias.** Evaluation bias according to [Friedman et Nissenbaum, 1996] occurs when the evaluation and/or the reference data used to evaluate a model does not represent the target population. A model is optimized on its training data, but its quality is often measured on benchmarks. For example, benchmarks datasets such as the UCI[3] data sets [Huang *et al.*, 2008], Faces in the Wild [Wan *et al.*, 2020], ImageNet [Deng *et al.*, 2009] are often used to evaluate face recognition models. However, it can not be certain that the obtained performance is reliable enough for deployment.

8. **Deployment Bias.** This type of bias arises when there is a mismatch between the problem that a model is designed to solve and the way it is actually used. This often happens when a system is built and evaluated as if it were entirely autonomous, when in reality it operates in a complex socio-technical system moderated by institutional structures and human decision-makers [Barocas et Selbst, 2016]. In other words, a model is built for a particular task; if this is not the task actually accomplished after deployment, there is no guarantee that a good evaluation performance will be maintained. Likewise, a model trained and used on patients with skin cancer in Europe is not certain to be effective on patients from another continent such as Asia for example.

2.3 Qualification of data bias in the data science process

In this section we present a qualification of data bias in the data science process. This work constitutes our first contribution [Dorleon *et al.*, 2021b]. The goal is to alert readers so that they can be aware of the areas where bias can occur in the data science process when they are about to develop a machine-learning data-based-model. Development of such systems requires a deep understanding on why and where biases arise in a data science process.

2.3.1 Presentation of the Data Science Process

The classic data science process considered (Fig. 2.1) has tree stages: data collection, data preparation and model usage.

— **Data Collection** is the first stage of the data science process. It contains actions devoted to gather information from the actual world, extracting significant sample of a population. We notice three actions in this stage: Data generation, selection of a population and sample selection.

[3]University of California Irvine (https://archive.ics.uci.edu/ml/index.php)

— **Data Preparation** is the second stage of the data science process, it contains actions which help to clean the data, measure and select feature to build the training and the test datasets.

— **Model Usage** is the last stage and gathers all actions linked to built, deploy and evaluate the model.

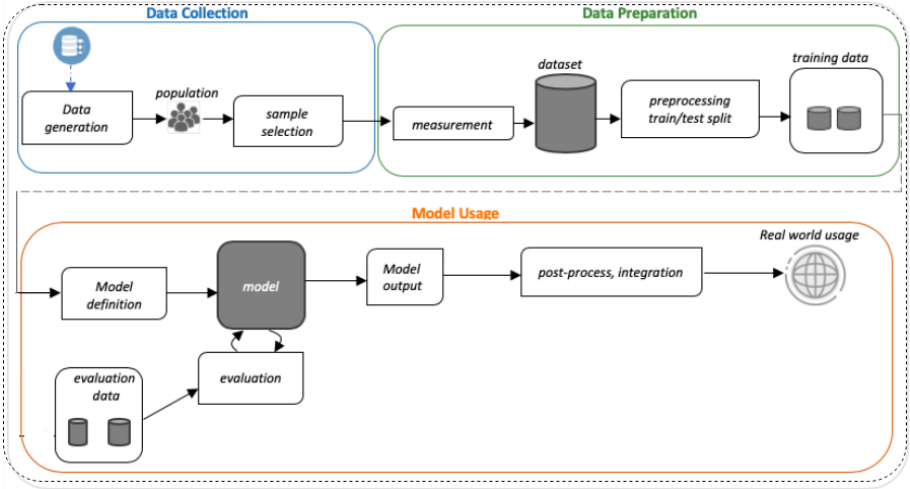

Figure 2.1: The three stages of the data science process

Below, we present a simple formalization of the settings used for describing the data science process. The general data science process considered begins with collecting data from an operational environment, the collected data helps to build a population. The input data of the process can be modeled as a set of data annotated $X_{n,m} \in R^{n \times m}$ (with no added constraints on data type) consisting of n independents distributed samples and m non-independents features with $F = \{F_1, ..., F_m\}$ being the feature space. Using a matrix, we can then write our input data as:

$$X_{n,m} = \begin{pmatrix} x_{1,1} & \cdots & x_{1,m} \\ x_{2,1} & \cdots & x_{2,m} \\ \vdots & \ddots & \vdots \\ x_{n,1} & \cdots & x_{n,m} \end{pmatrix} = (x_{i,j})_{1 \leq i \leq n, 1 \leq j \leq m}$$

Each variable of $X_{n,m}$ is represented by a column vector annotated

$$x_{\bullet j} = (x_{i,j})_{1 \leq i \leq n} \in R^n$$

and each individual of $X_{n,m}$ is represented by a line vector annotated

$$x_{i\bullet} = (x_{i,j})_{1 \leq j \leq m} \in R^m$$

In the data collection stage, the goal is to select a set of data by sampling a population from the data generated. To do so, we introduce the following parameters: let $X' \subseteq X$ be a sample, $S : X \to X'$ a sampling function and P be a distribution probability of all the samples $X' \subset X$ identically distributed such that $P(X') \geq 0$ and $\sum_{X' \subset X} P(X')$=1.

In the data preparation stage, measurement techniques such as feature selection or other dimensionality reduction methods are used; the initial dataset $X_{n,m} \in R^{n \times m}$ is then redefined through a reduced features set. Let w be a feature selection function over $x_{\bullet j}$ such that $w : x_{\bullet j} \to x'_{\bullet j}$ and h a dimensionality reduction function as described by [Khalid *et al.*, 2014] over $x_{\bullet j}$ such that $h : x_{\bullet j} \to x'_{\bullet j}$.

The dataset is then divided into training and evaluation sets which are used to develop and evaluate a desired model. The generated model produces an output according to the objective for a real-world application. This steps represents the model usage stage in the process. At this stage, a learning model is chosen and configured to learn with the training dataset. This learning process can be described by $\big(X_{n,m}, Y_{n,1}\big)$ where $X_{n,m} \in R^{n \times m}$, and $Y_{n,1} \in R^n$, which is a learning model learned using $Y_{n,1} = f\big(X_{n,m}\big) + \epsilon_{n+1}$ with $\epsilon_{n+1} \in R^n$ representing the noise in the data. The learning process then consists in estimating the function f by a learned function $\hat{f}$ in order to generate a model noted z.

The different steps of the data science process could obviously induce different kinds of bias. In section 2.2.3, we benefit from different taxonomies of bias mentioned in the literature. Our purpose is then to identify different areas where bias can occur in the process.

2.3.2 Data Bias: Qualification and Impacts

To clarify our presentation of bias, we identify for each stage section which kind of bias that could be encountered in the process. We first identify areas where in the process bias can occur and then recall about risks and impacts on the process induced by such bias.

2.3.2.1 Data Collection

Results, analysis, evaluations and conclusions depend on the data collected and then on this stage. The appearance of bias may be the result of several anomalies. Either generated data are not representative of the chosen population, or they reflect existing prejudices. The data collection process can also generate biases according to several other practices such as : the use of survey questions constructed with a particular inclination, data transfer into non-related categories or non-random sampling between groups.

In Fig. 2.2, we illustrate the data collection stage of the process and indicate areas where each type of biases is likely to occur. We then qualify each type of these bias and point out their impacts on the decision of the learning process. At the stage of data collection in the data science process, we highlight three type of bias:

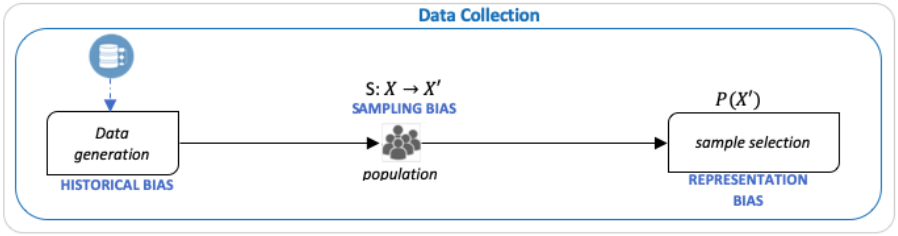

Figure 2.2: Data collection stage. P is a probability distribution over each sample, S is a sampling function.

- ***Historical bias*** occurs in the data generation step. This type of bias is likely to arise depending on how the dataset $X_{n,m}$ is generated. If the generated dataset contains n samples $x_{i\bullet} = (x_{i,j})_{1 \leq i \leq m}$ predefined, unverified and accepted in the dataset, these samples will lead to an historical bias during the learning process by a model which will seek to reflect the existing reality of the dataset.

- ***Sampling bias*** impacts the sampled population. It occurs if the samples annotated X' from X formed by the sampling function S are generated in a non-random way. In other words, samplis bias occurs if the algorithm used by the function S did not randomly generate the X' samples and the trends estimated for a sample X' cannot be generalized to new samples of $X_{n,m}$.

- ***Representation bias*** occurs during the population distribution. There will be *representation bias* if the probability distribution P samples too few samples $x_{i\bullet} \in X'$ and if certain samples $x_{i\bullet} \in X$ are underrepresented in the X' sample.

In Table 2.1, we notice the impacts that different types of bias of the data collection stage may have on the output of the process.

Table 2.1: **Data collection: potential cause and impacts on the process**

Data Collection		
Potential Cause	**Bias**	**Impact on the process**
Bias can be introduced by anyone involved in the data collection process. This can happen during sampling or during group building, as well as during data submission and collection.	Historical Bias	The algorithm will try to reflect the existing trends in the data.
	Sampling Bias	Erroneous deductions because certain subgroups may not exist in sufficient number for a learning algorithm.
	Representation Bias	Limitation of the decision to the most represented groups.

2.3.2.2 Data Preparation

Data preparation is the second stage of the process. Different types of measures and processing methods can be applied to the dataset before using it. Training and test data

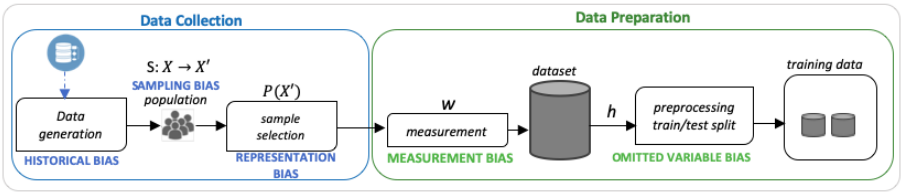

Figure 2.3: Data preparation stage, w represents a feature selection function and h is a dimensionality reduction function.

are created at this stage. However, it is possible to generate biases during this stage which includes the selection and measurement of features that the algorithm will use to build the model. We noticed two types of bias that can arise at this stage and that are illustrated in Fig 2.3:

- ***Measurement bias*** is likely to occur in the data preparation stage. It impacts the feature selection and dimensionality step in the data preparation stage. It will arise from the different techniques used by w, the feature selection function and h, the dimensionality reduction function, to measure and select features f_j that the algorithm will consider for training a model.

- ***Omitted feature bias*** occurs if important features f_j defined by h are not considered or missed when learning the function $\hat{f}$.

The table 2.2 below provides a summary of this stage and the biases that are likely to be generated.

Table 2.2: **Data preparation: potential cause and impacts on the process**

Data Preparation		
Potential Cause	**Bias**	**Impact on the process**
Using incorrect criteria to select the attributes that the algorithm will use can lead to bias.	Measurement Bias	Erroneous deductions because some measures may be inconsistent between groups.
	Omitted Variable Bias	Removing important features by using inconsistent measures will generate a model built on insignificant data.

2.3.2.3 Model Usage : Learning, Evaluation and Deployment

Model usage is the last stage of the process. This is the stage where we build, train, evaluate and deploy the model using the training data created in the data preparation stage. After training the model, test data is used to report the performance of the output model. In addition to the test data, other available datasets - also called reference datasets - can be used to demonstrate the robustness of the model or to allow comparison with different existing methods. Therefore, it is important to choose a well-adapted performance measures in other to avoid biased conclusion.

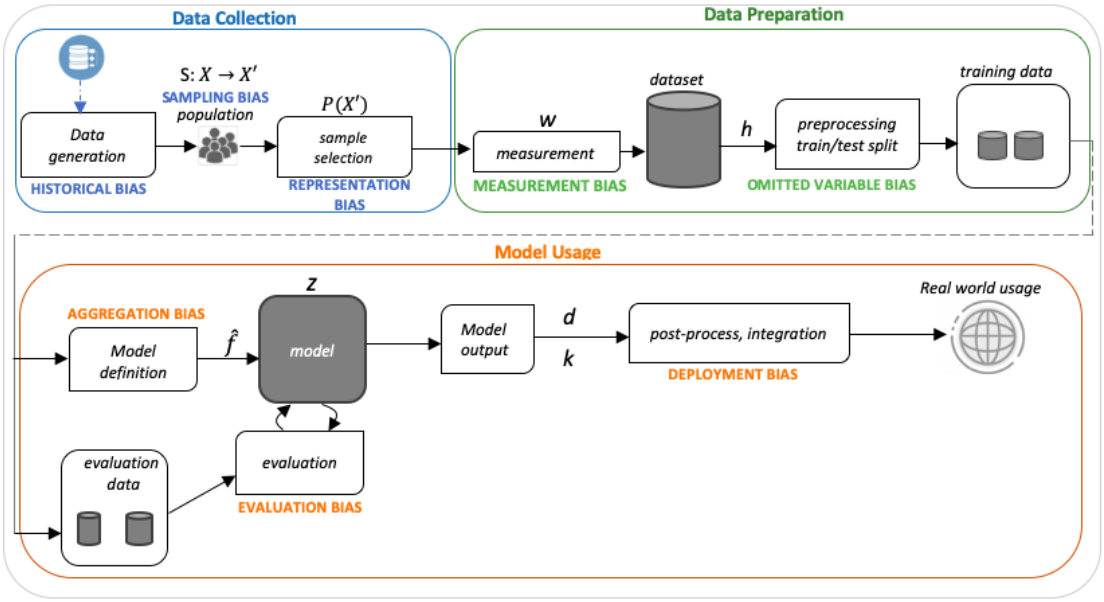

Figure 2.4: Final stage in the process: Model Usage. $\hat{f}$ is the learned function, z is the output model, d is the final decision and k an external factor (may be human) that has a potential influence on the final decision d.

We noticed three types of bias at this stage as illustrated in Fig 2.4, a summary is presented in Table 2.3.

- ***Aggregation bias*** occurs in the population where the model is used. It will occur if decisions are taken by the learned model $\hat{f}$ for a $X_{n,m}$ sub-population with different distributions.
- ***Evaluation bias*** occurs in the model's evaluation. It will occur if a new dataset with different modality from $X_{n,m}$ is used to evaluate the learned model $\hat{f}$.
- ***Deployment bias*** occurs when k, the external factor potentially a human, does not simply reproduce the predictions made by z but introduces unexpected behavior affecting the final decision d.

Table 2.3: **Model usage: potential cause and impacts on the process**

Model Usage		
Potential Cause	**Bias**	**Impact on the process**
Evaluation criteria that do not correspond to the studied population or a wrong combination of distinct populations.	Aggregation Bias	A single model cannot be effective on a heterogeneous population.
	Evaluation Bias	Using evaluation criteria that are not appropriate leads to a false model's evaluation.
	Deployment Bias	Using the model in a different context from the primary objective may harm not because the model is defective but because the context is different.

2.3.3 Bias Mitigation Methods

To reduce or mitigate bias in data, some general methods referred to as "Bias Mitigation Methods" have been proposed. These methods aim to support having good practices while working with data, from processing the data to the selection of which features that should be used to train the model.

- **Data Collection**
  For representation and sampling issues, existing solutions most often analyze them as problems linked to imbalanced data. The proposed methods are mostly focused on oversampling the data using different techniques such as SMOTE by [Chawla *et al.*, 2002] and ADASYN by [He *et al.*, 2008].

- **Data Preparation**
  For bias related to feature selection and dimensionality reduction methods, authors in [Cascaro *et al.*, 2019] and [Mostert *et al.*, 2018] have suggested to use different techniques such as embedded, filter and wrapper methods. In [Oneto *et al.*, 2019], authors have proposed a learning function that consider groups difference, this method can help reducing aggregation bias by taking advantage of multitask learning. Finally, authors in [Buolamwini et Gebru, 2018] have proposed to use per-groups metrics in order to create subgroup to evaluate a model and avoid evaluation bias.

- **Model Usage**
  For bias related to the model usage, authors in [Calmon *et al.*, 2017], have introduced a novel probabilistic formulation. It consists in three approaches: i) controlling discrimination, ii) limiting distortion in individual data samples and iii) preserving utility in order to ensure an unbiased learning. Other preliminary mechanisms,

such as those proposed by [Kamiran et Calders, 2011] and [Luong *et al.*, 2011], have suggested to relabel examples or readjust their weight before the training stage in order to have a better output.

2.3.4 Summary

Data bias is a major problem due to its impact on data analysis and decision-making systems. Being able to define and qualify bias in the data is an important step in the process of reducing these harmful impacts. Current solutions focus on the problem from the view of AI results, but tackling the process where these biases are generated and identifying the causes is very important.

In this contribution, we have defined and qualified the problems that can harm AI systems in terms of data bias. This notion of bias has been examined mainly at the data level [Dorleon *et al.*, 2021b]. For our contribution, using data science process, we identified areas in the process where different types of bias are likely to occur. The goal is to propose taxonomy of bias and where exactly they occur on the process. By using this taxonomy, one can be aware to these areas while working on a system or method to ensure that it has a low probability of causing potential harm or bias towards a group.

Being aware of bias and the inherent limitations of attempts to 'de-bias'decision systems, we will address in the next chapters in this thesis, different methods in order to mitigate and reduce bias in the data science process.

2.4 Fairness

Nowadays, we are observing extraordinary advances in the field of machine learning (ML). Automated decision-making based on ML algorithms are now replacing human at many critical decision points, such loan application or hiring. Against all odds, we might think that these ML algorithms are objective and free from bias, but that is not always the case. As we saw in section 2.2.1, many previously used real-world applications have exhibited bias towards race, place of living or gender. In the context of decision-making, fairness is the absence of any discrimination or favoritism towards an individual or a group based on their inherent or acquired characteristics referred to as protected feature by [Ntoutsi *et al.*, 2020]. According to [Fang *et al.*, 2020], a **Protected Feature** is a feature that is of particular importance either for social, ethical or legal reasons when making decision. The list of protected attributes may include: sex, race, ethnic or social origin, genetic features, language, religion or belief, political opinion, disability, age, sexual orientation, and so on. This type of features are protected by laws in some countries such as US, UK, Australia where several Equality Acts[4][5] has been adopted.

To assess fairness in automated decision-making, authors have used different strategies either by processing methods or data balancing. To evaluate fairness strategies, fairness metrics have used to qualify when a machine learning model is fair. We review below these strategies and the most used fairness metrics.

2.4.1 Fairness via Processing Techniques

For fairness methods involving processing techniques, we notice different mechanisms. The aim is to use processing technique either on the data, the algorithm or the output result. These mechanisms are generally classified into three categories: pre-processing, in-processing, and post-processing [Caton et Haas, 2020]. The following three subsections review the studies in each of these categories.

1. **Pre-Processing.** With these methods, the main idea is to keep the training data free of bias in order to learn a fair classifier. Thus, mechanisms in this category involve modifying training data before feeding it into a machine learning algorithm. Some mechanisms, such as [Kamiran et Calders, 2011] and [Thanh *et al.*, 2011], have proposed changing the labels of certain cases or readjusting their weight before training to make a more accurate classification.

2. **In-Processing.** The main idea of In-Processing methods is to modify the learning algorithms in order to eliminate potential bias and improve fairness [Agarwal *et al.*, 2018, Bechavod et Ligett, 2017]. For example, authors in [Kamishima *et al.*, 2012] have

[4] Australian Equality Acts
[5] UK Equality Acts

suggested to add a regularization parameter to the objective function that penalizes mutual information between protected feature and the classifier.

3. **Post-Processing.** In Post-processing methods, the idea is to modify the results of a the trained classifier in order to ensure that the fairness goal is achieved. Some post-processing methods like in [Corbett-Davies *et al.*, 2017] have also tried to adjust and modify the decision boundary of the classifier in order to improve fairness. For example, in [Dwork *et al.*, 2018] authors have proposed a decoupling technique to learn a different classifier for each group. They further combined a transfer learning technique with their procedure for learning from out-group samples. Generally, post-processing methods do not use classical fairness metrics to evaluate their proposed method, they tend to use instead individual or group fairness as it is shown in Table 2.4.

2.4.2 Fairness via Data Balancing

Fairness methods using data balancing techniques aim to address the problem of class imbalanced dataset which is a challenging issue in a wide variety of fields such as finance or justice [He et Garcia, 2009].

In a binary classification task for example, class imbalanced dataset refers to the skewed label distributions within data; i.e between-classes, where one class, referred to as **majority class**, dominates the other referred to as **minority class**. For example, given a dataset containing 100 observations for a binary output such as positive and negative, the majority class may contain 90 instances labeled as positive and the minority class 10 instances labeled as negative.

To solve the problem posed by class-imbalanced dataset on fairness, authors have introduced several methods to help overcoming and reducing the effects that class-imbalanced dataset may have on machine learning models. These methods can be either using oversampling strategy, undersampling or hybrid resulting in a mix of both. Below, we give more details on these strategies.

1. **Oversampling Methods.** The main idea of oversampling methods for dealing with class-imbalanced dataset is that new instances are generated in the class with less instances, i.e the minority class. The new oversampled data set contains equal number of samples in each class. A wide variety of oversampling methods exist in the literature [Viloria *et al.*, 2020, Mohammed *et al.*, 2020, Huda *et al.*, 2018, He *et al.*, 2008] each with the aims of avoiding bias and improving machine learning model's accuracy towards minority group. To date, one of the most used technique for oversampling can be found in the work of [Chawla *et al.*, 2002] who introduced a method named SMOTE: Synthetic Minority Oversampling Technique. With this method, authors seek to oversample instances in the minority class by using the instances's neighborhood with the k-NN algorithm [Zhang, 2016]. SMOTE

first locates neighborhoods of minority instances, and then it generates synthetic instances by combining the attribute values of the neighborhood.

2. **Undersampling Methods.** These methods aim to reduce instances from the majority class; i.e the class with more instances. The new under-sampled data set contains equal number of samples in each class. Like the oversampling strategy, there is a variety of work that exist [Hoyos-Osorio *et al.*, 2021, Krawczyk *et al.*, 2021, Zheng *et al.*, 2021].

   Particularly, we notice a recent work [Yao et Wang, 2021] from the literature. In this work, authors have proposed to the solve the problem of low classification accuracy of minority classes caused by data imbalance; they proposed an undersampling classification algorithm based on mean shift clustering for imbalanced data (UECMS). The UECMS method uses mean shift clustering and instance selection for the samples of majority classes to complete the undersampling. The selected samples and all the minority samples from the original data set form a new balanced data set. Then, bagging-based ensemble learning algorithms are used to classify the balanced data sets. Authors do believe that the experimental results show that the UECMS method improves classification accuracy and fairness for minority classes.

3. **Hybrid Methods.** These methods are a mix of oversampling and undersampling methods to deal with class imbalanced datasets . According to [Liu *et al.*, 2017], combining undersampling or oversampling methods can result in models with better performance. Like in the other strategies, the new resulted data set contains equal number of samples in each class.

   For hybrid methods that deal with class imbalanced problem, we notice the recent work of [Elyan *et al.*, 2021] where authors have proposed a new hybrid approach. The approach aims to reduce the dominance of the majority class instances using class decomposition and increasing the minority class instances using an oversampling method. Unlike other undersampling methods, which suffer data loss, this method preserves the majority class instances, yet significantly reduces its dominance, resulting in a more balanced dataset and hence improving the result's accuracy and fairness. A large-scale experiment using 60 public datasets was carried out to validate the proposed methods.

2.4.3 Fairness Metrics

Fairness metrics are used to evaluate how fair is a machine learning model. To evaluate a model, these fairness metrics are conditionally based on the use of what is referred to as protected (sensitive) feature in the data.

We present below the different fairness metrics that can be used to deal with bias related to protected features. To date, more than twenty two fairness metrics exist [Verma et Rubin, 2018], however there is no consensus on which one is better. In this section, we are giving more details on fairness metrics that have been used to mitigate bias in machine learning models and their definitions, particularly in supervised tasks.

To introduce the related fairness metrics, we consider an input dataset $S = \left(X_{m,n}\,, Y_{1,n}\right)$ consisting of n observations and m features. Let f be a learning model and its performance score $f[S]$ which will be used to predict a binary output $\hat{y} \in \{0, 1\}$. Let $F = \{F_1, F_2, ..., F_m$ be the feature space. Each sample $x_{\bullet i}$ is associated to a protected feature P, with $P \in F$. For simplicity we consider that P is binary: $P \in \{P_0, P_1\}$, thus P_0 represents an unprivileged group and P_1 a privileged group. Likewise, we consider $\hat{y} = 1$ to be the preferred outcome, assuming it represents the more desirable of the two possible outcomes. For instance, P ='gender' could be the protected attribute with P_0 ='female', the unprivileged group, and P_1='male' the privileged one. Suppose for some samples we know the ground truth; i.e., the true value $y \in \{0, 1\}$. Note that these outcomes may be statistically different between different groups, either because the differences are real, or because the model is somewhat biased. Depending on the situation, we may want our estimate $\hat{y}$ to take these differences into account or to compensate them.

Below, we are going to give an overview of the most frequently used metrics to assess fairness in machine learning algorithms in the literature.

1. **Demographic Parity [Bellamy *et al.*, 2018]**. Also called Statistical Parity, this metric suggests that a predictor is unbiased if the prediction ŷ is independent of the protected feature P such that Pr(ŷ| P) = Pr(ŷ) (Pr is the prediction rate). This means that the same proportion of each subgroup is classified as positive. This metric has been used in many recent state-of-the-art methods [Räz, 2021, Hertweck *et al.*, 2021, Yeom et Tschantz, 2021] for fairness improvement. Difference between prediction rates of the subgroups can be used to assess fairness from this metric. Assuming this difference is noted Demographic Parity Difference (DPD), it can be defined as:

$$DPD = Pr(\hat{y} = 1|P = 1) - Pr(\hat{y} = 1|P = 0) \tag{2.1}$$

2. **Equalized Odds [Bellamy *et al.*, 2018]**. This metric states that the prediction ŷ is conditionally independent of the protected feature P, given the true value y: Pr (ŷ|y, P) = Pr (ŷ|y). This metric is widely used and adopted by recent state-of-the-art methods [Mary *et al.*, 2019, Salazar *et al.*, 2021, Iosifidis et Ntoutsi, 2019, Park *et al.*, 2021] This means that the true positive rate and the false positive rate will be the same between the unprivileged (P_0) and privileged groups (P_1). To assess fairness from this metric, one might use the difference between prediction

rates (positive and negative). Assuming this difference is called Equalized of Odds Difference (EOD), it can be defined it as:

$$EOD = Pr(\hat{y} = 1|P = 1, y = y_i) - Pr(\hat{y} = 1|P = 0, y = y_i), y_i \in \{0, 1\} \tag{2.2}$$

3. **Equal Opportunity [Bellamy *et al.*, 2018]**. This metric is basically the same as Equalized Odds, however it only focuses on a particular label y = 1 of the true value so that Pr ($\hat{y}$|y = 1, P) = Pr ($\hat{y}$|y = 1). It is also widely used in recent fairness studies [Park *et al.*, 2021, Khalili *et al.*, 2021, Yeom et Tschantz, 2021]. One might use use the difference between prediction rates between the subgroups to assess fairness from this metric. Assuming this difference is called Equal Opportunity Difference (EOpD), it can defined it as follow:

$$EOpD = Pr(\hat{y} = 1|P = 1, y = 1) - Pr(\hat{y} = 1|P = 0, y = 1) \tag{2.3}$$

We also notice that some work in the literature have combined both processing techniques and fairness metrics. So, we present in Table 2.4 a summary of these work that used both fairness metrics and other mechanisms to improve fairness.

Table 2.4: **State-of-the-art of fairness metrics and other mechanisms**

Method	Strategy	Metrics
[Kamishima *et al.*, 2012]	In-Processing	Normalized Prejudice
[Feldman *et al.*, 2015]	Pre-Processing	Disparate Impact
[Zafar *et al.*, 2017]	In-Processing	Equalized Odds
[Zemel *et al.*, 2013]	Pre & In-Processing	Demographic Parity
[Goh *et al.*, 2016]	In-Processing	Disparate impact
[Heidari et Krause, 2018]	In-Processing	Disparate Impact
[Lohia *et al.*, 2019]	Post-Processing	Individual/Group Fairness
[Kim *et al.*, 2019]	Post-Processing	Group Fairness
[Petersen *et al.*, 2021]	Post-Processing	Group Fairness

2.4.3.1 Datasets used for fairness study

Beside the fairness metrics and processing techniques used for fairness improvement, we notice that most of the work from the literature have used at least twelve common

datasets. We presented in Table 2.5 those datasets with information about their size, the number of protected feature and their availability.

Table 2.5: **Commonly used datasets in fairness study**

Dataset	Area	Size	Protected Feature	Availability
Ricci	Job promotion	118	Race	UCI
ProPublica	Justice	6 167	Race, Sex	GitHub
Adult Income	Economic	48 842	Age, Sex, Race	UCI
German Credit	Credit Scoring	1 000	Sex, Age	UCI
Mexican Poverty	Demographic	183	Race	Atlas
Diabetes	Medical	100 000	Age	UCI
Heritage Health	Medical	143 473	Age	Kaggle
College Admission	Education	20 000	Race, Sex	Kaggle
Bank Churn	Marketing	45 211	Age	UCI
Loan	Finance	30 000	Sex	Kaggle
Dutch Census	Demographic	189 275	Sex	Micro-Data
Communities & Crime	Crime	1 994	Race	UCI

2.4.4 Summary

In this section we have presented a state-of-the-art of fairness via processing techniques and data balancing. We have also presented the fairness metrics that are used to assess fairness of a machine learning model. As highlighted by [Verma et Rubin, 2018], there exists a plethora of fairness metrics however, we have presented the ones that are mostly used in the recent work on fairness. We have also presented the most used datasets in Table 2.5. Among those datasets, we highlight in bold the ones that we have used in most of the work of our thesis. The reason why we have chosen to use those specifics datasets is because they represent a complete benchmark and allow us to compare our results to other work that have used the same dataset. We have also pointed out in Table 2.4 some work that combine both metrics and processing techniques to assess fairness.

2.5 Feature Selection (FS)

Nowadays, many real world applications [Amarasinghe *et al.*, 2018, Farahani *et al.*, 2019, Paparrizos *et al.*, 2011] deal with so called high-dimensional data. High dimensional data refers to a dataset in which the number of features m is larger than the number of samples n [Nordhausen, 2009]. In gene expression, health care, financial or genomics datasets for instance, it is not uncommon to encounter high-dimensional datasets $(m > n)$.

However, in the filed of machine learning, high-dimensional datasets became a major issue [Reddy *et al.*, 2020] due to their size and the amount of resources required to process them. Learning performance is impacted by high-dimensional datasets [Ullah *et al.*, 2017]. Naturally, one may believes that more features we get, the more information we get from the features, but that is far from true because it becomes more and more difficult to extract meaningful conclusions from a dataset as the dimensionality of the data increases. Hence there is a need to resort to dimensionality reduction techniques in order to reduce the size of these data.

Among the methods commonly used in dimensionality reduction, we notice particularly feature extraction and feature selection [Khalid *et al.*, 2014, Ullah *et al.*, 2017] which are the most used. In feature extraction, new features are created based on the transformation or combination of the original input features set. The new transformed or reduced features set is believed to be more manageable facilitating its later processing [Guyon *et al.*, 2008]. In feature selection, we try to find the best subset among the input feature set. A feature is an attribute that has an impact on a problem or is useful for the problem, and choosing the important features is an important step in building reliable and efficient machine learning model. On Fig 2.5, we present an overview of the basic difference between these two dimensionality reductions techniques.
In this thesis, we are focused on feature selection as dimensionality reduction. As we did see in section 2.2.3, feature selection could be a source of bias in the data preparation stage of the data science process. The selection of features is based on some methods and measures. Depends on the methods or measures used, it is possible to introduce bias in the process. During the work of this thesis, we were mainly focus on bias related to feature selection. We made this choice because we believe that mitigating bias at an entry level could be benefit to ensure fairness in automated decision-making system.

2.5.1 Definition

Machine learning model's performance could be negatively impacted by using irrelevant features from the input data. Indeed, it is important to identify and select the most relevant features in the data. The goal is therefore to obtain a meaningful subset of relevant that is vital for improving efficiency and reducing over-fitting of a learning model. This is done by feature selection, as it helps to understand data, improve prediction performance and reduce computational costs [Gutkin *et al.*, 2009].

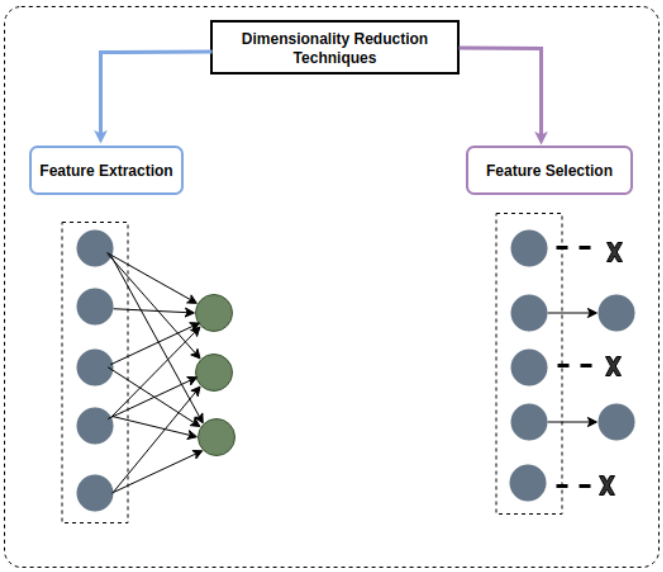

Figure 2.5: Dimensionality Reduction Techniques: Feature Extraction (FE) vs Feature Selection (FS). FE creates new features by combining features of the original input, whereas FS removes features that are considered either irrelevant or redundant, while the rest are kept unaltered.

Feature selection, also known as attribute selection or variable subset selection, is the process of selecting a subset of relevant features for use in model construction [Li *et al.*, 2017]. Depends on the strategy used to select the best subset of features, the process of selecting feature can be seen as Filter, Wrapper, Embedded or Hybrid. Below, we are giving more details on each feature selection method and explain the pro and cons of each method. We also show by using a example how using different feature selection methods can lead to different selected subsets. However, before introducing the different categories of feature selection, we introduce in the next sections some basics settings to explain what is a relevant or irrelevant feature.

2.5.2 Context and Description

A fundamental problem in machine learning is to approximate the functional relationship $f()$ between inputs data $X = \{x_1, x_2, ..., x_M\}$ and output Y. Sometimes the output Y is not determined by the complete set of input features $X = \{x_1, x_2, ..., x_n\}$ but only by a subset of $X = \{x_{(1)}, x_{(2)}, ..., x_{(m)}\}$ where $m < M$.

We use Figure 2.6 below to show the different stages of the FS process which is then explained. The performance of the process depends on the decision taken at each level.

1. **Search Direction:**

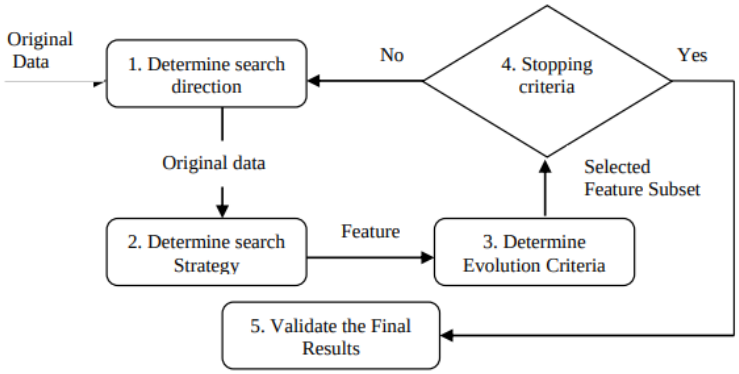

Figure 2.6: Stages in Feature Selection by [Venkatesh et Anuradha, 2019]

[Chandrashekar et Sahin, 2014] state that the first step in the FS process is to find the search direction and starting point. Search directions are broadly classified into three types: forward search, backward search and random search [Chandrashekar et Sahin, 2014, Ang *et al.*, 2016]. The search process can start with an empty set where new features are recursively added at each iteration, such a phenomenon is called forward search. Conversely, the backward search method starts with a full set of features and features are iteratively removed until the desired subset of features is reached. The other approach is a random search, which builds the feature subset by iteratively adding and removing features. Once the search direction is complete, the search strategy can then be applied.

2. **Search Strategy:**
   The scientific literature allows us to classify search strategies into random, exponential and sequential search [Chandrashekar et Sahin, 2014]. Exponential search has the disadvantage of requiring 2^m combinations including empty sets for m features. It is an exhaustive search strategy. To overcome this drawback, random search strategies have been introduced. In sequential search, features are sequentially added to an empty set or removed from the complete set [Pudil *et al.*, 1994, Doak, 1992].

3. **Evaluation Criteria:**
   The best features are selected based on an evaluation criteria [Chandrashekar et Sahin, 2014, Li *et al.*, 2017]. Based on this criteria, FS methods are classified into Filter, Wrapper and Embedded. More details on these concepts are discussed later in section 2.5.4.

4. **Stopping Criteria:**
   Stopping criteria specify when the FS process should stop. A good stopping criterion leads to low computational complexity to find an optimal subset and also

overcomes over-fitting. Stopping criterion choice is influenced by the choices made in the previous steps [Kumar et Minz, 2014]. Some common stopping criteria are a predefined number of features, a predefined number of iterations or a progress percentage over two successive iteration steps [Visalakshi et Radha, 2014].

5. **Final Results Validation:**
   To validate the results, feature set validation techniques are used [Ambure *et al.*, 2019]. Cross validation, confusion matrix, F1-score, AUC are some of the validation techniques [Raschka, 2018]. Cross-validation is the most commonly used validation method [Venkatesh et Anuradha, 2019].

2.5.3 Selection Condition

Reduction of the feature set is based on conditions such as relevance and redundancy with respect to the objective. Specifically, according to [Yu et Liu, 2004] a feature is usually classified as 1) strongly relevant, 2) weakly relevant, but not redundant, 3) not relevant, and 4) redundant.

There are a number of definitions [Yu et Liu, 2004, Li *et al.*, 2017] in the machine learning literature of what feature "relevance" means. The reason for this variety is that it usually depends on the question: "relevant for what?" Different definitions may be more appropriate depending on the objectives set. Here, to describe the relevance and its meaning, we introduce a simple modeling.

Let $M = \{m_1, m_2, ..., m_p\}$ where there are p features used to describe examples and each feature m has a domain F_m. For example, a feature can be boolean (is it red?), discrete with multiple values (what color?) or continuous (what length?). An example is a point in the space $X = \{x_1, x_2, ..., x_n\}$. Let S be a sample of features where each data point x_i is an example associated with an associated label or classification (which can also be boolean, discrete or continuous). Let T be a probability distribution on the space of instances X , and a target function denoted C. We then model the sample S as having been produced by repeatedly selecting examples by T and then labeling them according to the function C. Given this configuration, the most simple relevance is perhaps that of being "relevant to the target concept C". Then, a feature m_i is relevant for the target concept C if there exists a pair of instances x_i and x_j in the instance space such that x_i and x_j differ only in their assignment to m_i and $C(x_i) \neq C(x_j)$. We can then resume the above conditions as follows according to [Yu et Liu, 2004].

1. **Strongly relevant.** A strongly relevant feature is always needed for an optimal feature subset; it can not be removed without affecting the condition of the target distribution of origin.

2. **Weakly relevant, but not redundant** A weakly relevant feature may not always be necessary for a subset.

3. **Irrelevant.** Irrelevant features are not at all necessary.

4. **Redundant and irrelevant.** Redundant features are those that are of little relevance but can be completely replaced by a set of other features so that the target distribution is not disturbed.

Redundancy is thus always inspected when analyzing a subset of features while relevance is established for individual features. Thus, the goal of feature selection is to maximize relevance and minimize redundancy. This generally corresponds to the researches for a subset of features composed only of relevant features.

2.5.4 Feature Selection Methods

Feature selection methods can be categorized in several ways.

Based on the label information, feature selection approaches could be classified into three classes, such as supervised methods [Wolf et Shashua, 2003, Zhao et Liu, 2007, Nie *et al.*, 2010], semi-supervised methods [Zhao et Liu, , Xu *et al.*, 2010] and unsupervised methods [Wang *et al.*, 2017]. Labels provide convenience information, so that relevant features are selected to distinguish samples of different classes via supervised feature selection methods. In case where only a part of the data is labeled, semi-supervised feature selection could be used, so that both labeled and unlabeled data are exploited.

Based on the selection strategy, i.e. how features are selected, FS methods are classified into: Filter method - Wrapper method - Embedded method. This classification on the basis of selection strategy being the most common [Yu et Liu, 2004], we focus in this thesis on this classification and we will discuss below these three methods, their strategy, their advantages and their disadvantages.

2.5.4.1 Filter Methods

Filter methods [Cherrington *et al.*, 2019, Sánchez-Marono *et al.*, 2007] are typically used as a pre-processing step. In Filter method, the feature selection process is independent of any machine learning algorithm as illustrated on Fig. 2.7. Features are selected based on their scores using various statistical tests for their correlation with the output variable (feature). Below, we define some statistical tests that are generally used by Filter methods.

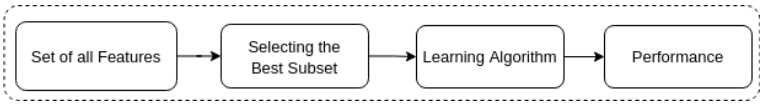

Figure 2.7: Filter feature selection method

1. **Pearson Correlation**
   Pearson Correlation [Benesty *et al.*, 2009] is used as a measure to quantify the linear dependence between two continuous features $x_{i\bullet}$ and $x_{j\bullet}$. Its value varies from -1 to $+1$. The Pearson correlation is given as follows by:

$$r = \frac{\sum_{i=1}^{n}(x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \overline{x})^2(y_i - \overline{y})^2}} \tag{2.4}$$

   With r the correlation coefficient, x_i the values of the X-variable in a sample, $\bar{x}$, mean of the values of the $x_{i\bullet}$-variable, y_i values of the $x_{j\bullet}$-variable in a sample and $\bar{y}$ mean of the values of the $x_{j\bullet}$-variable. Pearson correlation can only detect linear dependencies linear between features and the output target.

2. **Mutual Information**
   Mutual information [Estévez *et al.*, 2009, Peng *et al.*, 2005] is a measure of the mutual dependence between two features. It is based on the information theoretical concept of entropy, a measure of uncertainty of a random variable. The entropy of a random variable $x_{i\bullet}$ is defined by:

$$H(x_{i\bullet}) = -\sum_{i} P(x_i)\log_2 P(x_i) \tag{2.5}$$

   and the entropy of $x_{i\bullet}$ observing another variable $x_{j\bullet}$ is:

$$H(x_{i\bullet}|x_{j\bullet}) = \sum_{j} P(y_j)\sum_{i} P(x_i|y_j)\log_2\left(P(x_i|y_j)\right) \tag{2.6}$$

   Where $P(xi)$ represents the prior probabilities for all values of $x_{i\bullet}$ and $P(x_i|y_j)$, the conditional probabilities of $x_{i\bullet}$ being given the values of $x_{j\bullet}$. The statistical difference between $H(x_{i\bullet})$ and $H(x_{i\bullet}|x_{j\bullet})$ is then called information gain or mutual information and represents the degree of correlation between $x_{i\bullet}$ and $x_{j\bullet}$. Thus, using formulas 2.5 and 2.6, information gain or mutual information can be defined as:

$$IG(x_{i\bullet}, x_{j\bullet}) = H(x_{i\bullet}) - H(x_{i\bullet}|x_{j\bullet}) \tag{2.7}$$

3. **Chi-Square**
   Chi-square [Jin *et al.*, 2006] is a statistical test applied to groups of categorical features to assess the likelihood of correlation or association between them using their frequency distribution. The general formula is as follows:

$$\chi^2 = \sum \frac{(O_i - E_i)^2}{E_i} \tag{2.8}$$

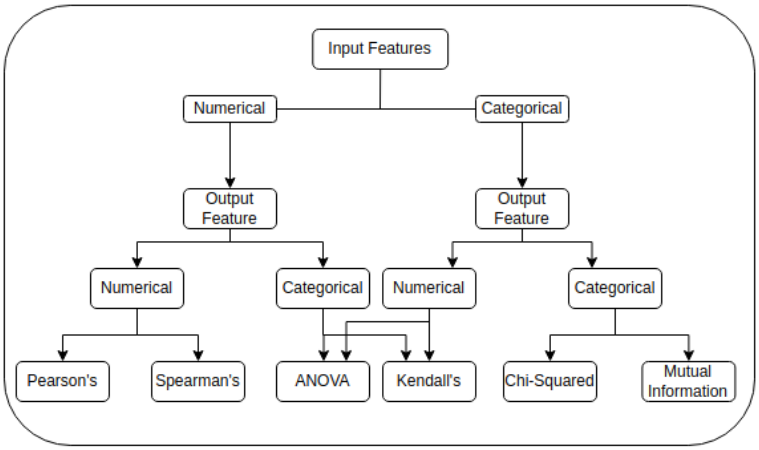

Figure 2.8: Statistical tests for Filter selection method [Jović *et al.*, 2015]

with O_i the observed value and E_i the expected value. Many other statistical tests are used in the literature. Based on the input features types, different statistic tests can be used. The following diagram gives an overview of how these tests can be used.

2.5.4.2 Wrapper Methods

In so-called "Wrapper" methods, the process aims to use a subset of features and create a model for their use [Chandrashekar et Sahin, 2014, Zhu *et al.*, 2007]. Based on the inferences drawn from the previous model, one decides to add or remove features from the subset. Fig. 2.9 gives an overview of how Wrapper feature selection works. The

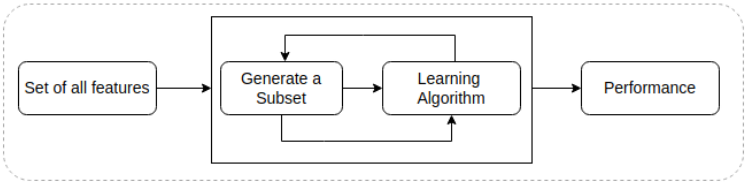

Figure 2.9: Wrapper feature selection method

feature selection under Wrapper is essentially reduced to a research problem. These methods are usually very computationally expensive. Based on how the research is done, Wrapper methods can be categorized as: Forward Feature Selection, Backward Feature Selection, Recursive Feature Elimination.

1. **Forward Feature Selection (FFS)**
   FFS [Ren *et al.*, 2008, Mao, 2002] is an iterative method where we start with

having no features in the model. At each iteration, we keep adding the feature that best improves our model until adding a new variable does not improve the model's performance. On Fig. 2.10, we show using a example of 5 features, how forward feature selection works. The process starts with an empty set of features then adds the most relevant one in step two. It then trains a model with the added feature then adds more features to the previous model until all the relevant features until a defined stopping criteria is reached or until there is no more features left.

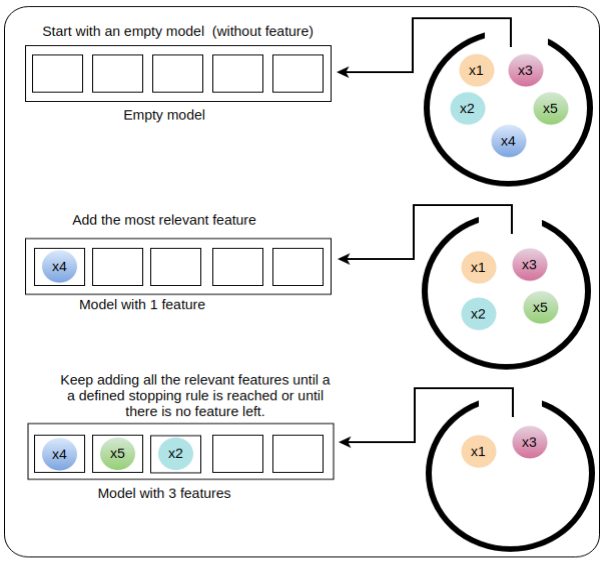

Figure 2.10: Forward feature selection process

2. **Backward Feature Selection (BFS)**
   In BFS [Kostrzewa et Brzeski, 2017, Tharmakulasingam *et al.*, 2020], we start with all features and remove the least significant feature at each iteration, which improves model performance. We repeat this until no improvement is seen when removing features. Below on Fig. 2.11, we show using a example of 5 features how backward feature selection works.

3. **Recursive Feature Selection (RFS)**
   RFS [You *et al.*, 2014, Zeng *et al.*, 2009] is an optimized Wrapper method that aims to find the best performing feature subset. It repeatedly creates models and discards the best or worst performing features within each iteration. It builds the next model with features set aside until all features are exhausted. It then ranks the features according to the order of their selection.

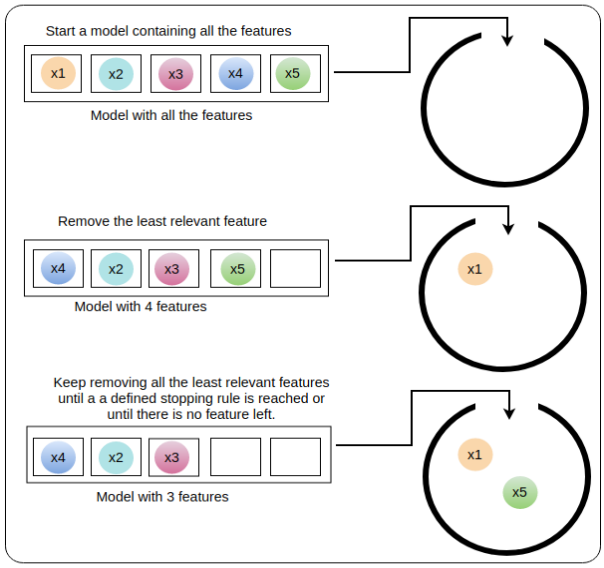

Figure 2.11: Backward feature selection process

2.5.4.3 Embedded Method

Embedded feature selection methods aim at combining the advantages of "Filter" and "Wrapper" methods. Embedded methods are implemented by algorithms that have their own built-in feature selection methods as mentioned by [Guyon et Elisseeff, 2003, Chandrashekar et Sahin, 2014]. Some of the most popular examples of those methods incorporate regression algorithms such as LASSO [Muthukrishnan et Rohini, 2016] and RIDGE [Zhang *et al.*, 2018] which have built-in penalty functions to reduce over-fitting.

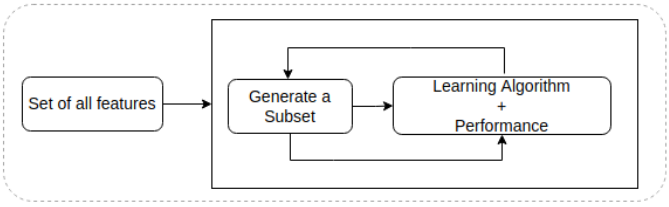

Figure 2.12: Embedded feature selection method

2.5.4.4 Experimentation

Different feature selection methods exist and there are no consensus on which one is better. They each have the own pros and cons. Here, first we summarize in Table 2.6

the pros and cons of the three categories of feature selection methods that we mentioned above. Secondly, we show below using an example, another drawback of different feature

Table 2.6: **Feature Selection Methods - Pros and Cons**

Method	Pros	Cons
Filter	Efficient and independent of the learning algorithm. Suitable for low-dimensional data. Computationally faster than Wrapper and Embedded	Does not consider correlation between features. Does not correctly recognize patterns during the learning phase.
Wrapper	Considers the correlation between features and labels. Considers dependencies between features. More precise than the Filter method. Computationally efficient.	More complex calculation and time consuming. Some features may not be taken into account for evaluation in the event of abandonment at the first stage. Can leads to over-fitting.
Embedded	More accurate than Filter and Wrapper	Computationally more expensive than Filter and it is not suitable for high-dimensional data.

selection methods.

We notice that when using different statistical tests for selecting feature in Filter or different strategies for Wrapper, we end up with different selected subsets. In the following, we show how using different feature selection methods on the same dataset can output different subsets based on each feature's importance.

For this example, we will use different feature selection strategies to asses features importance in a dataset. We will use two different statistical tests for Filter (Chi^2, ANOVA), two Wrapper strategies (FBS, BFS) and one Embedded feature selection method. For the data, we will be using Heart Disease dataset (2.7) from the UCI public directory [Dua et Graff, 2017]. This dataset consists of 14 features including 1 output variable, 303 observations. The task using this dataset is, based on the set of features, to predict individuals with heart disease and those without it.

Table 2.7: **Heart Disease- Dataset details**

Nb of features	14
Nb of observations	303
Type	Medical
Features List	age, cp, trestbps, chol, fbs, restecg, thalach, sex, exang, oldpeak, slope, ca, thal

In Table 2.8, we report the features importance according to different feature selection methods. We can clearly see that the ordered feature importance list varies from a method to another. In the case where we would have wanted to choose 5, 6 or 7 features, we

would in fact be spoiled for choice because each feature selection method would have given a different list of feature importance. Even by doing so, it is not certain that these methods can limit, in such cases, biases related to the chosen FS method.

Table 2.8: **Heart Disease - Features Importance with Filter (ANOVA & CHI²) , Wrapper (FBS & BFS) & Embedded**

Method	Ordered Features List from 1 - 13
ANOVA	exang, cp, oldpeak, thalach,ca,slope, thal, sex, age, trestbps, restecg, chol,fbs
CHI²	thalach, oldpeak, ca, cp, exang, chol, age, trestbps, slope, sex, thal, restecg, fbs
FBS	cp, fbs, restecg, ca, thal, slope, trestbps, sex, chol, thalach, exang, oldpeak, age
BFS	cp, ca, thal, slope, sex, oldpeak, chol, fbs, thalach, age, exang, restecg, trestbps
Embedded	cp, thalach, thal, ca, oldpeak, age, chol, trestbps, exang, slope, sex, restecg, fbs

2.5.4.5 Other Existing Methods

Other feature selection methods are emerging in the literature. Those methods are based on existing feature selection methods in order to propose a more robust and efficient selection strategy. Among those emerging feature selection methods, we can site:

- **Hybrid Methods.** These methods [Hsu *et al.*, 2011, Lu *et al.*, 2017] combine different types of other feature selection methods in order to benefit from their advantages.
- **Ensemble Methods.** Rather than using a single approach to select a subset of features like the previous methods, ensemble methods [Moghimi *et al.*, 2018, He *et al.*, 2019, Ndirangu *et al.*, 2019] combine different approaches to obtain the best possible subset of features. There are no consensus on how to combine these approaches, since many methods are available. The great advantage offered by these methods is that they benefit the best advantages from other selection methods and, as such, can reduce their disadvantages. Ensemble methods are known for their high performance and precision and the fact that they are more robust against high-dimensional datasets. However, they can be computationally expensive as they aim to process large amounts of data.

2.6 Summary

In this chapter, we have presented a state-of-the-art of the three main problematic of our thesis. Basically we have addressed the notion of bias, the notion of fairness and the notion of feature selection.

We have investigated the different types of biases that can occur when building a machine learning model and introduced a taxonomy of these different types of biases. We have also looked at the different sources of bias in terms of data imbalance and protected feature that can impact machine learning algorithms in terms of fairness. For data imbalance or protected features issues, we have given details on the different methods that have been used or proposed in the machine learning literature.

For the notion of fairness, we have detailed the most used metrics fairness and different fairness strategies that have been used or proposed in the past years.

For feature selection methods, we have looked at all the three categories of feature selection methods named as Filter, Wrapper and Embedded. We also show by using an example, an experiment where using different feature selection methods on a same dataset can result in different selected subsets. We also presented other existing feature selection methods with their strategy of combining different other feature methods in order to maximise the chance of selecting the most relevant features.

In the next chapters, we will present our contributions to the main problematic that we have investigated in this thesis under the notions of bias, protected features, data imbalance, fairness and feature selection.

Chapter 3

The Feature Selection and Redundancy Dilemma

Contents

3.1 Introduction

In this chapter, we investigated the notion of feature redundancy. By looking at existing and traditional feature selection methods, we pointed out their limit regarding feature redundancy, indeed most of these methods often make the assumption of the independent feature only, i.e the output variable. To overcome the limits of existing feature selection methods regarding redundant features, we introduce a new feature redundancy strategy. We present this strategy in this chapter.

Generally, FS methods are used to assess features relevancy for a specific task. An efficient feature selection method normally should be able to select relevant and non-redundant features in order to improve learning performance and training efficiency. Globally, it is easier to remove irrelevant features than finding redundant ones. Thus, the difficulty in selecting features now is finding the ones that are redundant. Existing works for features redundancy analysis such as [Yu et Liu, 2004, Peng *et al.*, 2005, Wang *et al.*, 2020] introduced approaches to reduce redundancy. However in the case of non-independent features, our study shows that these methods inappropriately remove redundancy because they required user to set a threshold.

In this chapter, our focus lies on feature redundancy and our contribution can be summarized as follows:

- we propose a new method for analyzing feature redundancy.
- the proposed redundancy method does not require user to set a threshold.

In section 3.2, we look at the existing feature redundancy methods and how they addressed the issue of feature redundancy. We exposed the proposed approach in section 3.3 then detailed experiments and results obtained in section 3.4.

3.2 Related Work

We notice many existing work that discuss feature redundancy and it is mainly described is in the sense of correlation between features with respect to the output variable.

This is the case of authors in [Wang *et al.*, 2020] whose redundancy approach is defined by a high correlation coefficient between features. This "high correlation" is determined by a chosen threshold. In their approach, authors in [Wang *et al.*, 2020] believed that redundancy could be strong, moderate or weak. In their method called Redundancy Analysis Based Feature Selection (RABFS), they use maximum information coefficient (mic) as correlation measure in order to establish a threshold, analyze the redundancy between features and create a subset of relevant features. Correlation between features determine if the redundancy is strong, moderate or weak.

In [Peng *et al.*, 2005], authors have presented a feature selection method described as "Minimum Redundancy & Maximum Relevance (mRMR). This method, based on

Mutual Information (MI) as a correlation measure, makes it possible to select features that have a strong correlation with the output and a weak correlation between them in order to maximize relevancy and minimize redundancy. The authors in [Yu et Liu, 2004] proposed another feature selection method called Fast Correlation-Based Filter(FCBF), which uses symmetrical uncertainty (3.4) as correlation measure and approximate Markov blanket to remove redundancy among features. In [Zhang *et al.*, 2020], authors introduced GRRO (Global Relevance and Redundancy Optimization), a multi-label feature selection. In this method, authors proposed a general global optimization framework incorporating feature relevance, feature redundancy, and label correlation based on the use of information gain.

However, when analyzing the methods cited above, we found that they inappropriately remove redundancy because they require users to set a single-defined threshold. We observed several problems with the strategy of using a single-defined threshold:

- feature redundancy depends on the threshold set, that being said, different thresholds led to different sets of redundant features; thus, different models.
- as more redundant features are removed according to the single-specified threshold, we observed a significant loss of performance.

Given the limitations of these above redundancy approaches, we propose a new criterion to evaluate the redundancy between relevant features. Unless other proposed redundancy methods, our method does not require users to set a threshold.

3.3 The Proposed Redundancy Analysis Method

In this section, we present our proposed redundancy analysis method which uses a redundancy criterion based on symmetrical uncertainty [Ullah *et al.*, 2017] as a correlation measure. We identify a list of redundant features without a threshold setup by users.

3.3.1 Correlation Measure

We use symmetrical uncertainty (SU) as correlation measure, it is based on information gain [Raileanu et Stoffel, 2004] which for two variables X,Y is given by:

$$IG(X,Y) = H(X) - H(X|Y) \tag{3.1}$$

Using this measurement, a feature Y is considered to be more correlated to a feature X than a feature Z if and only if: $IG(X|Y) > IG(X|Z)$. Information gain uses the notion of entropy to measure the mutual dependence between two variables. The entropy for a random variable X is:

$$H(X) = -\sum_{i} P(x_i) \log_2 P(x_i) \tag{3.2}$$

and between two variables X and Y, it is given by:

$$H(X|Y) = \sum_{j} P(y_j) \sum_{i} P(x_i|y_j) \log_2 \left(P(x_i|y_j)\right) \tag{3.3}$$

With $P(x_i)$: probabilities for all values of X, $P(x_i|y_j)$: conditional probabilities between X given the values of Y.
However, the study in [Raileanu et Stoffel, 2004] showed that features with more values are favored with information gain, thus its normalized version known as Symmetrical Uncertainty (SU) is used. Using equation (3.1), (3.2) and (3.3), SU is defined as:

$$SU(X,Y) = 2\Big[\frac{IG(X,Y)}{H(X) + H(Y)}\Big] \tag{3.4}$$

3.3.2 The Redundancy Criterion

To avoid the use of a defined threshold, we define our redundancy criterion by focus on absolute feature redundancy. The criteria is defined as follow:

- Two features F_j and F_i are redundant with respect to a target class Y(the output variable) if and only if they provide exactly the same amount of information for the output variable. In other words if and only if:

$$SU(F_j, Y) = SU(F_i, Y) \tag{3.5}$$

$SU(F_j, Y)$ refers to the symmetrical uncertainty between a feature F_j and the class Y and $SU(F_i, Y)$ refers to the symmetrical uncertainty between a feature F_i and the class Y. If F_j and F_i are redundant, the least relevant one needs to be deleted.

3.3.3 Redundancy Analysis Algorithm

The steps of redundancy analysis are summarized in Algorithm 1. As inputs, the algorithm takes a list **F** of ranked relevant features resulting from a feature selection method and Y as the target class. From this list **F**, we choose the most important or relevant feature F_j (line 1). Then, the symmetrical uncertainty between F_j and the next remaining feature F_i in the list **F** (line 2 to 5) is calculated. If the redundancy criterion is true (line 6), the feature F_i is deleted from the list **F** (line 7) then the next feature F_i in **F** is used (line 9) until all the remaining features in **F** have been used. Then we start over by varying F_j in the list (line 11) and so on until we have considered every remaining feature as F_j (line 12). After removing all the redundant ones, we add all the remaining features to the list **F'**, (**F'** $\leq$ **F**). This list, **F'**, containing only relevant and non-redundant attributes, will be used for the desired learning task.

Algorithm 1: Absolute Redundancy Algorithm

Input: **F**$(F_1, F_2, ..., F_n)$, Y // Features list **F** and target class Y
Output: **F'** //Final List of non-redundant attributes

$F_j \leftarrow$ getFirstElement (**F**)
do begin:
    $F_i \leftarrow$ getNextElement(**F**)
    while F_i <> NULL:
        compute SU(F_j, Y), SU(F_i, Y)
        if SU(F_j, Y) == SU(F_i, Y):
            $F \leftarrow F \setminus \{F_i\}$
        end if
        $F_i \leftarrow$ getNextElement (**F**)
    end while
    $F_j \leftarrow$ getNextElement(**F**)
end until (F_j == NULL)
F' $\leftarrow$ **F**

3.4 Experimental Approach

3.4.1 Experimental Design

The diagram in 3.1 reflects our experimental design to perform the redundancy analysis. We carry out our experiments in a way so that we can compare our results with other existing methods. The existing methods that can reduce redundancy and to which we compare our results are RABFS, mRMR and FCBF respectively. RABFS [Wang *et al.*, 2020] uses maximum information coefficient to establish a threshold and analyze features redundancy and build a subset of features for training. In mRMR [Peng *et al.*, 2005], the aim is to select features with a high relevance with the target and a low redundancy between themselves. FCBF uses symmetrical uncertainty as correlation measure and approximate Markov blanket to remove redundancy [Yu et Liu, 2004].

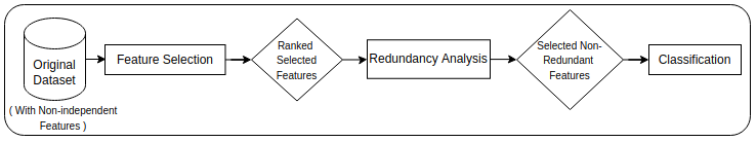

Figure 3.1: Redundancy Analysis: Design of our experimental approach

Feature Selection: we determine features' importance using a wrapper feature selection method and then rank the features in order of importance from the most important one to the least. Importantly, this ordered list constitutes the "Ranked Relevant Features".

Redundancy Analysis: to obtain the best subset of non-redundant features, we use the "Ranked Relevant Features" list to proceed to the redundancy analysis using the redundancy criterion that we defined in section 3.3.2 and the algorithm in 3.3.3.
Classification: the redundancy analysis produces a final reduced list of features, "Selected Non-Redundant Features". This list is used to perform a supervised learning task using SVM [Ring et Eskofier, 2016] and C4.5 [Salzberg, 1994] classifiers. These classifiers were used for comparison purpose with other existing methods.

3.4.2 Datasets used

To evaluate the performance of our method in finding redundant features and improving the performance of the learning task, six datasets including biological and text data from the UCI [Dua et Graff, 2017] were used. Those datasets (Table 3.1) were chosen based on their differences, the number of features varying from 325 to 22283. Plus, this choice will help us to compare the result of our method against other proposed methods that have used the same datasets. Specifically, the datasets used are:

- **Colon Dataset:** this dataset is composed of 40 colon tumor samples and 22 normal colon tissue samples. It contains information from 2,000 genes (features) and the goal is to classify the 62 samples into two classes: normal or tumor. It is available here [Alon *et al.*, 1999].
- **ALLAML Dataset:** Leukemia dataset [Abinash et Vasudevan, 2018], referred to as ALLAML dataset is a text-based benchmark in the cancer classification community. It contains in total 72 samples in 2 classes, ALL(acute lymphatic leukaemia) and AML(acute myeloid leukaemia), which have 47 and 25 samples, respectively. This dataset consists of 7,129 gene expression values which represent the features. This goal again here is to obtain a molecular classification of cancer.
- **PCMAC Dataset:** this is a two classes text-based dataset that consists of 1943 samples and 3289 features [Jiang *et al.*, 2019].
- **Prostate-GE Dataset:** the Prostate-GE [Yu et Zhao, 2018] is a clinical gene expression dataset consisting of 5966 features over 102 samples. It is a gene classification task which the goal of obtaining two classes: malignant and benign.
- **GLI-85 Dataset:** this dataset [Golub *et al.*, 1999] is a molecular classification dataset that contains two classes with 85 samples for 22283 features.
- **lung_small Dataset:** this is a text-based dataset with the goal of classifying samples based on lung cancer into two classes with or without tumors. Comparing to the others, this is a very small sized-dataset that contains only 73 samples with 325 features [Dua et Graff, 2017].

In Table 3.1 below, we give a summary of the datasets used.

Table 3.1: **Experimental Datasets used**

Dataset	samples	Nb of Features
Colon	62	2000
ALL/AML	71	7129
PCMAC	1943	3289
Prostage-GE	102	5966
GLI-85	85	22283
lung_small	73	325

3.4.3 Results

On a classification task, results of our method were compared to others based on the number of non-redundant selected features and the classification accuracy. We have used SVM with Gaussian kernel [Ring et Eskofier, 2016] and C4.5 [Salzberg, 1994] as classifiers. SVM is a supervised machine learning model that uses classification algorithms for two-group classification problems (see Annex A.1.1.6). C4.5 is an algorithm used to generate a decision tree that can be used for classification (see Annex A.1.1.7).

In tables 3.2, 3.3 and 3.4, we report the results obtained during our experiments including results reported by the other methods. To select our features, we used a wrapper features selection method with a forward strategy [Raschka, 2016]. Then, with the list of obtained features, we apply our redundancy criterion in order to obtain the list of the best features without redundancy. And finally, we perform a classification task using SVM and C4.5 so we can compare our results with other existing methods. To assess the effectiveness of our result in term of accuracy, we applied cross validation techniques on each dataset.

- **Selected Features**
  In Table 3.2, the other algorithms have found fewer features than our proposed method. This can be understood by the fact that the other methods used a filter strategy by setting a threshold to select the features while we have used a wrapper approach to subsequently select feature where we observe an absolute redundancy before training a model.

- **Results with SVM**
  In table 3.3 with SVM, our algorithm has a higher accuracy than other methods on

Table 3.2: **Number of Selected Features by method**

Dataset	Our Method	RABFS	FCBF	mRMR
Colon	7	3	9	3
ALLAML	5	3	6	3
PCMAC	54	28	112	28
Prostage-GE	21	16	4	16
GLI-85	13	4	5	4
lung_small	52	34	112	34

5 datasets. On the Prostate-GE dataset, both RABFS and mRMR have a higher accuracy (94%) than our method (93.83%). We notice that both of these methods have used exactly 16 features in their training data where we have used 21. Thus, on this dataset, theses methods have found more relevant features than our approach. But beside this dataset, our proposed method outperforms these methods on all the other datasets.

Table 3.3: **SVM Classification accuracy by method**

Dataset	Our Method	RABFS	FCBF	mRMR
Colon	**92.03**	91.66	90.0	78.57
ALLAML	**97.8**	96.07	92.85	97.14
PCMAC	**83.01**	80.91	77.51	56.25
Prostage-GE	93.83	**94.0**	91.99	**94.0**
GLI-85	**94.01**	92.77	90.69	89.30
lung_small	**88.0**	84.82	59.88	84.64
average	**91.44**	90.03	87.71	79.42

- **Results with C4.5**
  In table 3.4 with C4.5, the accuracy of our method is better on all the six datasets than all the other methods. We observe an average score of 91.65% for our method while the other methods have respectively 90.58%, 83.49% and 82.33%.

  In general, results on Tables 3.3 and 3.4 show that our method performs well.

Table 3.4: **C4.5 Classification accuracy by method**

Dataset	Our Method	RABFS	FCBF	mRMR
Colon	**92.04**	91.90	75.47	91.78
ALLAML	**96.56**	96.07	95.71	94.28
PCMAC	**84.01**	82.50	77.81	59.42
Prostage-GE	**91.73**	90.09	86.18	84.18
GLI-85	**96.60**	95.13	84.58	85.69
lung_small	**89.01**	87.76	81.14	78.61
average	**91.65**	90.58	83.49	82.33

3.5 Summary

In this chapter, we have addressed the issue of feature redundancy in feature selection methods. We have reviewed existing feature redundancy methods and pointed out their limits. Among these limits, we basically pointed out the fact that the existing methods usually require a single defined method and the fact the performance of the learning model will depend on the chosen threshold.

In order to overcome these limits, we have presented in this chapter a new redundancy analysis criterion based on symmetrical uncertainty which is a measure of correlation between features [Dorleon *et al.*, 2021a]. We have designed a redundancy analysis algorithm according to this criterion. Unless the other proposed redundancy methods, our algorithm does not require users to set a threshold. The performance of our method was experimentally compared to other methods such as RABFS, FCBF and mRMR on six different data sets. The comparative results show that our method finds satisfactory results.

However this method could be extended to a higher level of contribution and therefore has several limits. Firstly, the proposed method does not take care of protected feature. Moreover, the proposed method does not provide any fairness guarantee in order to deal with potential bias that can exist in the inputs data. For this reason, we provide in the next chapter an extended version of this work. The extended method is a trade-off between fairness and performance that deals with fairness, protected features and bias.

The source code, data and results for this contribution are available and can be accessed under request only via the thesis's repository on GitHub here Contrib3.

Chapter 4

Dealing with Fairness, Protected Features and Bias: Approach with Feature Suppression

Contents

4.1 Introduction

In a data-based learning problem, datasets with a lot of features can be problematic because some features may be either irrelevant or redundant; and as result it becomes difficult to extract meaningful conclusions [Reddy *et al.*, 2020]. To deal with this problem, dimensionality reduction technique such as feature selection is used [Reddy *et al.*, 2020]. The main objective of any FS method is to select a subset of relevant features from the input data that helps improving learning model [Guyon et Elisseeff, 2003]. Fairness is a quality of the prediction model which can be of high importance for the usability of the model. Some specific features known as protected could induce problems when dealing with fairness and it has been proved [Yeom *et al.*, 2018] that protected features can lead to unfair decisions against minority groups.

Looking at traditional feature selection methods while selecting features, we have observed two major problems which are:

- the selection of protected features whose presence leads to biased results and
- the selection of redundant features to the protected whose deletion leads to a loss in prediction performance.

In our work, redundancy is considered in the sense of correlation between non-independent features and the fact that the later can be strongly correlated with others enabling a classifier to reconstruct them.

Thus, in the work presented in this chapter, we focused on these two problems identified when using traditional feature selection methods that directly impact performance and fairness. Dealing with such issues, performance and fairness are computationally related and improving one leads to decreasing the other. In order to solve this, we introduce a method that allows to obtain the best trade-off between performance and fairness. Our method finds a set of relevant features without protected feature and with the least possible redundancy which maximizes the performance while ensuring fairness of the model obtained.
The contribution in this chapter can be summarized as follows:

- we introduce a more flexible way to use threshold for redundancy analysis by defining a threshold space instead of using a single value which could be subjective.
- we define an outcome-fairness algorithm for dealing with protected features in decision support algorithm:
  - the algorithm takes into consideration redundant features while making decisions on fairness, so that the overall performance remains high.
  - our method is based on two different fairness metrics in order to ensure the robustness of the approach.

- with our method, we show that it is possible to comply with data privacy policy by not using protected feature while remaining efficient and fair.
- our introduced method to achieve fairness is easily adaptable to any decision making problems (regression, clustering...) involving protected features.

The rest of this chapter is organized as follows: in section 4.2, we summarize the different existing methods to tackle the issues identified with their limitations. Section 4.3 presents our new approach on protected features, redundancy and fairness. The experimental results are described and analyzed in section 4.4.

4.2 Related Work

Many existing work proposed various feature selection methods to deal with the problem of protected features with regards to redundancy, fairness and bias.

We noticed some existing work that introduced different strategies to handle the problem posed by protected features. In [Dwork *et al.*, 2012], the authors have introduced a naive approach, named "Fairness Through Unawareness", consisting of removing completely all protected features of the dataset to ensure fairness. In [Fang *et al.*, 2020], the authors have used a fair-group strategy based on a bias metric (disparate impact), to improve the fairness of prediction results within each sub-group. In another method [Yan *et al.*, 2020] called "fair class balancing", the authors tackled the problem at a data processing level by proposing a method that allows to enhance model fairness without using any information about protected feature.

We highlight various limitations to the approaches cited above trying to improve fairness while considering protected features. Firstly, the approach of [Dwork *et al.*, 2012] of completely removing protected features may not solve the problem because there may be redundant features or even proxies to the protected. Because, as underlined by [Yeom *et al.*, 2018], some features known as proxies such as zip code, for example, can reveal the economic level or even predominant race of a residential area. Thus, this can still lead to racial discrimination in a decision making problem such as loan application despite the fact that zip code appears to be a non-protected feature [Zhang *et al.*, 2016]. Secondly, the approach of [Fang *et al.*, 2020] using fair-group does not take into account the existence of redundant features or proxies which, potentially carry the same information as the protected and can affect the prediction within groups. As underlined by [Yeom *et al.*, 2018], it is possible to reveal information about a protected feature using its redundant. We notice the same observation for the work in [Yan *et al.*, 2020], where redundant features to the protected are also ignored; this is dangerous in terms of fair outcomes when dealing with decisions problems involving minority groups.

Given the limitations of these above methods, there is a need for more in-depth research to overcome these limitations. Thus, we propose a new feature selection method

which allows the building of efficient and fair models without protected feature and with the least possible redundancy. Our new method is a trade-off between performance and fairness. To compute fairness, we use two different bias metrics that have been proposed in the literature [Bellamy *et al.*, 2018]: Demographic Parity and Equality of Odds. As each one of this metric uses a different criteria, they allow us to evaluate different fairness aspects of our approach.

4.3 The Proposed Trade-off Method

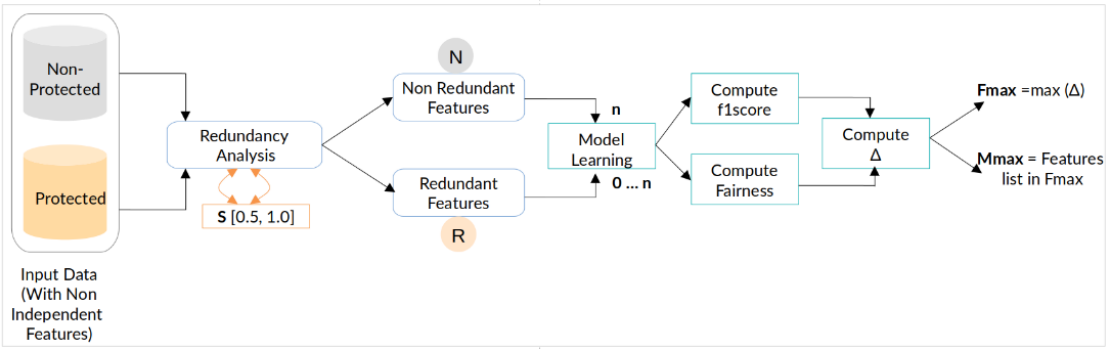

Figure 4.1: The proposed approach and its different stages

In this section, we present our approach, the different steps are illustrated in Figure 4.1. Our method takes as input a dataset divided into protected and unprotected features. Then, it performs a redundancy analysis based on a defined threshold space (S). Following the redundancy analysis, two subsets of features are obtained: a list of non-redundant features (N) and a list of redundant features (R). These two lists are used subsequently to train various models using all possible partitions between (N) and (R).The partitions are created by taking iterative combinations without duplication between the two subsets (N) and (R). Each partition is used to train a model, then for each model obtained, we calculate its f1-score, its fairness and a trade-off score. We will keep as final model the one which has the highest trade-off (delta) score, i-e, the most efficient and fair one.

With this new method, we propose an efficient solution to the problem related to protected and redundant features on performance and fairness. This method makes it possible to take into account i) redundancy, ii) protected features and iii) fairness. Below, we give more details and explain every step of the proposed approach.

4.3.1 Input Data

Once the input data is processed, we divided the input features into protected and non-protected. In the majority of cases, protected features are known or designated by a

system expert. In our case, to select protected features, we referred to the general data protection regulation (GDPR) of the European Parliament and Council on processing of personal data and the protection of privacy. According to article 4(13), (14) and (15) of the GDPR [EU, 2002], protected features include: gender, race, ethnicity, age and more. For example in the datasets [Dua et Graff, 2017] used (German Credit, Adult Income and Loan Approval), gender and race were used as protected.

4.3.2 Redundancy Analysis

As we saw above in section 4.2, using a single-defined threshold could be subjective for redundancy analysis because for each chosen threshold, we would have a different feature lists and thus, different models. To avoid this subjectivity in our redundancy analysis, we introduced a more flexible way to use threshold for redundancy analysis. To do so, we defined a redundancy space $S = [0.5, 1.0]$ in which we vary different redundancy thresholds (hyper-parameters) with a step $t = 0.05$. $S = [0.5, 1.0]$ is chosen as redundancy space as it includes different thresholds that one can use to describe high correlation between features. This strategy is efficient and allows to vary the thresholds precisely in order to have several graduation of the selected redundancy level. Thus, using the list of non-protected and protected features from our input dataset, we sought to determine the lists of non-redundant (N) and redundant (R) features based on the thresholds space S by using symmetrical uncertainty as measure of correlation [Raileanu et Stoffel, 2004]. The formula for calculating symmetrical uncertainty (SU) as detailed in section 3.3.1 is defined by:

$$SU(X, Y) = 2\Big[\frac{IG(X, Y)}{H(X) + H(Y)}\Big] \tag{4.1}$$

We choose to use the SU as correlation measure for several reasons. Firstly, it produces a normalized result between 0 and 1 and observes not only linear correlations, but also non-linear relationships between features. Secondly, it compensates the bias of information gain towards features with more values and restricts its values between [0,1]. A value of 1 indicates a strong correlation, a value of 0 indicates that X and Y are independent.

4.3.3 Model Evaluation

Once the list of non-redundant features (N) and the list of redundant features (R) are obtained from 4.3.2, we then seek to train various models using all possible combinations between N and R. We start by training a model with N only then we add iteratively every combination of features of R until all the iterative combinations between N and R have been used. Like this, we have a list of models, each trained with a different combination of features (partition).

4.3.3.1 Computing F1-Score

F1-score is used as measure to assess performance of the learning models obtained in section 4.3.3. In a classification problem, f1-score, is used to find the balance between precision and recall. Precision is the fraction of true positive (TP) examples among the examples that the model classified as positive (TP:true positive, FP:false positive). Recall, also known as sensitivity, is the fraction of the number of all correct examples classified as positive (TP) out of all positive that could have been classified (TP, FN:false negative). Based on the definition of Precision and Recall, the f1-score can be written as:

$$F1 - score = \frac{2 * Precision * Recall}{Precision + Recall} = \frac{2 * TP}{2 * TP + FP + FN} \tag{4.2}$$

4.3.3.2 Computing Fairness

To compute fairness, we use two different bias metrics that have been proposed in the literature [Bellamy *et al.*, 2018]. In order to introduce the bias metrics that we used, we recall the basic settings for the dataset introduced in section 2.3.1. Let X be an input dataset modeled as a set of data annotated $X_{n,m} \in R^{n \times m}$ consisting of *n* individuals and *m* variables of a space representing the process. Using a matrix, we can then write our input data as:

$$X_{n,m} = \begin{pmatrix} x_{1,1} & \cdots & x_{1,m} \\ x_{2,1} & \cdots & x_{2,m} \\ \vdots & \ddots & \vdots \\ x_{n,1} & \cdots & x_{n,m} \end{pmatrix} = (x_{i,j})_{1 \leq i \leq n, 1 \leq j \leq m}$$

Each variable of $X_{n,m}$ is represented by a column vector annotated

$$x_{\bullet j} = (x_{i,j})_{1 \leq i \leq n} \in R^n$$

and each individual of $X_{n,m}$ is represented by a line vector annotated

$$x_{i\bullet} = (x_{i,j})_{1 \leq j \leq m} \in R^m$$

Let f be a learning model and its performance score $f[X]$ which will be used to predict a binary output $\hat{y} \in \{0, 1\}$. Each data point X_i is associated to a protected feature P, here we consider that P is binary: $P \in \{$ 0,1$\}$. We consider $P = 0$ to be an unprivileged group and $P = 1$ a privileged group. Likewise, we consider $\hat{y} = 1$ to be the preferred outcome, assuming it represents the more desirable of the two possible outcomes. For instance, P =‘gender’ could be the protected attribute with ‘female’ = 0, the unprivileged group, and ‘male’ = 1 the privileged.
Suppose for some data points we know the ground truth; i.e., the true value $y \in \{0, 1\}$. Note that these outcomes may be statistically different between different groups, either

because the differences are real, or because the model is somewhat biased. Depending on the situation, we may want our estimate $\hat{y}$ to take these differences into account or to compensate them. So, we used the two bias metrics introduced in [Bellamy *et al.*, 2018] and defined below:

1. **Demographic Parity**. This metric suggests that a predictor is unbiased if the prediction ŷ is independent of the protected feature P such that Pr($\hat{y} \mid P$) = Pr(ŷ) (Pr: Prediction rate). This means that the same proportion of each subgroup is classified as positive. To assess fairness from this metric, we use the difference between prediction rates of the subgroups. Let us call this difference Demographic Parity Difference (DPD), we defined it as:

$$DPD = Pr(\hat{y} = 1|P = 1) - Pr(\hat{y} = 1|P = 0) \tag{4.3}$$

2. **Equality of Odds**. This metric states that the prediction ŷ is conditionally independent of the protected feature P, given the true value y: Pr ($\hat{y}$ |y, P) = Pr ($\hat{y}$ |y). This means that the true positive rate and the false positive rate will be the same between unprivileged and privileged groups. To assess fairness from this metric, we use the difference between prediction rates (positive and negative). Let us call it Equality of Odds Difference (EOD), we defined it as:

$$EOD = Pr(\hat{y} = 1|P = 1, y = y_i) - Pr(\hat{y} = 1|P = 0, y = y_i), y_i \in \{0, 1\} \tag{4.4}$$

Using the measurements obtained with the two metrics defined above in formulas (4.3) and (4.4), we are ready to compute a fairness score. For these two metrics, the result obtained is between -1 and 1, however the ideal value we would like to obtain is 0. Since the domain of performance values (F1-score) is between 0 and 1, we use the absolute value of the measurement obtained in order to normalize the fairness domain between 0 and 1. Then, we invert the value obtained so that its greatest value is 1. For example, let $val \in$ [-1, 1] be the fairness value obtained for a metric, to invert it we proceed like this: new_val = 1-$|val|$.

For the partitions used (section 4.3.3), when the list N is used alone, i.e when there is no redundant feature, we consider that the fairness score is 1 since there are no protected features nor redundant to the protected ones. When we add partitions from R to N, the added redundant feature (from R) is used as P to assess fairness. If there are multiple redundant features in R to the protected, we calculate an intermediate fairness score for each of the redundant and then average it to obtain a final fairness score. With the other FS methods used for comparison in our experiment, if there is any protected feature (i.e. P), in their list of selected features, it is used to assess fairness using the formulas in 4.3 and (4.4). Otherwise, we consider their fairness score to be 1. Since we have two values

for fairness from the two bias metrics used, the final fairness value used is an average of the two fairness scores obtained:

$$Fairness = \frac{DPD + EOD}{2} \tag{4.5}$$

4.3.4 Computing the Trade-off (Delta)

We have defined our trade-off formula as follow:

$$\Delta = (1 + \beta^2) * \frac{\textit{F1-score} * \textit{Fairness}}{\beta^2 * \textit{F1-score} + \textit{Fairness}} \tag{4.6}$$

The formula is inspired from the traditional F-measure [Powers, 2020] and helps to compute the harmonic mean between our f1-score and fairness (fairness is the value obtained from 4.5. The reason we have chosen to use the harmonic mean instead of other means is that it allows us to weigh the fairness higher than the f1-score. With the classical arithmetic mean, the higher score would have been more important (sometimes it could be the f1-score and sometimes the fairness). But by choosing to do so, we decide to assign a greater importance to fairness (whether this score is smaller than the f1-score or not). In the formula defined for delta, two commonly used values for β are 1 which weighs fairness higher than f1-score, and 0.5, which weighs fairness lower than f1-score. In our experiments, the beta (β) in the formula is then set to 1 (β = 1). For each trained model in step 4.3.3, we obtain a f1-score and a fairness value. Then we compute a delta for each model using their $f1 - score$ and $fairness$. The final delta will be the max, which we note $Fmax$, of all the delta. The list of features according to $Fmax$ is noted as $Mmax$.

4.3.5 Algorithm of the Proposed Method

Algorithm 2 shows the process of the proposed method. Let N be the set of non-redundant features, we denote by f_i a feature of N. Let P the set of known protected features or designated by a system expert before any analysis, we denote by p_i any feature of P. The algorithm takes as input two lists: the list of non-protected features N and the list of protected features annotated P from the input dataset. We use the following defined parameters: t the step (t = 0.05) to iterate over S and the hyper-parameter space (S = [0.5, 1.0]). We start by initializing the values of $Fmax$, $Mmax$, d, t and the empty list R (line 2-6). For each hyper-parameter (threshold) d in S, we seek to find the redundancy between the list of protected P and the list of non-protected N, if any redundant feature is found according to d, it is added to R then removed from N (line 7-16). Using N, we iteratively increment over all possible partitions cr of R to train all possible models using all the partitions between the lists N and R, evaluate them and calculate their delta according to the specified formula(17-25). Then we decrement, start over using a new

value of d until all possible value in the hyper-parameter S have been used (line 27). For the output of the algorithm, we have **Fmax** which is the max of all the calculated delta, and **Mmax** which is the list of features constituting the model which led to the delta max (Fmax).

Algorithm 2: Pseudo-code of the proposed method

Input: N, **P** // Non protected and protected Features
Output: Fmax, Mmax //max performance & feature list
1 **Begin**
2 t ← 0.05 $//iteration\ step\ over\ S$
3 d ← 1.0 $//highest\ threshold\ value\ in\ S$
4 R ← { } $//redundant\ list$
5 Fmax = 0 $//max(\Delta)\ to\ maximize$
6 Mmax ← { } $//feature\ list\ of\ Fmax$
7 **while** d ∈ S **do**
8 $//finding\ redundant\ features$
9 **for** $f_i \in$ N **do**
10 **for** $p_i \in$ P **do**
11 **if** |compute corr(p_i, f_i)| ≥ d **then**
12 R ← $R \cup \{f_i\}$
13 N ← N $\setminus \{f_i\}$
14 **end if**
15 **end do**
16 **end do**
17 $//search\ for\ the\ best\ model\ with\ the\ best\ trade\text{-}of\ \hat{f}$
18 **for** $cr \in$ partition(R) **do**
19 compute $\hat{f}$ using N $\cup\ cr$
20 compute Δ using eq. 4.6
21 **if** Δ ≥ Fmax **then**
22 Fmax ← Δ
23 Mmax ← $\hat{f}$
24 **end if**
25 **end do**
26 d ← d - t
27 **end do**
28 **End**

4.4 Experimental Validation

In this section, we present the experimental approach that we carried out and the comparative results obtained. The goal of these experiments is to compare the results obtained with our method with other feature selection methods. Two other existing feature selection methods were used for comparison: mRMR [Peng *et al.*, 2005] and FCBF [Yu et Liu, 2004]. In particular, this comparison was made based on three criterion (performance, fairness and the trade-off score) using a classification task. We have also compared the numbers of selected features by each method and the execution time.

Baseline.

- The first existing method used for comparison with our proposed method is mRMR [Peng *et al.*, 2005]. It aims to select a subset of highly relevant features while reducing redundancy between themselves. This method is a two stage process; a stage includes an incremental feature selection which later is combined to another more sophisticated wrapper feature selectors. We used the backward and forward selectors as in the original paper.

- FCBF [Yu et Liu, 2004] is the other existing method used for comparison to our proposed one. It uses symmetrical uncertainty as correlation measure and approximate Markov blanket to remove redundancy [Wang *et al.*, 2020]. This method requires the user to set up a threshold and this is somehow subjective. However, the authors stated in their paper that setting the threshold to a reasonably large value does not sacrifice the goodness of the selected subsets, thus in our experiment, the threshold was set to 0.6.

When used, each above method outputs a final set of relevant features, normally this represents a "partition" (a subset of features) of our proposed method. Then we use this final relevant list from each method to train and evaluate a model (4.3.3) and compute its trade-off score (4.3.4).

Classifiers.

To evaluate and compare the proposed method to existing methods, we proceeded to a learning task by considering a binary classification problem over the four datasets that we describe below (section 4.4.1). For this binary classification problem, Random Forest [Pal, 2005] and AdaBoost [Schapire, 2013] (detailed in Annex A.1.1) were used as classifiers. This choice is explained by the main advantages of these classifiers which ensures high precision through cross validation, providing an easy interpretation of the obtained result. And also the fact that these two classifiers use two different ensemble strategies (bagging and boosting), this allowed us to seize different aspects of each method on the learning task. Each model is trained and evaluated using the classic cross-validation procedure. F1 Score is used as measure to assess the performance of each trained model.

4.4.1 Datasets

To evaluate our method, we carried out experiments on four well-known datasets in the literature [Dua et Graff, 2017]. They each contain known protected features, which allowed us to evaluate our method on appropriate cases. These datasets were chosen on the basis of the differences they present, their types and the number of observations varying from 615 to 32561 and the number of features.

- **German Credit Dataset.** The German credit dataset classifies people described by a set of features as good or bad credit risks. This dataset contains 1000 observations with 20 features. In this dataset, each entry represents a person who takes a credit by a bank. Each person is classified as good or bad credit risks according to the set of features.
- **Adult Income Dataset.** The Adult income dataset task it to predict whether income exceeds "50K/yr" based on census data. It is a widely quoted public dataset in machine learning literature and is used for introducing supervised machine learning algorithms for binary classification. The dataset contains 32,561 rows with 14 features of census data pertaining to adult income. The prediction task is to determine whether a person makes over 50K income in a year.
- **Bank Churn.** The goal in Bank Churn dataset is to predict customer churn in a bank. This data set contains details of a bank's customers and the target variable is a binary variable reflecting the fact whether a customer left the bank (closed the account) or continues to be a customer.
- **Loan Approval.** The last dataset used is Loan Approval. The goal is to automate the loan eligibility process based on customer detail provided while filling online application form.

Based on the general data protection regulation (GDPR) policy, we have identified which feature is protected in each dataset. In Table 4.1, we give more details on the datasets used.

Table 4.1: **Experimental datasets used**

Dataset	Observations	Features	Protected
Adult Income	32561	15	2
Bank Churn	10147	13	1
German Credit Scoring	1000	9	1
Loan Approval	615	14	2

4.4.2 Results Analysis

The analysis of the results is based on two criterion: the number of selected features and the trade-off score (fairness and performance).

4.4.2.1 Selected Features

We present in Table 4.2 below the first comparison results obtained with the different methods over selected features.

Table 4.2: **Comparison over selected features**

	Our Method				**mRMR**				**FCBF**			
Data	**N**	**P**	**R**	**(R** $\in \hat{f}$**)**	**N**	**P**	**(R)**	**(R** $\in \hat{f}$**)**	**(N)**	**P**	**R**	**(R** $\in \hat{f}$**)**
German	7	1	1	1	5	1	0	0	6	1	0	0
Adult	12	2	3	2	7	2	1	1	9	1	1	1
Bank	11	1	2	1	8	0	0	0	7	0	2	2
Loan	11	2	3	1	6	1	1	1	8	1	0	0

In the Table 4.2, for our proposed method: **(N)** represents the number of selected features, **(P)** the number of protected features, **(R)** the number of redundant features detected and **(R** $\in \hat{f}$**)** the number of redundant features that is part of the list used for the final model. For the two other existing methods: **(N)** represents the number of selected features, **(P)** the number of protected features that we observed in their selected list, **(R)** the number of redundant features observed in their selected list and **(R** $\in \hat{f}$**)** the number of redundant features that is part of the list used for the final model.

We notice that the number of selected features of our method is higher than the other approaches. This is due to our fairness goal without removing any feature beforehand while the others remove some features based on their relevancy and redundancy approaches. We also recall that our method, in order to find redundant features, uses protected feature.

Observing the results, we have used two protected features for Adult and Loan Approval dataset while for German Credit and Bank Churner dataset we have used one protected feature. For our method, the protected features are used in the redundancy analysis step only, they help finding the redundant features using the threshold space (S = [0.5, 1.0]).

When applying the two other existing methods (mRMR and FCBF), we have observed that their final lists contains protected features **(P)** and also features that have been

highlighted as redundant **(R)** by our method. This is explained by the fact that theses methods do not take care or propose any processing to handle protected features which is not the case for for our method. Here comes one of the contribution of our method on not using personal protected or sensitive data with respect to data privacy policy.

4.4.2.2 Trade-off score (F1-score, Fairness)

For every classifier used, we report the comparative performance for all the datasets for the three methods of the basline. On Tables 4.3 and 4.4 below, we report the results based on the trade-off score (f1-score, fairness), obtained by our method compared to the two other FS methods.

The experiments were carried out using two classifiers: i) Random Forest and ii) AdaBoost, on four well known datasets for fairness study. F1-score was used to assess performance of each classifier. We compute fairness based on three different bias metrics (Demographic Parity, Equal Opportunity, Equality of Odds). The score for Delta is calculated using equation 4.6 with the process explained section 4.3.4.

- **Result with Random Forest**
  Figures 4.2, 4.3 and 4.3 show the results obtained with Random Forest.

  – For performance score, we report on Fig. 4.2. The performance results using f1-score show that our method outperforms the baseline on all the four datasets. FCBF achieved the lowest score on average.

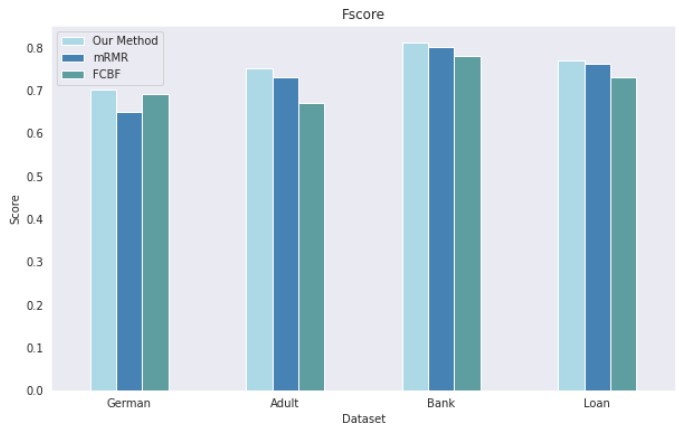

Figure 4.2: Random Forest: f1-score for all the datasets

  – For fairness score reported on Fig. 4.3, the proposed method has a better score than the baseline on three datasets. However, for Bank dataset, the other methods in the baseline have a better fairness score than our method.

This is explained by the fact we had to set their fairness to 1 because there was no protected feature in their lists of selected feature.

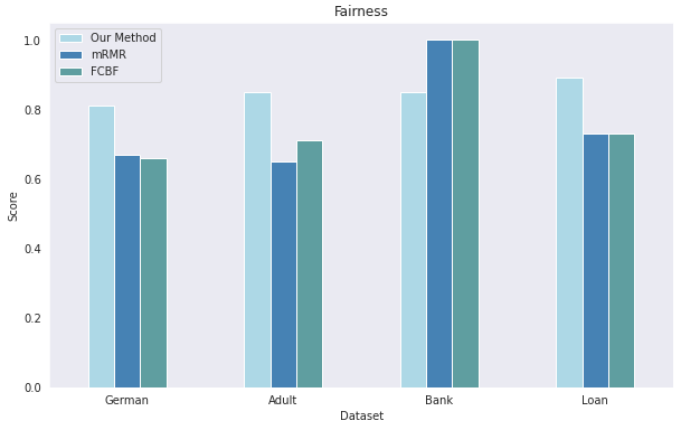

Figure 4.3: Random Forest: fairness score for all the datasets

– Fig. 4.4 reports on the trade-off score. We notice that mRMR and FCB have surpassed our method on Bank dataset. Beside this dataset, the proposed method achieved a better performance for the trade-off score than the baseline on all the other datasets.

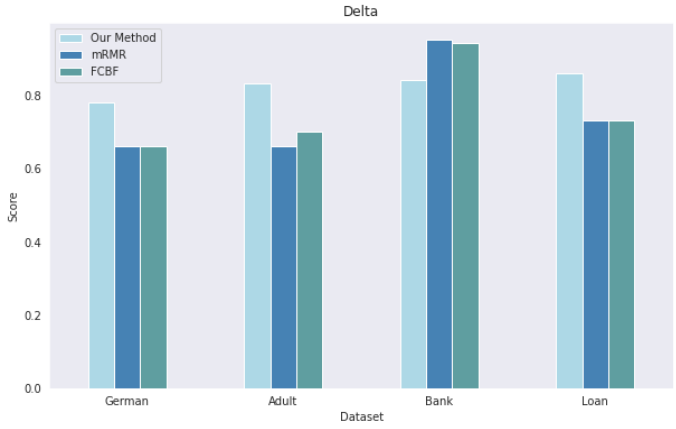

Figure 4.4: Random Forest: trade-off (Delta) score for all the datasets

We resume in Table 4.3 the comparison between our proposed method and the baseline. The results show that our method outperforms the two other existing methods on three datasets, except Bank where the mRMR and FCBF have a higher score (0.88 and 0.87) for Delta than our method (0.83). However, this

understandable because, since there was no protected feature in their final list of selected feature, we had to set their fairness score to 1.

Table 4.3: **Random Forest: Comparison based on performance, fairness and delta; *F1*: F1-score, Fs: Fairness, Dt: Delta**

	Proposed Method			**mRMR Method**			**FCBF Method**		
Dataset	**F1**	**Fs**	**Dt**	**F1**	**Fs**	**Dt**	**F1**	**Fs**	**Dt**
German	**0.70**	**0.81**	**0.75**	0.65	0.67	0.66	0.69	0.66	0.67
Adult	**0.75**	**0.85**	**0.78**	0.73	0.65	0.68	0.67	0.71	0.68
Bank	**0.81**	0.85	0.83	0.80	**1**	**0.88**	0.78	**1**	**0.87**
Loan	**0. 77**	**0.89**	**0.82**	0.76	0.73	0.74	0.73	0.73	0.73

- **Result with AdaBoost**
  On Figures 4.5, 4.6, 4.7 and Table 4.4, we report the performance using AdaBoost for all the four datasets in terms of performance, fairness and the trade-off score. On each figure, we present the score for every dataset that we used.

  – We report on Fig. 4.5 results for performance score using f1-score. We notice particularly that, on German and Adult datasets, the two other methods in the baseline have slightly surpassed our method with a f1-score of 78% where our method achieved only 75%. However, our method achieved the highest score for Bank and Loan datasets than the baseline, except FCBF that achieved the same score of 84% on Loan dataset.

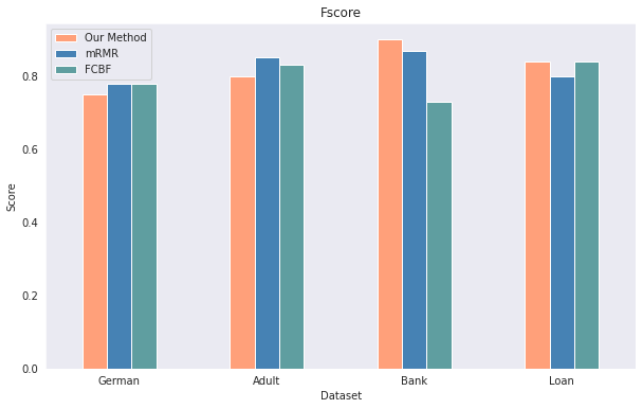

Figure 4.5: AdaBoost: f1-score for all the datasets

– Fig. 4.6 reports the fairness results obtained for all the four datasets. The proposed method has surpassed highly the baseline on three datasets. On Bank dataset, the baseline surpassed our method, again this is explained by the fact we add to set their fairness score to 1 prior applying the formula.

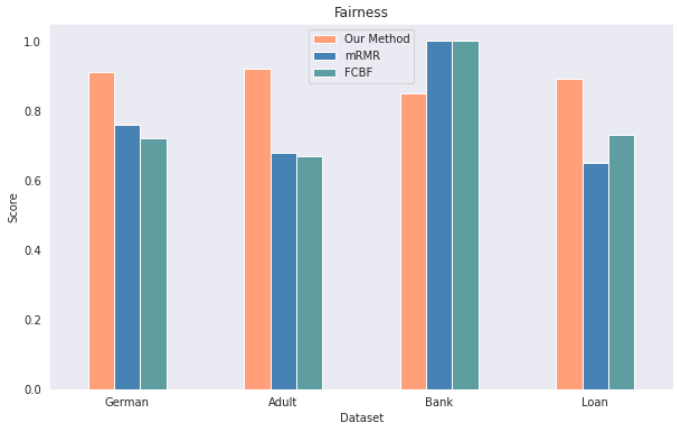

Figure 4.6: AdaBoost: fairness score for all the datasets

– Results for the trade-off score is reported on Fig. 4.7. For German, Adult and Loan datasets, the proposed method has achieved the highest score comparing to the baseline. mRMR surpassed slightly our method on Bank dataset, 93% to 90%, however, we notice that even with a fairness score of 1 for FCBF, our method has achieved a better score for the trade-off than this method.

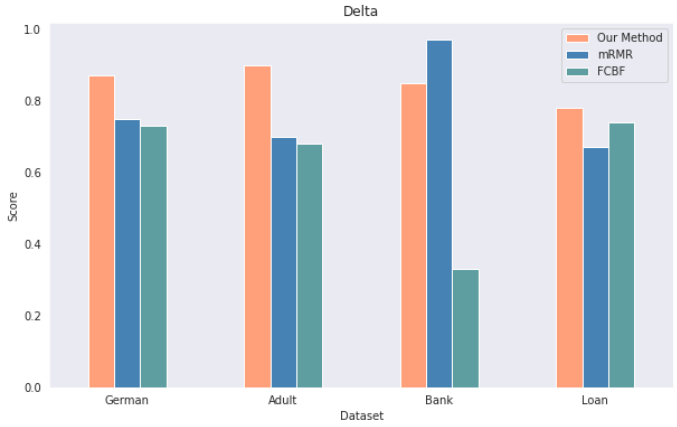

Figure 4.7: AdaBoost: trade-off score for all the datasets

Table 4.4 recalls the comparative results for the experiments using AdaBoost. We clearly notice that our method has higher Delta score than the other methods

on three datasets, except Bank dataset where mRMR performed better than the proposed. Particularly, we notice that mRMR and FCBF achieved the highest f1-score (78%) on the German dataset, but still our method achieved a higher trade-off score. We notice the same observation for the Adult dataset.

We also notice that FCBF has a higher fairness score (100%) than our method (92%) on Bank dataset, but again, our method achieved a higher score for the trade-off since our performance score is better.

Table 4.4: **AdaBoost: Comparison based on performance, fairness and delta; *F1*: F1-score, Fs: Fairness, Dt: Delta**

	Proposed Method			**mRMR Method**			**FCBF Method**		
Dataset	**F1**	**Fs**	**Dt**	**F1**	**Fs**	**Dt**	**F1**	**Fs**	**Dt**
German	0.75	**0.91**	**0.82**	**0.78**	0.76	0.77	**0.78**	0.72	0.74
Adult	0.80	**0.92**	**0.85**	**0.85**	0.68	0.75	0.83	0.67	0.74
Bank	**0.90**	0.92	0.91	0.87	**1**	**0.93**	0.73	**1**	0.84
Loan	**0.84**	**0.89**	**0.86**	0.80	0.65	0.71	**0.84**	0. 73	0.78

Overall, the results of the experiments show that our method performs well.

4.4.2.3 Execution Time

We also compared the execution time of our algorithm to other FS methods (Figure 4.8). The FCBF method is faster than our method over all the datasets. It is understandable since this method performs a single redundancy analysis using only one defined threshold while our method uses a threshold space to perform redundancy analysis. However, our method is faster than the mRMR method which is in fact is a two stage process with two wrapper selectors.

In general, on these four datasets, we get satisfactory results and we have maintained a good level of performance (F1-score), a higher fairness guarantying a higher score for the trade-off between f1 score and fairness.

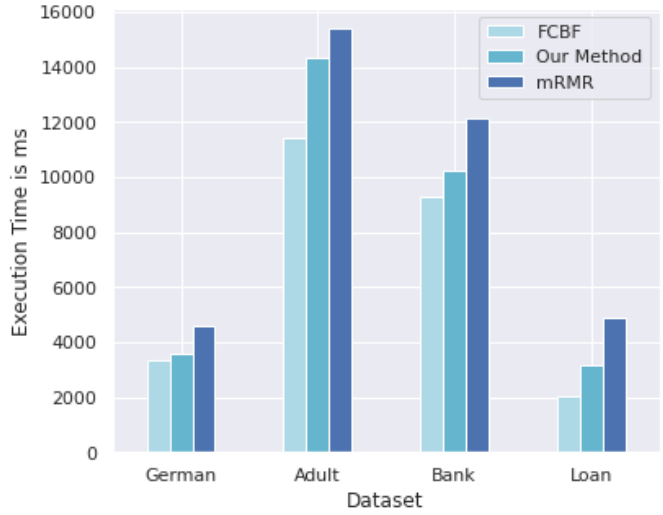

Figure 4.8: Comparison of execution time in milliseconds

4.5 Summary

In this chapter, we present a novel feature selection method to improve performance and fairness in the case of protected features while considering their redundant. To achieve our goal, we introduce a trade-off strategy between performance and fairness. Unlike existing methods, the proposed method allows to obtain a model that is both optimal and fair by considering protected and redundant features with respect to data privacy policy.

The performance of our method was experimentally evaluated on four well known biased datasets. Compared to three other feature selection methods, we obtain satisfactory results. The comparative results obtained show our method's effectiveness in boosting fairness while maintaining a high level of performance.

Nevertheless, our method has some limits regarding small sized datasets which impact performance loss. In order to overcome these limits, we present in the next chapter an improvement of this method. This improved contribution focuses on data imbalance regarding protected and redundant features and sort out the imbalance that led to bias.

The full source code, data and results for this contribution can be accessed under request only via the thesis's repository on GitHub here (Contrib4).

Chapter 5

Achieving Fairness in ML Models with Regard to Protected Feature and Imbalanced Data

Contents

5.1 Introduction

Nowadays, decision-making systems based on machine learning algorithms are becoming more and more automated by helping human judgment with algorithmic decisions that are largely based on data. However, concerns have been raised [Yeom *et al.*, 2018] that machine learning algorithms may lead to unfair decisions against certain groups characterised by sensitives or protected features such as gender, race, religion.

Basically, two majors problems were identified [Ristanoski *et al.*, 2013, Chawla *et al.*, 2004] as the main cause of the unfairness in automated decision-making: the uncontrolled use of protected/sensitive features and the used of imbalanced datasets [Kotsiantis *et al.*, 2006]. Protected or sensitive features, according to [Fang *et al.*, 2020], are features that are of particular importance either for social, ethical or legal reasons when making decisions. According to [Chawla *et al.*, 2004], a dataset suffers from class imbalance when there is significant or extreme disproportion between the number of examples of each class in the dataset. By class in the dataset we mean, in the context of supervised machine learning and with a classification task in particular, the label or output we want to predict based on a set of inputs values. Based on a protected feature such as $gender$, a privileged group (male for example) would be more likely to receive an advantageous treatment than the unprivileged group (here, female for example). Such a behavior is not only undesirable but may have serious impact on the unprivileged group [Romei et Ruggieri, 2013].

To this end, many machine learning approaches have been proposed to help improving fairness in decision-making systems that are based on machine learning algorithms. These approaches are generally classified into three categories: pre-processing, in-processing, and post-processing [Caton et Haas, 2020].

Some of the proposed machine learning approaches [Dwork *et al.*, 2012] for fairness improvement with regard to protected features tend to remove them prior the learning model in order to obtain a fair outcome. However, while this strategy may work, we found that it is limited and can lead to a significant performance loss in the case where protected features are relevant for the learning task. Some other approaches [Chawla *et al.*, 2003, Hu *et al.*, 2009] to improve fairness also tend to focus on maintaining an overall accuracy for both privileged and unprivileged. Again, we noticed that this strategy may not always work when using data that suffers from class imbalance. It has been proved [Gu *et al.*, 2009, Zhuang et Dai, 2006] that overall accuracy is not always a good performance indicator when using unbalanced dataset since it tends to favor the majority group over the minority.

Since most of fairness-related datasets suffer from class imbalance, addressing fairness with regards to protected features in machine learning algorithm also requires addressing the issue of imbalanced dataset. Thus, in our work, we were focused on these two issues, the use of protected features and class imbalance, that directly impact performance and fairness of machine learning algorithms. To this end, we propose FAPFID: A Fairness-aware Approach for Protected Feature and Imbalanced Data. Our

method allows to handle protected feature and class imbalance while ensuring an efficient and fair model for decision-making involving machine learning algorithms. Using the input dataset, our method creates a set of balanced and stable clusters while ensuring that both privileged and unprivileged groups are fairly represented in each cluster. Then an ensemble learning model is built upon the aggregated balanced and stable clusters which allow to obtain a cumulative and fair model. This work is an improved version of our previous contribution presented in Chapter 4 and the contributions here can be summarized as follows:

- We define a cumulative-fairness approach for dealing with protected features in decision support, it is tested on a binary classification task using an ensemble learning strategy.
- The proposed approach, FAPFID, is based on stable and balanced clusters, thus we propose a clustering stability algorithm to this end.
- FAPFID takes into consideration protected features and class imbalance while making fair decisions, so that the balanced-accuracy score remains high.
- To achieve fairness, FAPFID is based on Equalized Odds as fairness metric and it being tested on three real-world dataset suitable for fairness study and is easily adaptable to any social decision problems with regards to protected features and class imbalance.

The rest of this chapter is organized as follows: in section 5.2, we summarize the different existing methods to tackle the issues identified with their limitations. In section 5.3, we introduce some basics concepts and definitions. We present our new approach in section 5.4. The experimental results are described and analyzed in section 5.5. Chapter summary and perspectives are presented in section 5.6.

5.2 Related Work

Many existing work proposed various machine learning methods to deal with fairness issues related to the use of protected features and imbalanced data [del Barrio *et al.*, 2020]. Here we look at those existing methods under these two categories and we also look at what previous work has defined in terms of fairness metrics.

Many definitions of fairness [Chouldechova et Roth, 2018, Verma et Rubin, 2018, Friedler *et al.*, 2019] have been proposed over the recent years. Most of the recent proposed methods use fairness definitions such as demographic parity [Jiang *et al.*, 2021, Singh et Joachims, 2018, Wadsworth *et al.*, 2018]. This fairness metric suggests that a predictor is unbiased if the prediction ($\hat{y}$) is independent of the protected feature such that positive prediction rate between the two subgroups are the same. Other proposed methods

have instead used other fairness metric such as equalized odds [Ghassami *et al.*, 2018, Mishler *et al.*, 2021, Pleiss *et al.*, 2017]. Unlike demographic parity, this fairness metric instead suggests that the true positive rate and the false positive rate will be the same for both unprivileged and privileged groups. However while each of these definition has merit, there is no consensus on which one is consequently the best, and this issue is beyond the scope of this chapter. Our goal is not to address the relative virtues of these definitions of fairness, but rather to assess the strength of the evidence presented by a set of subgroup that a model is unfair to a certain group based on a given metric and the best possible trade-off between fairness and performance

For proposed methods that deal with fairness related to protected features, we notice several approaches [Lahoti *et al.*, 2020, Martínez *et al.*, 2021]. Particularly, we notice the work in [Dwork *et al.*, 2012] where authors introduced a first approach, named "Fairness Through Unawareness", consisting of removing completely all protected features of the dataset to ensure fairness. However, we notice that this approach may not solve the problem because there may be redundant features or even proxies to the protected. As underlined by [Yeom *et al.*, 2018], some features known as proxies such as zip code, for example, can reveal the predominant race of a residential area. Thus, this can still lead to racial discrimination in a decision making problem such as loan application despite the fact that zip code appears to be a non-protected feature. We also notice the work of [Iosifidis *et al.*, 2019] where authors introduced a framework that combines pre-processing balancing strategy with post-processing decision boundary adjustment in order to deal with fairness related to protected features and class imbalance. In the pre-processing strategy, they created local subgroups where they performed random under-sampling technique to guarantee equitable representation between minority and majority groups. While this strategy may work on large datasets with thousands of instances, we notice that it suffers from a performance loss when used on a restricted dataset.

Given the limitations of the above approaches, there is a need for more in-depth research to overcome these limitations. Thus, we propose a new fairness-aware strategy that allows the obtaining of an efficient and fair models with regards to protected features and unbalanced data. We would like to recall here, as part of our approach, a given model is said to be "fair", or "equitable", if its results are independent of one or more given features, in particular those considered to be protected [Oneto et Chiappa, 2020, Ji *et al.*, 2020].

5.3 Basic Concepts and Definitions

Before going into more details, Here, we would like to recall the basic concepts that we will use throughout this chapter. We consider an input dataset $S = \left(X_{m,n}, Y_{1,n}\right)$ con-

sisting of n observations and m features. Let f be a learning model and its performance score $f[S]$ which will be used to predict a binary output $\hat{y} \in \{0, 1\}$. Each sample $x_{\bullet i}$ is associated to a protected feature P, for simplicity we consider that P is binary: $P \in \{P_0, P_1\}$. We consider P_0 to be an unprivileged group and P_1 a privileged group. Likewise, we consider $\hat{y} = 1$ to be the preferred outcome, assuming it represents the more desirable of the two possible outcomes. For instance, P ='gender' could be the protected attribute with P_0 ='female', the unprivileged group, and P_1='male' the privileged one.
Suppose for some samples we know the ground truth; i.e., the true value $y \in \{0, 1\}$. Note that these outcomes may be statistically different between different groups, either because the differences are real, or because the model is somewhat biased. Depending on the situation, we may want our estimate $\hat{y}$ to take these differences into account or to compensate them.*

5.3.1 Fairness Metric

In this work, we have used Equalized Odds (EqOd) as fairness metric since it is widely used and adopted by recent state-of-the-art methods [Mary *et al.*, 2019, Salazar *et al.*, 2021, Iosifidis et Ntoutsi, 2019]. EqOd measures the difference of true classified examples between privileged and unprivileged group in all classes [Bellamy *et al.*, 2018]. That being said, prediction $\hat{y}$ is conditionally independent of the protected feature P, given the true value $y : Pr(\hat{y} \mid y, P) = Pr(\hat{y} \mid y)$. This means that the true positive rate and the false positive rate will be the same between the privileged and unprivileged groups. To compute the difference between classified instances of the two groups, EqOd is defined as follow:

$$EqOd = Pr(\hat{y} = 1 \mid P_1, y = y_i) - Pr(\hat{y} = 1 \mid P_0, y = y_i), y_i \in \{0, 1\} \tag{5.1}$$

According to this metric, a method is fair if the value of EqOd is between $[-0.1, 0.1]$. The ideal value of this metric is 0. A value < 0 implies higher benefit for the privileged group and a value > 0 implies higher benefit for the unprivileged group.

5.3.2 Ensemble Learning method

Ensemble Learning helps improving machine learning results by combining several intermediate models. This approach allows the production of better predictive performance compared to a single model. For our ensemble learning strategy, we will use the Bagging method (A.1.1.4). Also known as bootstrap aggregating, Bagging is the aggregation of multiple versions of a predicted model. Each model is trained individually upon a subset, and combined using a majority voting process. Thus, we believe using an ensemble learning is an efficient technique to tackle imbalanced ratio towards protected feature as it divides the learning problem into multiple sub-problems and then combines their

solutions (local models) into an final model. Intuitively, we found it easier to tackle the problem related to fairness in the subset with locals models rather than in a single and global model.

5.4 The FAPFID Approach

We depict in this section our approach to achieve fairness as illustrated in Fig. 5.1. It works as follows: first the input data is divided into K stable clusters by a clustering strategy [El Malki *et al.*, 2020]; then we ensure that obtained clusters are balanced with respect to the protected feature in each cluster. In the case where some clusters are unbalanced, we apply an oversampling technique, SMOTE [Chawla *et al.*, 2002]. Then a final set of balanced clusters is constructed. The final ensemble is then divided into bags where we apply an ensemble learning strategy, Bagging. A learner is trained on each bag and then a final model is obtained by majority vote. Below we describe each step.

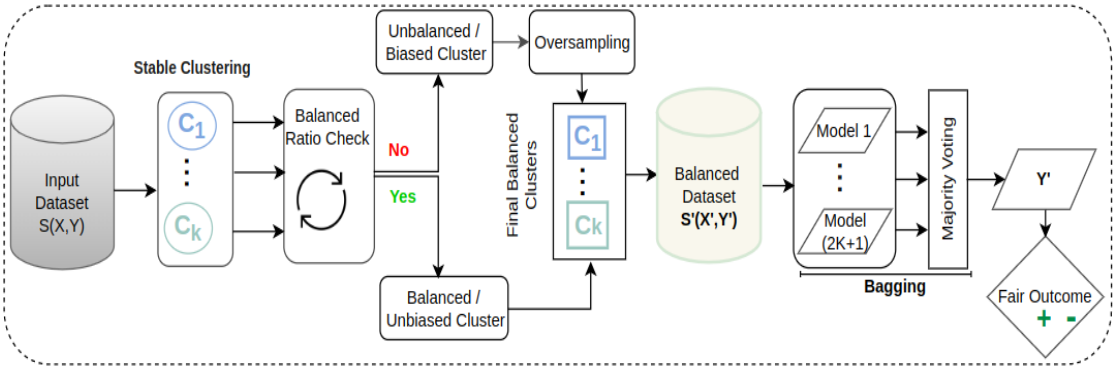

Figure 5.1: The FAPFID approach with different steps

5.4.1 Stable Clustering

In this step, we use a strategy to ensure that the number of clusters that we obtain are stable, i.e optimal clusters according to [von Luxburg, 2010] and [Jing, 2001]. For this, we define a stability strategy to boost our clustering solution.

5.4.1.1 Why using clusters ?

In addition of its cost-effectiveness, using clusters help reflecting the variance of the population of the dataset. Each cluster is a mini population and represents a mirror of the total population and of each other. We use clusters instead of other grouping strategies

because they guarantee a better homogeneity between clusters and heterogeneity within the clusters [Dash et Liu, 2000]. This is helpful in reducing bias towards different subgroups.

5.4.1.2 Why stable clusters ?

Obtaining stable clusters is useful to maintain a great performance hence ensuring a reliable fairness. We use stable clusters because they ensure that the instances are truly in their respective clusters. Thus, we establish a clustering stability strategy to avoid that wrongly clustered instances impact the balancing strategy that we later perform. In order to guarantee the obtaining of stable clusters, we define a statistical setup. Our stability strategy aims to provide information on the variation of instances for different values of k between two clusterings solutions of two sub-samples of the same dataset. Thus, for each value of k, we seek to obtain a stability rate by looking at the percentage of instances, points or pairs of points on which the two clusterings agree or disagree. The value of k whose instances variation percentage between the two clusterings is closer to zero will be the one that guarantees the best stability, and therefore the optimal value of k to choose.

5.4.1.3 Stability Strategy

Here we define our clustering stability approach. We consider a generic clustering algorithm such as K-means that receives as input a dataset $S = \left(X_{m,n}, Y_{1,n}\right)$ and an additional parameter K. It then assigns clusters to all samples of S. The dataset S is assumed to consist of n samples $x_{\bullet i}$, . . ., $x_{\bullet n}$ that have been drawn independently from a probability distribution T on some space X.
Assume we agree on a way to compute distances $d(C, C')$ between clusterings C and C'. Then, for a fixed probability distribution T, a fixed number K of clusters and a fixed sample size n, the stability of the clustering algorithm is defined as the expected distance between two clusterings $C_K(S_z), C_K(S'_z)$ on different datasets S_z, S'_z of size z with z_{max} the highest number of samples, that is:

$$C_{stab}(K, z) = d(C_K(S_z), C_K(S'_z)) \tag{5.2}$$

The algorithm 3 below shows how we performed the stability analysis. On line (8), since the two clusterings are defined on the same samples, then it is straightforward to compute a distance score between these clusterings using any of the well-known clustering distances such as the Rand index, Jaccard index, Hamming distance, Variation of Information distance [Meil, 2003].

All these distances estimate, in some way or the other, the percentage of points or pairs of points on which the two clusterings C_z and $C_{z'}$ agree or disagree. In our experi-

ments, we chose to use the Jaccard Index Distance[Shameem et Ferdous, 2009] inside our stability analysis because it offers more information about the cluster's consistency.

Jaccard Index Similarity. The Jaccard similarity is a measure of how close two clusters, $C_z, C_{z'}$ are. The closer the clusters are, the higher the Jaccard similarity. We can associate an actual distance measure to it, which is called the Jaccard distance. The Jaccard similarity of two clusters C_z and $C_{z'}$ is given by:

$$SIM(C_z, C_{z'}) = \frac{C_z, \cap C_{z'}}{C_z \cup C_{z'}} \tag{5.3}$$

The Jaccard distance $d(C_z, C_{z'})$ is then given by equation 5.4 and, it equals 1 minus the ratio of the sizes of the intersection and the union of the clusters C_z and $C_{z'}$.

$$d(C_z, C_{z'}) = 1 - SIM(C_z, C_{z'}) \tag{5.4}$$

Algorithm 3: Clustering stability algorithm

Input: dataset **S**, a clustering algorithm **A**, k_{max} clusters and z_{max} samples.
Output: Optimal value of **K**

Begin
for k = 2 ... k_{max}:
    Generate z_{max} sub-samples S_z (z = 1, ..., z_{max}) of S
    for z = 1 ... z_{max}:
        Split S_z into k clusters C_z using A
    end for
    for z, z' = 1 ... z_{max} :
        Compute pairwise distance $d(C_z, C_{z'})$ using Jaccard index distance (4)
        Compute stability as the mean distance between clustering C_z and $C_{z'}$ as: $C_{stab}(k, z_{max}) = \frac{1}{z_{max}^2} \sum_{z,z'=1}^{z_{max}} d\left(C_z, C_{z'}\right)$
        Choose the parameter K with highest C_{stab}
    end for
end for
End

5.4.2 Balanced Check Ratio

The main goal here is to divide the clusters into balanced and unbalanced clusters. We compute the ratio rp, $\left(rp = \#P_1/\#P_0\right)$, between privileged and unprivileged instances for each cluster. Clusters with ratio $rp \neq 1$ are considered to be biased thus are sent to the oversampling stage to be oversampled using SMOTE [Chawla *et al.*, 2002]. We

qualify these clusters as biased by the fact that the ratio $rp \neq 1$ reflects a group imbalance and what authors in [Mehrabi *et al.*, 2019] have called population bias. We apply the SMOTE strategy in a different way of what have being used. Hence, SMOTE in our approach is only applied to protected features label, that means our clusters are balanced towards the unprivileged and privileged groups only and not the class label.

Why using SMOTE on subgroups only ?

In a binary classification problem, SMOTE [Chawla *et al.*, 2002] is used to generate synthetic samples for minority class data points in order to equalize the two classes. Suppose a data point from minority class is denoted as X where $x_1, x_2, ..., x_m$ are the features. Suppose for this data point X, it exists a nearest neighbor X' whose features are $x'_1, x'_2, ..., x'_m$. Using SMOTE, a new data point X_{new} will be generated between X and X' that is:

$$X_{new} = X + rand(0, 1) * (X - X') \tag{5.5}$$

Indeed, SMOTE creates synthetic data points with regards to class labels but ignores the imbalance that exists between subgroups in the dataset.

Figure. 5.2 represents the class imbalance before applying SMOTE. As we can see, there is a huge imbalance between the two classes. On Figure 5.3, SMOTE definitely balances the two classes but it increases the imbalanced ratio between privileged and unprivileged groups (Fig. 5.3). That being said, considering the traditional application of SMOTE, class balancing methods using SMOTE will improve their overall accuracy while worsen fairness between privileged and unprivileged groups. This happens because SMOTE synthetically generates new data points just to equalize the two classes in terms of labels with no considerations whatsoever to protected features within subgroups. This is where the novelty of our approach lies, instead of balancing the two classes, we create balanced subgroups instead with regard to protected features. Once the unbalanced clusters are oversampled, we construct a set of final balanced clusters that are therefore aggregated into a final set from which bags will be created to fit different classification models.

5.4.3 Bagging

Estimating the number of bags b must be sufficient to construct enough learners, since we consider each bag as a sample of the training data. To ensure that all the clustered instances are at least in one of the bags, we estimate the number of bags b as: $b = 2K + 1$, K is the number of stable clusters obtained in 5.4.1.3 with Algorithm 3. Since we will consider a classification task, the final model will be chosen by a majority voting strategy.

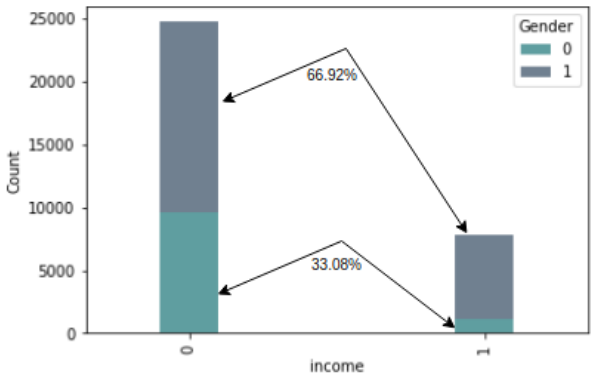

Figure 5.2: Adult Income: Group imbalance before applying SMOTE

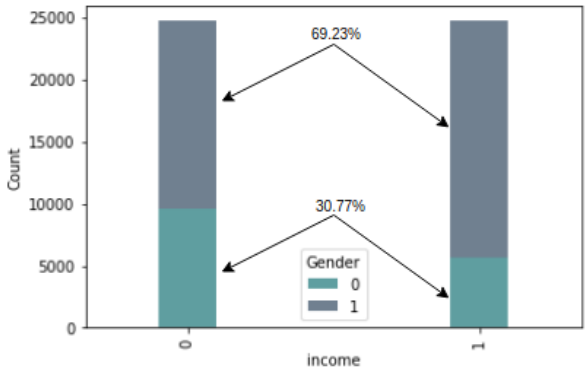

Figure 5.3: Adult Income: Group imbalance after applying SMOTE, class labels are balanced but group imbalance increases

5.4.4 Algorithm of the proposed method

Using the basic concepts that we previously defined in section 5.3, the algorithm 4 defined below takes as inputs a clustering algorithm A, a set of samples S, K number of clusters, privileged group P_1, unprivileged group P_0 and a base classifier G. We start by initializing an empty set of balanced clusters M (1) which later will contains the final balanced clusters as explained in 5.4.2. Then split S into K clusters using A to obtain C_i, $i = 1...K$ (2). For each C_i cluster, we compute the imbalance ratio between privileged group P_1 and unprivileged group (P_0) of clusters C_i. If the computed ratio is equal to 1, we add the current cluster C_i to M (3-5),that means this cluster is balanced toward privileged and unprivileged group. However, if the computed ratio is not equal to 1, we oversample the current cluster C_i using SMOTE [Chawla *et al.*, 2002] to obtain a balanced cluster C_i^{bal}. We add this balanced cluster C_i^{bal} to the final set M then we start over using a different value of K(6-10).

One we have used all the values of K and obtain our final list of balanced cluster M, we create a balanced dataset X' from M (11). Then, we create b, $(b = 2 * K + 1)$, number of bags from X'. For each bag X'_j extracted from X', we fit it using the base classifier G (12-14). The final output ensemble model E is obtained by a majority vote over G_j (16). After obtaining the final ensemble model E, we then compute the performance scores based on accuracy and balanced-accuracy, and we compute the fairness score using Equalized of odds (EqOd).

Algorithm 4: Algorithm of the proposed method

Input: a clustering algorithm **A**, **S** samples, **K** number of clusters, privileged group P_1 & P_0, a base classifier **G**
Output: Ensemble Model **E**

1 **Begin**
2 $M \leftarrow \{\ \}$ $//final\ set\ of\ balanced\ cluster$
3 Split **S** into **K** clusters C_i, $i = 1...K$ using **A**
4 **for** $i = 1...K$:
5 **if** Ratio $C_i(P_1)/C_i(P_0)$ = 1:
6 M $\leftarrow$ M $\cup$ $\{C_i\}$
7 **else**
8 $C_i^{bal} \leftarrow$ SMOTE(C_i)
9 M $\leftarrow$ M $\cup$ $\{C_i^{bal}\}$
10 **end if**
11 **end for**
12 Create X' from M
13 **for** $j = 1...2K + 1$:
14 Extract boostrap sample X'_j from X'
15 Fit $G_j(X'_j)$
16 **end for**
17 Output **E** : ensemble model of G_j
18 **End**

5.5 Experiment & Results

In this section we give more details on the experimental approach, the dataset used, baseline and results. We carried out an experimental approach with three goals. First, we compare our method to existing methods of fairness [Iosifidis et Ntoutsi, 2019, Iosifidis *et al.*, 2019, Chawla *et al.*, 2003] and secondly, we aim to assess the impacts of imbalance ratio between P_0 and P_1 on the performance of our method (section 5.5.3.3).

In particular, for the first goal, the comparison was made based on two criterion: performance and fairness score. For performance score, we have used Accuracy and Balanced-Accuracy. Accuracy summarizes the performance of the classification task by dividing the total correct prediction over the total prediction made by the model. It is the number of correctly predicted samples out of all the samples. However, since all of the three datasets used are highly imbalanced, we also use Balanced-Accuracy [Brodersen *et al.*, 2010] in order to shade more lights on our model's evaluation on imbalanced datasets compared to the Accuracy. It is the arithmetic mean of the true positive rate for each class.

5.5.1 Datasets

To evaluate our method, we carried out experiments using three well-known and real-world datasets [Dua et Graff, 2017]. They each contain known protected features, which allowed us to evaluate our method on appropriate cases. These datasets were chosen on the basis of the differences and the characteristics, i-e, number of instances, dimensionality and class imbalance. These datasets also provide an interesting benchmark, which is tough, for fairness evaluation as most of recent proposed fairness approaches in the literature have used them. Moreover, they facilitated our comparison with other competitors.

- **Adult census income** dataset [Dua et Graff, 2017] contains census data from the U.S whose task it to predict whether someone's income exceeds "50K/yr". After removing duplicate instances and instances with missing values, we ended up with $n = 45,175$ instances. Like our competitors, $P = gender$ was considered as protected feature with $P_0 = female$ and $P_1 = male$. Ratio between unprivileged and privileged instances is 2.23 and 3.53 between classes.

- **Bank dataset [Dua et Graff, 2017]** is related to direct marketing campaigns of a Portuguese banking institution with $n = 40,004$ instances. The task is to determine whether a person subscribes to the product (bank term deposit). As target class we consider people who subscribed to a term deposit. Again like our competitors, we consider $P = maritalstatus$ as protected feature with $P_0 = married$ and $P_1 = unmarried$. The dataset suffers from severe class imbalance with global ratio between unprivileged and privileged instances of 2.13. This dataset also suffers from a huge imbalance ratio between classes is 7.57.

- **KDD census dataset [Dua et Graff, 2017]** is basically the same with Adult census, however the target field of this data, was drawn from the "total person income" field rather than the "adjusted gross income" and, therefore, behave differently than the original Adult target field. This dataset is very skewed, the global ratio between unprivileged and privileged instances is 1.09 . $P = gender$ was considered as protected feature with $P_0 = female$ and $P_1 = male$ like in the other methods

used for comparison. In terms of class imbalance, this is a very skew dataset; the ratio between classes is 15.11. More details on these datasets are given in Table 5.1.

Table 5.1: **Experimental Datasets. For each dataset, n instances: number of instances of each dataset, m Features: number of features, P Feature: protected feature, P Ratio: ratio between privileged (P_1) and unprivileged (P_0) group of the protected feature, Class Ratio: ratio between class label of the dataset**

	Adult Income	**Bank**	**KDD Adult**
n Instances	45175	40004	299285
m Features	14	16	41
P Feature	Gender	Marital S.	Gender
Privileged	Male	Unmarried	Male
Unprivileged	Female	Married	Female
P Ratio	2.23	2.13	1.09
Class Ratio	3.53	7.57	15.11
Majority Label	1	1	1

5.5.2 Experimental Baseline

We compare our approach to three other recent state-of-the-art proposed methods tackling the problem of imbalance and protected attributes with the aim of improving fairness. The three other approaches used for comparison are:

- *AdaFair* [Iosifidis et Ntoutsi, 2019]. This method is a fairness-aware boosting approach that adapts AdaBoost to fairness by changing the data distribution at each round based on the notion of cumulative fairness.
- *Fairness Aware Ensemble (FAE)* [Iosifidis *et al.*, 2019]. This strategy is a fairness aware classification that combines pre-processing balancing strategies with post-processing decision boundary adjustment. They use a bagging approach to create sub-datasets while handling the imbalance by an undersampling strategy.
- *SMOTEBoost* [Chawla *et al.*, 2003]. This is an extension of AdaBoost for unbalanced data where new synthetic instances of the minority class are created

using SMOTE [Chawla *et al.*, 2002] at each boosting round to compensate the imbalance. This strategy does not tackle the fairness problem, however we used its performance score to evaluate fairness and see if by addressing only the imbalance between classes, the fairness problem can be resolved.

To evaluate and compare the proposed method to existing methods, we proceeded to a learning task by considering a binary classification problem over the three datasets. For this binary classification problem, Decision Tree is used as the base classifiers. This choice is made in order to be consistent with the evaluation protocol for concurrent methods. For training and testing, first we use the classic train-test split strategy with a 70%-30% respectively then use k-fold validation on the train set, $2K + 1$ folds in total with K the number of clusters obtained for the used dataset. The folds are made by preserving the percentage of samples for each class.

5.5.3 Results Analysis

For the results of our analysis, we present first the result for our stability algorithm that allows us to select the K numbers of stable clusters to use prior our learning strategy. Secondly, we present the predictive performance and fairness, reported on Balanced Accuracy (Bal. Acc.), Accuracy, and Equalized of Odds (EqOd) by comparing our method to the results of the three datasets of the benchmark. Third, we present the effects of different imbalance ratio on the performance.

5.5.3.1 Cluster Stability

In Table 5.2 below, we report the results for our stability algorithm, the value of K and the stability rate for each dataset. For Adult Income dataset, the best and stable value for K is 4 with a stability rate of 89%. This means, among all of possible values for K, we tried 12 values, $K = 4$ is the one that allowed us to obtain more consistent and stable clusters.

Table 5.2: **Cluster stability**

	Adult Income	Bank	KDD Adult
K Value	4	5	4
Stability Rate	89%	92%	92%

5.5.3.2 Performance and Fairness Analysis

Adult Income Dataset

Performance results with the different approaches for this dataset are presented in Table 5.3. For predictive performance, we can see that three methods, our proposed one FAPFID, AdaFair and FAE achieve the same and highest performance score of 83% for Accuracy. However, like we stated above, Accuracy is not that reliable when we are dealing with imbalanced data. Since this dataset suffers from class imbalance, Balanced-Accuracy is the metric that will tell us how good our model is in terms of performance score. For Balanced-Accuracy, our proposed method FAPFID outperforms our competitors with a score of 83% as the highest, then FAE and SMOTEBoost both with 81%. We notice that FAPFID performance score is the same for Balanced-Accuracy and Accuracy, this is meaningful since it highlights our strategy of balancing with regards to the protected features in each subgroup prior training the classifier.
For fairness score, we see clearly that our proposed method FAPFID has surpassed the other three methods used for comparison. FAPFID has the lowest Equalized Odds score, 0.05 (the lower the better for EqOd) following by AdaFair with 0.08. In short, the proposed method outperforms our competitors on this dataset in terms of Balanced-Accuracy and Fairness score.

Table 5.3: **Adult Income: Predictive and Fairness performance, the best results are in bold.**

Score	FAPFID	AdaFair	FAE	SMOTEBoost
Bal. Acc.	**0.83**	0.78	0.81	0.81
Accuracy	**0.83**	**0.83**	**0.83**	0.80
EqOd	**0.05**	0.08	0.15	0.47

Bank Dataset

Performance results with the different approaches for this dataset are presented in Table 5.4. For predictive performance, our proposed method and SMOTEBoost achieve the same and highest performance score of 90%. However, since we are dealing with imbalanced data, we look at Balanced-Accuracy instead. For this, our proposed method achieves the highest score for Balanced-Accuracy, 82% following by the others with a Balanced-Accuracy score under 79%.
For fairness score, our proposed method has surpassed the other three methods used for comparison since it has the lowest Equalized Odds score, 0.1 following by FAE and SMOTEBoost with -0.12 and 0.12 respectively. Again, the proposed method outperforms our competitors on this dataset in terms of Balanced-Accuracy and Fairness score.

Table 5.4: **Bank Dataset: Predictive and Fairness performance, the best results are in bold.**

Score	FAPFID	AdaFair	FAE	SMOTEBoost
Bal. Acc.	**0.82**	0.77	0.78	0.74
Accuracy	**0.90**	0.87	0.83	**0.90**
EqOd	**0.10**	0.27	-0.12	0.12

KDD Adult Dataset

Performance results with the different approaches for this dataset are presented in Table 5.5. For predictive performance, we can see that three methods, FAE achieves the highest performance score of 95% for Accuracy following by the proposed method, 92%. However, our proposed method has the highest Balanced-Accuracy score, 88% which is the one we look at if since this dataset is highly imbalanced. Despite the fact that FAE has the highest Accuracy score, it fails to provide a great Balanced-Accuracy score, it achieves the lowest score of 66%. That means, since this dataset is highly imbalanced, FAE has a higher predictive rate for one group at the expense of the other. Our proposed approach instead, has a better fairness score, 0.01 which is the lowest here on this dataset. In brief, the proposed outperforms our competitors on this dataset in terms of Balanced-Accuracy and Fairness score.

Table 5.5: **KDD Adult: Predictive and Fairness performance, the best results are in bold.**

Score	FAPFID	AdaFair	FAE	SMOTEBoost
Bal. Acc.	**0.88**	0.84	0.66	0.76
Accuracy	0.92	0.86	**0.95**	0.94
EqOd	**0.01**	0.07	0.27	0.36

Discussion

The results on these three datasets show that our method performs well. Compared to other fairness-aware method for dealing with protected feature and data imbalance in machine learning algorithm, we clearly see that our a method has a higher score for Balanced-Accuracy and the lowest score for fairness evaluation. Even in the case where other methods achieve an higher or equal value for Accuracy and Balanced-Accuracy, our method still outperforms them in terms of fairness core. This is very interesting for handling social decision problems guarantying a fair outcome for different groups.

In general, on these 3 datasets, we get satisfactory results and we have maintained a good level of performance (balanced-accuracy), and the best fairness score (the lowest) in terms of Equalized of Odds.

5.5.3.3 Effects of Imbalance Ratio

The third goal of our experiments is to evaluate the effects of different imbalance ratio on the performance. Our method FAPFID is able to achieve efficient and reliable results on the benchmarks datasets above. However, in this section, we investigate the effects of imbalance ratio between privileged and unprivileged group for a given dataset. The goal is to observe the evolution of performance scores of the proposed method with regards to different imbalance ratio. Thus, for a given dataset, we create 10 sub-samples where we maintain a fixed imbalance ratio between privileged and unprivileged group, then we report the balanced accuracy for these 10 sub-samples using box-plot.

Basically we proceed as follow: we consider a ratio of 40/60 between unprivileged (P_0) and privileged (P_1) group and create 10 sub-samples, i-e, each sub-sample is created with 40% of (P_0) and 60% of (P_1). We repeated this by varying the ratios such that we obtain different imbalanced ratios between privileged and unprivileged group. The different ratio that we have used are: 30/70, 20/80, 10/90 and 1/99.

We report on Fig. 5.4 the results obtained with Adult Income dataset for performance using Balanced-Accuracy. As we can see, there is a huge difference between performance scores for different ratio of imbalance. For an imbalance ratio of at least 20% (for P_0), our method still maintains a great averaged Balanced-Accuracy score of 80% at least. With an imbalanced ratio of 10/90, our method suffers from a decreasing in terms of Balanced-Accuracy. We also tested on an extreme case of imbalance ratio between P_0 and P_1 :1/99 where we observed a performance loss. This is because there are not enough P_0 in the cluster so the oversampling method used, SMOTE, can not generate as many meaningful samples as possible for the under-represented group P_0.

We also report on Fig. 5.5 the results obtained with Adult Income dataset for fairness using Equalized of Odds. For an imbalanced ratio between 20/80 and 40/60, we get satisfactory results in terms of fairness score with an average score under 0.1 which acceptable for Equalized Odds. However, starting at 10/90 to lower, our method has limited ability to maintain a high level of fairness on this dataset due to the limitations of the oversampling method used and the lack of data for the under-represented group. A limitation that we will later overcome in our future work.

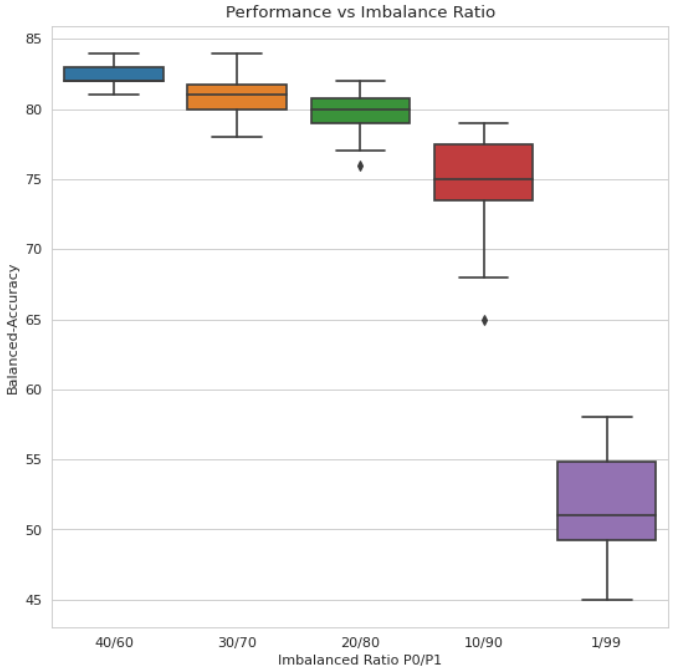

Figure 5.4: Effects of Imbalanced Ratio on Balanced-Accuracy

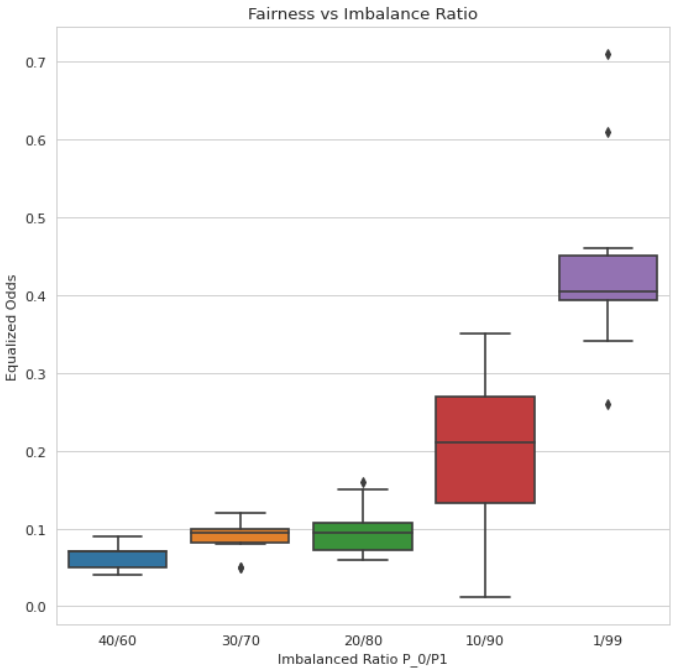

Figure 5.5: Effects of Imbalanced Ratio on Fairness

5.6 Summary

In this chapter, we have proposed a fairness-aware ensemble learning method based on balanced and stable clusters. The proposed method achieves fairness with regards to protected features and class imbalance while maintaining a great performance score.

To do this, we divide the inputs dataset into stable clusters and ensure that privileged and unprivileged groups are fairly represented in each clusters. To obtain stable clusters, we introduce a stability clustering approach that helps maintaining a better homogeneity between clusters. To ensure that privileged and unprivileged instances are fairly represented in each cluster, we have used a novel strategy where we compute a balanced ratio rate within cluster and apply SMOTE only on clusters where the balanced ratio is $\neq 1$; i.e where it exists a group imbalance between privileged and unprivileged instances.

The performance of our method was experimentally evaluated on three well known biased datasets that are largely used in fairness study. Compared to recent state-of-the-art fairness-aware methods, we obtain satisfactory results and the proposed approach outperforms our competitors in terms of performance (Balanced-Accuracy) and fairness (EqOd) scores. The comparative results obtained show our method's effectiveness in boosting fairness with regard to protected feature and class imbalance while maintaining a high level of performance. Beside the comparatives results, we also analyzed the effect of imbalance ratio between subgroups on our model's performance.

For our future work, we will look forward to generalise our approach on datasets that are not part of this benchmark and improve our model's performance in dealing with datasets that suffer from a high (10/90) imbalance ratio.

Source code, data and results for this contribution are avaible and can be accessed under request only via the thesis's repository on GitHub here (Contrib5).

Chapter 6

Conclusions and Future Work

Contents

6.1 Conclusion and Future Work

In this thesis, we have focused on data bias and its implications. Our research addressed the entire learning-based decision-making process in AI with the aim of understanding the different sources of bias, detecting them and mitigating their effects on the results produced for specific applications. Besides the problematic of data bias, we have investigated the issues of fairness and protected features that are related to data bias. To support our work, we were focused on using machine learning approaches under supervised and unsupervised learning.

In **chapter 2**, we have reviewed the scientific literature on different aspects. Firstly, we have investigated the notion of data bias with definitions of the different types of bias that may exist and that other researchers have pointed out in the literature. Based on the different definitions and the impacts of these biases on a learning process, we were able to propose a bias taxonomy. This bias taxonomy makes it possible, using a data science process, to identify where biases can occur in the process. Besides, in the scientific literature on bias, we have investigated the main causes and practices that can lead to bias, the presence or the use of protected attribute/feature is somehow one. Then, we have presented some concrete examples where the use of protected attribute/feature had led to biased decisions in real life applications. Secondly, we have also reviewed the issue of class imbalance in machine learning. We saw that class imbalance can be considered as one of the causes of discriminatory output in decision-making based on machine learning algorithms. We have also recalled on different methods that have been used to handle class imbalance. These methods can be classed either as under-sampling, over-sampling or a combination of the two of them. Finally, we have pointed out the different fairness metrics that exist which make it possible to estimate the degree of fairness of different methods when mitigating bias. For fairness metrics, we also saw that there exist others techniques that can be used. Such techniques include pre-processing, in-processing and post-processing mechanisms. We also saw that despite the fact that different fairness metrics exist and have their own merit, there is no consensus whatsoever on which metric is the best.

In **chapter 3**, we have investigated the problem of feature redundancy in machine learning. For all the existing redundancy methods reviewed, they are mostly based on correlation between features. Reviewing existing redundancy methods, we showed that using a single defined threshold in redundancy analysis can be subjective. The subjectiveness is due to the fact that there is no consensus on what is referred as to highly correlated features and the fact that the latter one is based on a user fixed or defined threshold. We showed that, using a same dataset, different users can find different redundant features if they choose a different threshold. We demonstrated how dangerous that can be in the case of a dataset containing protected features. To overcome this situation, we have proposed an new redundancy method to evaluate feature redundancy in machine learning. The proposed redundancy method uses symmetrical uncertainty

as correlation measure. Unlike existing proposed methods, the proposed redundancy method does not require to set any threshold. Experiments were carried out using well-known datasets. Performance of the proposed method was experimentally compared to three others state-of-the-art redundancy methods, results show that the proposed method achieved satisfactory results.

In **chapter 4**, we have addressed the issues related to fairness, redundancy and the use of protected features in machine learning algorithms. Moreover, we have pointed out the limited ability of traditional features selection methods to deal with protected features over data distribution due to data imbalance. In order to overcome the limits pointed out, we have proposed in chapter 4 an outcome-fairness algorithm to deal with protected features in decision support based on machine learning. The proposed algorithm improves fairness and performance in the case of protected features while considering their redundant. Unlike existing fairness methods, the proposed method in chapter 4 does not use any protected feature in their model but instead, takes into consideration redundancy while making decision on fairness. To take into consideration the redundancy, we introduce in the proposed algorithm a more flexible way to assess features redundancy by using a threshold space.

In **chapter 5**, we focused mainly on two problems that directly impact performance and fairness on machine learning algorithms: the use of protected features and class imbalance. In the proposed approach called FAPFID in chapter 5, we introduce a fairness-aware approach for protected features and imbalanced data. The method allowed to handle protected features and class imbalance while ensuring a fair model for decision-making involving machine learning algorithms. To do so, the method creates a set of balanced and stable clusters from the input dataset; clusters were balanced using SMOTE. This is done in order to ensure that both privileged and underprivileged groups are fairly represented. To create its final model, the algorithm uses an ensemble learning strategy by aggregating all previously balanced and stable clusters. The performance of this proposed method is evaluated on well known and real-world biased datasets from the literature that are widely used in fairness study. Comparison to recent state-of-the art fairness methods shows our method's effectiveness in boosting model fairness with regard to protected features and class imbalance. In this chapter, we also showed that imbalance ratio has a significant impact on model fairness in machine learning algorithm.

Overall, in this thesis, we have addressed the issue of data bias in decision-making systems based on machine learning algorithm. We have drawn up a rich literature of the different types of data bias. Furthermore, we also looked at various issues that hamper the obtaining of fair results in decision-making based on machine learning. By pointing out the limits of existing solutions, we have proposed various fairness methods which make it possible to deal with biases induced by protected features and by class imbalance. Experiments, although with limits, proved that our solutions work well in the studies case that have been used.

Future Work:

The proposed bias taxonomy in chapter 2 addresses biases that occur in the data science process. However, the proposed taxonomy does not cover all the existing types of bias. A more complete taxonomy can be created based on the idea of the previous one in order to extend the taxonomy and include more types of bias like algorithm bias for example that we did not consider in our data science process.

The method proposed in chapter 4 is a very interesting approach, however it is limited to dataset containing protected features only. A more in-depth study can be done to extend this method to other datasets. Another improvement of this method could seek to consider more than one protected features at once, which we had not done in the initial method. Basically, one will need to modify the algorithm and allow user to set the number of protected feature that they want to assess redundancy for. As for now, our approach has used only known protected features of the dataset. However robustness of the proposed method can be evaluated by letting user the choice of using any feature as protected.

In chapter 5, the proposed method has some merits regarding fairness improvements. However, future work can be done to reduce computational time and improve its global performance. Besides the fact that the proposed approach in chapter 5 improves fairness with regard to class imbalance and protected features, it does have some limits that can be boosted. In our work, we were mainly focused on binary classification and datasets containing protected features only. An improvement to this work can extend the solution to a multi-class task or using a deep-learning approach. Also, one can seek to change the balancing strategy used prior training the model.

Appendix A

Annex

Contents

A.1 Algorithms Used

In this thesis we have used supervised and unsupervised learning techniques. For each one, there is a set of variables that might be denoted as inputs. These inputs have some influence on one or more outputs. For supervised learning, the goal is to find the precise mapping between input and output, referred to as labeled data as shown on Fig. A.1. In the unsupervised learning problem, we observe only the features without labels (Fig. A.2) and have no measurements of the output. The task is rather to describe how the data are organized or clustered.

To understand the algorithms presented below, we consider a set of input data annotated $X_{n,m} \in R^{n \times m}$ consisting of n independents distributed samples and m non-independents features with $F = \{F_0, ..., F_m\}$ being the feature space.

For the ease of simplicity, we consider a binary classification scenario, where $y \in \{0, 1\}$ are the class labels with 0 is a negative label and 1, the positive label. The goal of supervised classification is to find the learned function $\hat{f}$: $X \rightarrow Y$ to predict the class labels of unseen instances. In the context of fairness, the function $\hat{f}$ should help minimizing bias. The function $\hat{f}$ is obtained via machine learning algorithms, below we describe we describe the algorithms that we have used in this thesis.

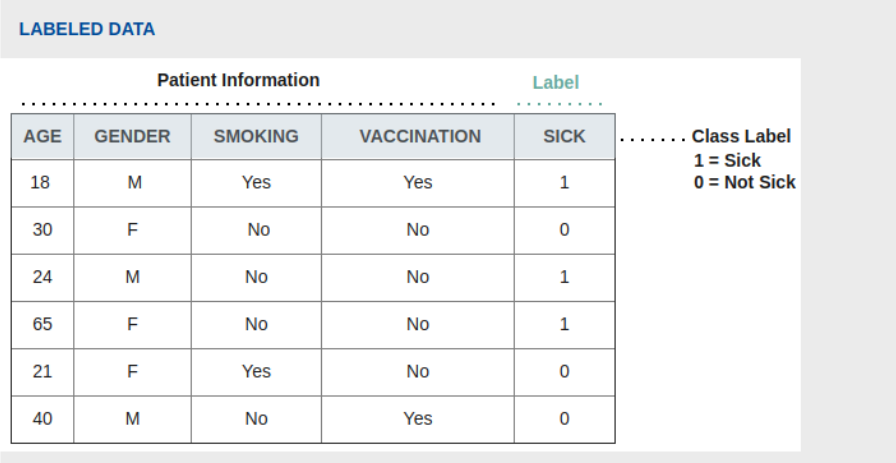

Figure A.1: An example of labeled data

A.1.1 Supervised Algorithms

A.1.1.1 Naive Bayes

Naive Bayes [Webb *et al.*, 2010] are a family of simple probabilistic classifiers based on applying Bayes' theorem [Joyce, 2003] with strong (naive) independence assumptions

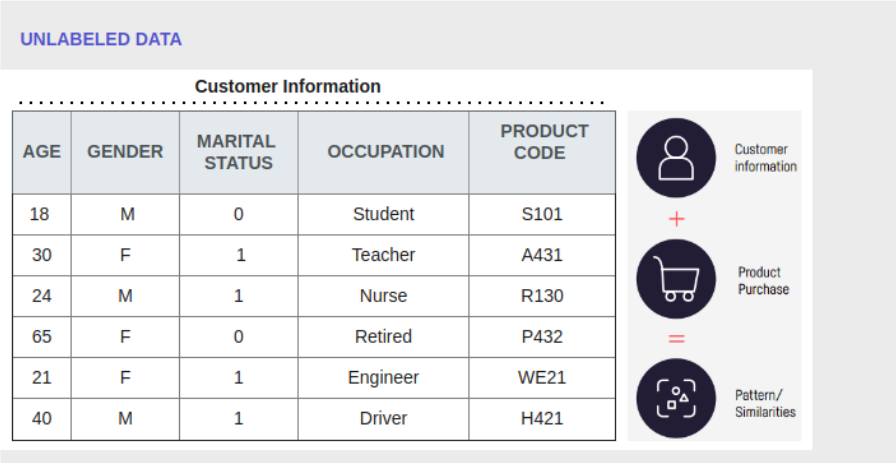

Figure A.2: An example of unlabeled data

between the features. In simple terms, a Naive Bayes classifier assumes that the presence of a particular feature in a class is unrelated to the presence of any other feature. By probabilistic classifiers, we mean a machine learning model that is able to predict. In other words, Naive Bayes methods are a set of supervised learning algorithms based on applying Bayes' theorem with the "naive" assumption of conditional independence between every pair of features given the value of the class variable. Bayes' theorem states the following relationship, given class labels y and dependent feature vector x_i through X :

$$\mathbb{P}(y \mid x_1, \ldots, x_n) = \frac{\mathbb{P}(y)\mathbb{P}(x_1, \ldots, x_n \mid y)}{\mathbb{P}(x_1, \ldots, x_n)} \tag{A.1}$$

Using the naive conditional independence assumption that

$$\mathbb{P}(x_i|y, x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n) = \mathbb{P}(x_i|y), \tag{A.2}$$

for all i, this relationship can be simplified as

$$\mathbb{P}(y \mid x_1, \ldots, x_n) = \frac{\mathbb{P}(y) \prod_{i=1}^{n} \mathbb{P}(x_i \mid y)}{\mathbb{P}(x_1, \ldots, x_n)} \tag{A.3}$$

Since $\mathbb{P}(x_1, \ldots, x_n)$ is constant given the input, we can write the following classification rule:

$$\begin{gathered} \mathbb{P}(y \mid x_1, \ldots, x_n) \propto \mathbb{P}(y) \prod_{i=1}^{n} \mathbb{P}(x_i \mid y) \\ \Downarrow \\ \hat{y} = \arg\max_y \mathbb{P}(y) \prod_{i=1}^{n} \mathbb{P}(x_i \mid y), \end{gathered} \tag{A.4}$$

and use Maximum A Posteriori (MAP) estimation [Leung, 2007] to estimate $\mathbb{P}(x_i \mid y)$ and $\mathbb{P}(y)$, with $\mathbb{P}(y)$ the relative frequency of class y in the training set.

Naive Bayes classifiers have worked quite well in many real-world situations, such as document classification and spam filtering. They require a small amount of training data to estimate the necessary parameters [Zhang, 2004].

Naive Bayes make the assumption that the features are independent, however, since we have worked with non-independent features as defined in section A.1, we have used **Gaussian Naive Bayes** [Raizada et Lee, 2013, Bustamante *et al.*, 2006] which is a variant of Naive Bayes that follows Gaussian normal distribution and supports continuous data. The Gaussian Naive Bayes can be written as:

$$\mathbb{P}(x_i \mid y) = \frac{1}{\sqrt{2\pi\sigma_y^2}} \exp\left(-\frac{(x_i - \mu_y)^2}{2\sigma_y^2}\right) \tag{A.5}$$

with (σ_y) and (μ_y) two parameters that are estimated using maximum likelihood estimation [Myung, 2003]. In Fig. A.3, we show how Gaussian Naive Bayes work.

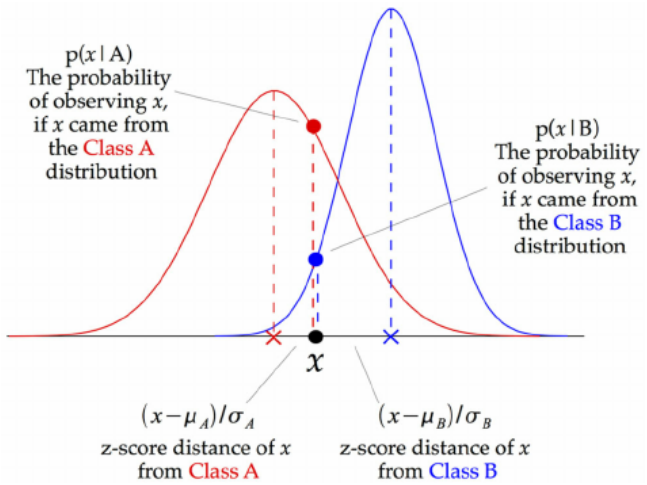

Figure A.3: Illustration of how a Gaussian Naive Bayes (GNB) by [Bustamante *et al.*, 2006]. For each data point, the z-score distance between that point and each class-mean is calculated, namely the distance from the class mean divided by the standard deviation of that class.

A.1.1.2 Decision Trees

Decision tree [Chauhan et Chauhan, 2013] is a very popular learning algorithm used in supervised learning tasks [Kraiem *et al.*, 2021]. It can be used for both classification (prediction of a category) and regression (prediction of a numerical output) problems.

It is a tree-structured classifier that consist of three main layers: the root node, the internal nodes and finally the leaf nodes as shown in Fig A.4. Given a labeled set of input data, the method splits the instances, based on a splitting criterion, into different leaves. The internal nodes contain the decisions of each split. The expansion of the decision tree is based on i) the split criteria, and ii) the stop criterion. The splitting

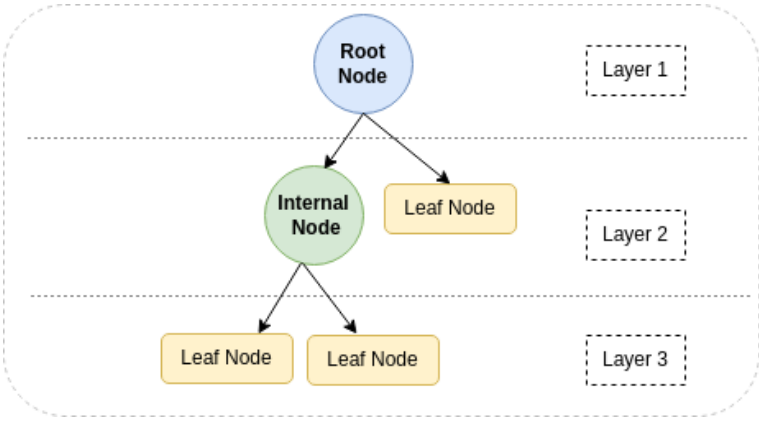

Figure A.4: Basic structure of a decision tree [Chiu *et al.*, 2016]

criteria is used to split a node into sub-nodes that can be an internal nodes or leaf nodes. It can be defined using a specific measure. Commonly used specific measures are information gain (also called mutual information) [Rokach et Maimon, 2005], gini impurity [Grabmeier et Lambe, 2007], chi_square [Wang et Chen, 2021]. However, through our thesis, we have used the information gain which as splitting criteria, since as explained by [Tangirala, 2020], there is no significant difference in terms of performance between these two. Information gain is an entropy-based measure; i.e. the measure of uncertainty of a random variable. By splitting the nodes based on a feature, decision trees try to decrease the entropy on the node. Entropy is defined as:

$$H(X) = -\sum_{i} \mathbb{P}(x_i) \log_2 \mathbb{P}(x_i) \tag{A.6}$$

and the entropy of X observing Y is:

$$H(X|Y) = \sum_{j} \mathbb{P}(y_j) \sum_{i} \mathbb{P}(x_i|y_j) \log_2 \left(\mathbb{P}(x_i|y_j)\right) \tag{A.7}$$

Where $\mathbb{P}(xi)$ represents the prior probabilities for all values of X and $\mathbb{P}(x_i|y_j)$, the conditional probabilities of X being given the values of Y. The statistical difference between $H(X)$ and $H(X|Y)$ is called information gain or mutual information Thus, using equations A.6 and A.7, information gain or mutual information can be defined by:

$$IG(X,Y) = H(X) - H(X|Y) \tag{A.8}$$

Decision trees need a stopping criteria otherwise it would be undesirable to grow a tree in which each case occupied its own node. The resulting tree would be computationally expensive, difficult to interpret and would probably not work very well with new data. Some stopping criteria [Banfield *et al.*, 2006] include but not limited to the number of cases in the node is less than some prespecified limit or the depth of the node is more than some pre-specified limit.

A.1.1.3 Random Forest

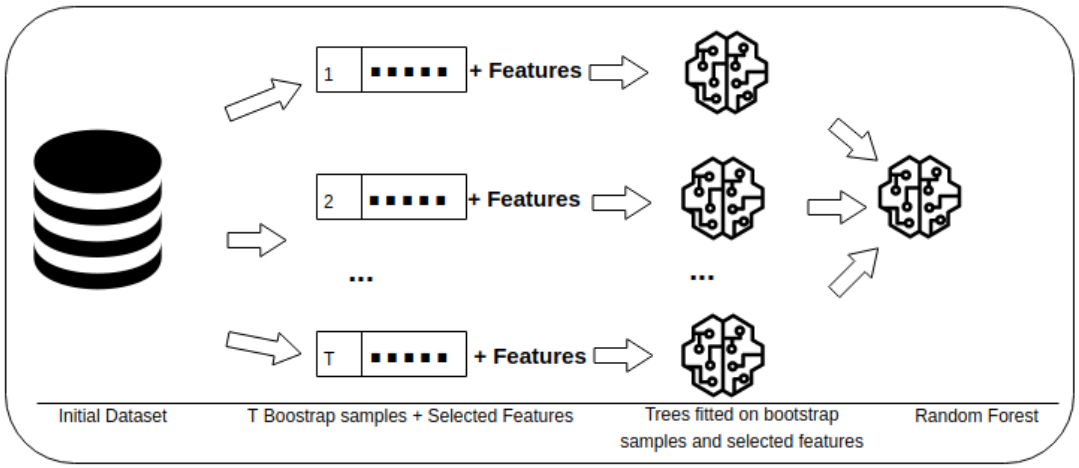

Figure A.5: Random Forest process

Random Forest [Liaw *et al.*, 2002], as presented on Fig.A.5, is a tree-based ensemble that makes use of building multiple classification trees based on bootstrapped random training samples. It builds decision trees on different random subsets and takes their majority vote for classification and average in case of regression. The size of a random subset is typically the square root of the total number of features. More formally, for a m-dimensional random vector $X = (X_1, ..., X_m)^T$ representing the real-valued inputs (or predictor variables) and a random variable Y representing the real-valued response with a joint distribution $P_{XY}(X, Y)$. The goal is to find a prediction function $\hat{f}(x)$ for predicting Y. The prediction function is determined by a loss function $L(Y, \hat{f}(x)$ and defined to minimize the expected value of the loss:

$$E_{XY}\big(L(Y, \hat{f}(x)\big) \tag{A.9}$$

where the subscripts denote expectation with respect to the joint distribution of X and Y. Intuitively, $L(Y, \hat{f}(x)$ is a measure of how close $\hat{f}(x)$ is to Y; it penalizes values of $\hat{f}(x)$ that are a long way from Y. Typical choices of L are squared error loss

$L(Y, f(X)) = (Y\text{-}f(X))^2$ for regression and zero-one loss for classification:

$$L(Y, \hat{f}(x)) = I(Y \neq \hat{f}(x)) = \begin{cases} 0 \; if \; Y = \hat{f}(x) \\ 1 \; else \end{cases} \tag{A.10}$$

Ensembles construct f in terms of a collection of so-called "base learners" $h_1(x), ..., h_J(x)$ and these base learners are combined to give the "ensemble predictor" $\hat{f}(x)$. In regression task, the base learners are averaged to obtain the final prediction as:

$$\hat{f}(x) = 1J \sum_{j=1}^{J} h_j(x) \tag{A.11}$$

while for a classification task, $\hat{f}(x)$ is the most frequently predicted class ("voting")

$$\hat{f}(x) = \underset{y \in \hat{y}}{\operatorname{argmax}} \sum_{j=1}^{J} I(y = h_j(x)) \tag{A.12}$$

with $\hat{y}$ the set of possible outcomes of Y.

A.1.1.4 Bagging

Bagging [Sutton, 2005] is an ensemble learning technique which aims to reduce the learning error by implementing a set of homogeneous machine learning algorithms. The main idea behind Bagging is that L intermediate and independent base learners are used and trained separately by using bootstrapped samples, i.e. smaller samples of the same size are repeatedly drawn, with replacement, from the original input set. These intermediate base learners produce a final stable and accurate model through a voting or averaging strategy.

Consider the basics settings as in section A.1, Bagging algorithm works as follow: Given a training set $X = (x_1, y_1), ...(x_n, y_n)$,

- sample T sets of n elements from X (with replacement) such as $X_1, X_2, ...X_T \rightarrow T$ quasi replica training sets;
- fit a machine learning model on each $X_i, i = 1, ..., T$ to obtain a sequence of T outputs $f_1(x), ...f_T(x)$
- the final aggregate model for classification is then $\hat{f}(x) = \underset{y \in \hat{y}}{\operatorname{argmax}} \sum_{i=1}^{T} I(f_i(x))$ with I the identity function.

Below on Fig.A.6, we show how this technique works.

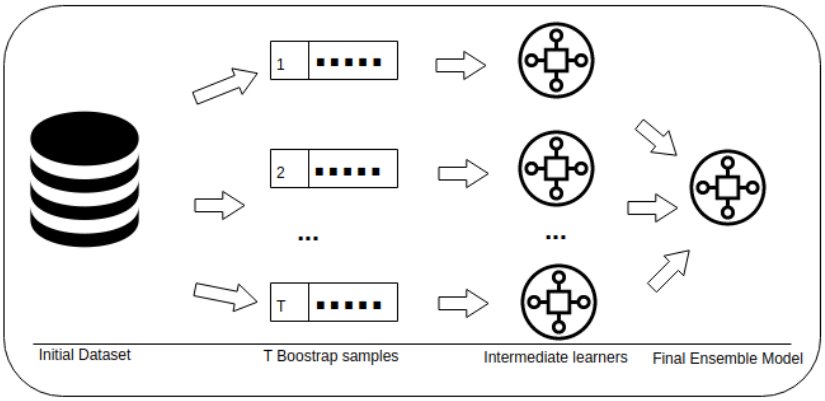

Figure A.6: Bagging - Fitting L intermediate models on different bootstrap samples and build an ensemble model that "averages" the results of these weak learners.

A.1.1.5 Boosting

Boosting [Sutton, 2005] is another ensemble learning method where different models are trained sequentially and iteratively. Each current model carries forth the performance

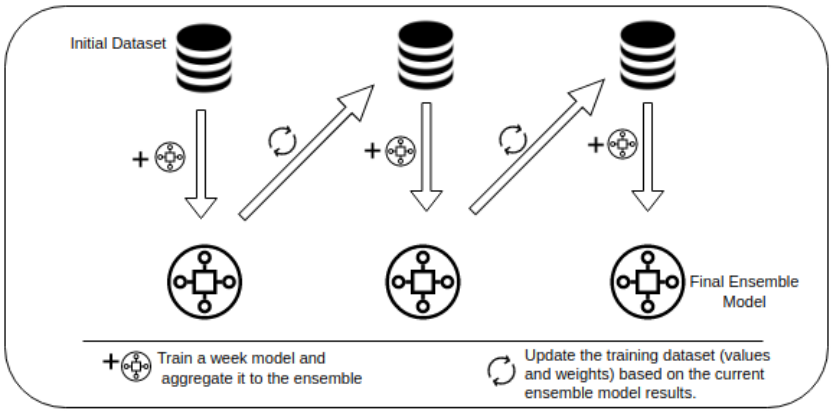

Figure A.7: Boosting process

of the previous so that it attempts to correct the errors and improve its previous model. The current model is aggregated to the ensemble model and "update" the training dataset in order to benefit from the strengths of the current ensemble model when fitting the next base model. Using the basics settings defined in section A.1, Boosting algorithm works as follow:
Given a class $F = f : X :\rightarrow \{0, 1\}$ of weak learners and the data $(x_1, y_1), ..., (x_n, y_n), y_i \in 0, 1$. Initialize the weights as $w_1(i) = 1/n$. For $t = 1, ...T$:

- Find a weak learner ft based on weights $w_t(i)$, compute the weighted error $\theta_t = \sum_{i=1}^{n} w_t(i) I(y_i \neq f(\hat{x})_t(x_i))$ and the importance of f_t as $\alpha_t = 1/2ln\left(\frac{1-\theta_t}{\theta_t}\right)$
- Update the distribution $w_t + 1(i) = \frac{wt(i)e^{-\alpha_t y_i f_t(xi)}}{\sum_{i=1}^{n} w_t(i)e^{-\alpha_t y_i h_t(xi)}}$

The final boosting hypothesis is given by: $g(x) = \sum_{t=1}^{T} \alpha_t f_t(x))$. On Fig.A.7, we show how Boosting technique works.

A.1.1.6 Support Vector Machine (SVM)

A Support Vector Machine (SVM) is a type supervised machine learning algorithm that can be used for both classification and regression purposes. However, SMS is more commonly used in classification problems [Sha'Abani *et al.*, 2020].

The idea of SVM is based on the idea of finding a hyperplane that best divides a dataset into two classes (blue vs red for example). Hyperplane, in two dimensions, it's simply a line) that best separates the tags. This line is the decision boundary, i.e anything that falls to one side of it we will classify as blue, and anything that falls to the other as red.

A.1.1.7 C4.5

C4.5 is an supervised learning algorithm developed by Ross Quinlan [Quinlan, 2014] that use Decision Trees. This algorithm can be used for classification problems. It improves (extends) Decision Tree by dealing with both continuous and discrete attributes, missing values and pruning trees after construction. Being a supervised learning algorithm, it normally requires a set of labeled training examples and each example can be seen as a pair: input object and a desired output class. The algorithm then analyzes the training set and builds a classifier that can classify both training and test examples. A test example is an input object and the algorithm mus assign the example to a class.

A.1.2 Unsupervised Learning Algorithm Used

In this thesis, we have used only one unsupervised learning algorithm. This algorithm (K-means) is mainly used in the work presented in chapter 5.

A.1.2.1 K-means

K-means algorithm is an iterative algorithm that tries to partition the dataset into K-predefined distinct non-overlapping subgroups (clusters) where each data point belongs to only one subgroup (cluster) [Fig. A.8]. In order word, the algorithm divides a set of N sample X into K disjoint clusters, each described by the mean μ_j of the samples in the

cluster [Ahmed *et al.*, 2020]. The means are commonly called the cluster "centroids". K-means try to make the intra-cluster data points as similar (close) as possible while also keeping the clusters as different (far) as possible. It assigns data points to a cluster such that the sum of the squared distance between the data points and the cluster's centroid is at the minimum. The algorithm works as follows:

1. Specify number of clusters K.
2. Initialize centroids by first shuffling the dataset and then randomly selecting K data points for the centroids (center of the cluster). [Hamerly et Elkan, 2003].
3. Assign each data point to the closest cluster (centroid).
4. Re-initialize centroids by calculating the average of all data points of that cluster.
5. Repeat steps 3 and 4

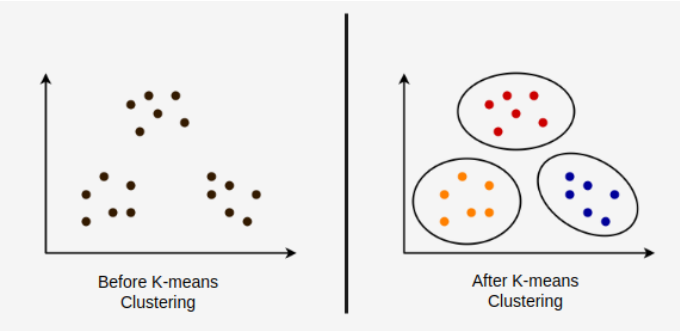

Figure A.8: An example of K-means clustering

Author Publications

Through this Thesis, I have co-authored and published the following papers:

(1) Ginel Dorleon and Nathalie Bricon-Souf and Imen Megdiche and Olivier Teste Qualification du biais de données dans le processus de la science des données, Revue des Nouvelles Technologies de l'Information EGC'21.

(2) Ginel Dorleon, Nathalie Bricon-Souf, Imen Megdiche, and Olivier Teste Absolute Redundancy Analysis Based on Features Selection. In DSIT '21: 4th International Conference on Data Mining and Big Data (DMBD 2021), Shangai, China. ACM, New York, NY, USA

(3) Ginel Dorleon, Imen Megdiche, Nathalie Bricon-Souf, and Olivier Teste. 2022 Feature Selection Under Fairness Constraints. In Proceedings of ACM SAC Conference (SAC'22). ACM, New York, NY, USA

(4) Ginel Dorleon, Imen Megdiche, Nathalie Bricon-Souf, and Olivier Teste. Feature Selection Under Fairness and Performance Constraints. In Proceedings of The 24th International Conference on Big Data Analytics and Knowledge Discovery (DaWaK 2022), Vienna, Austria

(5) Ginel Dorleon, Imen Megdiche, Nathalie Bricon-Souf, and Olivier Teste. 2022 FAPFID: A Fairness-aware Approach for Protected Feature and Imbalanced Data [TLDKS 2023]

Bibliography

Bibliography

[Abinash et Vasudevan, 2018] ABINASH, M. et VASUDEVAN, V. (2018). A study on wrapper-based feature selection algorithm for leukemia dataset. *In Intelligent Engineering Informatics*, pages 311–321. Springer.

[Agarwal *et al.*, 2018] AGARWAL, A., BEYGELZIMER, A., DUDÍK, M., 0001, J. L. et WALLACH, H. M. (2018). A Reductions Approach to Fair Classification. *In Proceedings of the 35th International Conference on Machine Learning*, pages 60–69. PMLR.

[Agarwal *et al.*, 2019] AGARWAL, A., DUDÍK, M. et WU, Z. S. (2019). Fair regression: Quantitative definitions and reduction-based algorithms. *CoRR*, abs/1905.12843.

[Ahmed *et al.*, 2020] AHMED, M., SERAJ, R. et ISLAM, S. M. S. (2020). The k-means algorithm: A comprehensive survey and performance evaluation. *Electronics*, 9(8): 1295.

[Alipourfard *et al.*, 2018] ALIPOURFARD, N., FENNELL, P. G. et LERMAN, K. (2018). Using simpson's paradox to discover interesting patterns in behavioral data. *In Proceedings of the 12th International AAAI Conference On Web And Social Media (ICWSM2018)*. AAAI.

[Alon *et al.*, 1999] ALON, U., BARKAI, N., NOTTERMAN, D. A., GISH, K., YBARRA, S., MACK, D. et LEVINE, A. J. (1999). Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. *Proceedings of the National Academy of Sciences*, 96(12):6745–6750.

[Amarasinghe *et al.*, 2018] AMARASINGHE, T., APONSO, A. et KRISHNARAJAH, N. (2018). Critical analysis of machine learning based approaches for fraud detection in financial transactions. *In Proceedings of the 2018 International Conference on Machine Learning Technologies*, ICMLT '18, page 12–17, New York, NY, USA. Association for Computing Machinery.

[Ambure *et al.*, 2019] AMBURE, P., GAJEWICZ-SKRETNA, A., CORDEIRO, M. N. D. et ROY, K. (2019). New workflow for qsar model development from small data sets: small dataset curator and small dataset modeler. integration of data curation, exhaustive double cross-validation, and a set of optimal model selection techniques. *Journal of Chemical Information and Modeling*, 59(10):4070–4076.

[Amini *et al.*, 2019] AMINI, A., SOLEIMANY, A. P., SCHWARTING, W., BHATIA, S. N. et RUS, D. (2019). Uncovering and mitigating algorithmic bias through learned latent structure. *In Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*, AIES '19, page 289–295, New York, NY, USA. Association for Computing Machinery.

[Ang *et al.*, 2016] ANG, J., MIRZAL, A., HARON, H. et HAMED, H. N. A. (2016). Supervised, unsupervised, and semi-supervised feature selection: A review on gene selection. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 13:971–989.

[Angwin et Kirchner, 2016] ANGWIN, Jeff Larson, S. M. et KIRCHNER, L. (2016). *Machine Bias : there is a software used across the country to predict future criminals. And it is biased against blacks*. ProPublica 2016.

[Backurs *et al.*, 2019] BACKURS, A., INDYK, P., ONAK, K., SCHIEBER, B., VAKILIAN, A. et WAGNER, T. (2019). Scalable fair clustering. *In* CHAUDHURI, K. et SALAKHUTDINOV, R., éditeurs : *Proceedings of the 36th International Conference on Machine Learning*, volume 97 de *Proceedings of Machine Learning Research*, pages 405–413. PMLR.

[Baeza-Yates, 2018] BAEZA-YATES, R. (2018). Bias on the web. *Commun. ACM*, 61(6):54–61.

[Banfield *et al.*, 2006] BANFIELD, R. E., HALL, L. O., BOWYER, K. W. et KEGELMEYER, W. P. (2006). A comparison of decision tree ensemble creation techniques. *IEEE transactions on pattern analysis and machine intelligence*, 29(1):173–180.

[Barocas *et al.*, 2017] BAROCAS, S., HARDT, M. et NARAYANAN, A. (2017). Fairness in machine learning. *Nips tutorial*, 1:2.

[Barocas et Selbst, 2016] BAROCAS, S. et SELBST, A. D. (2016). Big data's disparate impact. *California Law Review*, 104:671.

[Bechavod et Ligett, 2017] BECHAVOD, Y. et LIGETT, K. (2017). Learning fair classifiers: A regularization-inspired approach. *ArXiv*, abs/1707.00044.

[Belkacem *et al.*, 2020] BELKACEM, S., BOUSSAID, O. et BOUKHALFA, K. (2020). Ranking news feed updates on social media: A comparative study of supervised models. *Revue des Nouvelles Technologies de l'Information*, Extraction et Gestion des Connaissances , RNTI-E-36:499–506.

[Bellamy *et al.*, 2018] BELLAMY, R. K. E., DEY, K., HIND, M., HOFFMAN, S. C., HOUDE, S., KANNAN, K., LOHIA, P. K., MARTINO, J., MEHTA, S., MOJSILOVIC, A., NAGAR, S., RAMAMURTHY, K. N., RICHARDS, J. T., SAHA, D., SATTIGERI, P., SINGH, M., VARSHNEY, K. R. et ZHANG, Y. (2018). Ai fairness 360: An extensible toolkit for detecting, understanding, and mitigating unwanted algorithmic bias. *ArXiv*, abs/1810.01943.

[Benesty *et al.*, 2009] BENESTY, J., CHEN, J., HUANG, Y. et COHEN, I. (2009). Pearson correlation coefficient. *In Noise reduction in speech processing*, pages 1–4. Springer.

[Breiman et Wald Lecture, 2002] BREIMAN, L. et WALD LECTURE, I. (2002). Looking inside the black box. *Wald Lecture II, Department of Statistics, California University*.

[Brodersen *et al.*, 2010] BRODERSEN, K. H., ONG, C. S., STEPHAN, K. E. et BUHMANN, J. M. (2010). The balanced accuracy and its posterior distribution. *In 2010 20th international conference on pattern recognition*, pages 3121–3124. IEEE.

[Buolamwini et Gebru, 2018] BUOLAMWINI, J. et GEBRU, T. (2018). Gender shades: Intersectional accuracy disparities in commercial gender classification. *In FAT*.

[Bustamante *et al.*, 2006] BUSTAMANTE, C., GARRIDO, L. et SOTO, R. (2006). Comparing fuzzy naive bayes and gaussian naive bayes for decision making in robocup 3d. *In Mexican International Conference on Artificial Intelligence*, pages 237–247. Springer.

[Calmon *et al.*, 2017] CALMON, F., WEI, D., VINZAMURI, B., NATESAN RAMAMURTHY, K. et VARSHNEY, K. R. (2017). Optimized pre-processing for discrimination prevention. *In* GUYON, I., LUXBURG, U. V., BENGIO, S., WALLACH, H., FERGUS, R., VISHWANATHAN, S. et GARNETT, R., éditeurs : *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

[Cao *et al.*, 2021] CAO, Y., FANG, Z., WU, Y., ZHOU, D.-X. et GU, Q. (2021). Towards understanding the spectral bias of deep learning. *In* ZHOU, Z.-H., éditeur : *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, pages 2205–2211. International Joint Conferences on Artificial Intelligence Organization. Main Track.

[Cascaro *et al.*, 2019] CASCARO, R. J., GERARDO, B. D. et MEDINA, R. P. (2019). Filter selection methods for multiclass classification. *Proceedings of the 2nd International Conference on Computing and Big Data - ICCBD 2019*.

[Caton et Haas, 2020] CATON, S. et HAAS, C. (2020). Fairness in machine learning: A survey. *arXiv preprint arXiv:2010.04053*.

[Chandrashekar et Sahin, 2014] CHANDRASHEKAR, G. et SAHIN, F. (2014). A survey on feature selection methods. *Comput. Electr. Eng.*, 40(1):16–28.

[Chauhan et Chauhan, 2013] CHAUHAN, H. et CHAUHAN, A. (2013). Implementation of decision tree algorithm c4. 5. *International Journal of Scientific and Research Publications*, 3(10):1–3.

[Chawla *et al.*, 2003] CHAWLA, N., LAZAREVIC, A., HALL, L. O. et BOWYER, K. (2003). *SMOTEBoost: Improving Prediction of the Minority Class in Boosting*. In PKDD.

[Chawla *et al.*, 2002] CHAWLA, N. V., BOWYER, K. W., HALL, L. O. et KEGELMEYER, W. P. (2002). Smote: Synthetic minority over-sampling technique. *J. Artif. Int. Res.*, 16(1):321–357.

[Chawla *et al.*, 2004] CHAWLA, N. V., JAPKOWICZ, N. et KOTCZ, A. (2004). Special issue on learning from imbalanced data sets. *ACM SIGKDD explorations newsletter*, 6(1):1–6.

[Cherrington *et al.*, 2019] CHERRINGTON, M., THABTAH, F., LU, J. et XU, Q. (2019). Feature selection: filter methods performance challenges. *In 2019 International Conference on Computer and Information Sciences (ICCIS)*, pages 1–4. IEEE.

[Chiu *et al.*, 2016] CHIU, M.-H., YU, Y.-R., LIAW, H. et HAO, L. (2016). The use of facial micro-expression state and tree-forest model for predicting conceptual-conflict based conceptual change.

[Chouldechova *et al.*, 2018] CHOULDECHOVA, A., BENAVIDES-PRADO, D., FIALKO, O. et VAITHIANATHAN, R. (2018). A case study of algorithm-assisted decision making in child maltreatment hotline screening decisions. volume 81 de *Proceedings of Machine Learning Research*, pages 134–148, New York, NY, USA. PMLR.

[Chouldechova et Roth, 2018] CHOULDECHOVA, A. et ROTH, A. (2018). The frontiers of fairness in machine learning. *arXiv preprint arXiv:1810.08810*.

[Corbett-Davies *et al.*, 2017] CORBETT-DAVIES, S., PIERSON, E., FELLER, A., GOEL, S. et HUQ, A. (2017). Algorithmic decision making and the cost of fairness. *In*

Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '17, page 797–806, New York, NY, USA. Association for Computing Machinery.

[Cortes *et al.*, 2001] CORTES, C., PREGIBON, D. et VOLINSKY, C. (2001). Communities of interest. *In Proceedings of the 4th International Conference on Advances in Intelligent Data Analysis*, IDA '01, page 105–114, Berlin, Heidelberg. Springer-Verlag.

[Danks et London, 2017] DANKS, D. et LONDON, A. J. (2017). Algorithmic bias in autonomous systems. *In Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, pages 4691–4697.

[Dash et Liu, 2000] DASH, M. et LIU, H. (2000). Feature selection for clustering. *In Pacific-Asia Conference on knowledge discovery and data mining*, pages 110–121. Springer.

[Datta *et al.*, 2014] DATTA, A., TSCHANTZ, M. C. et DATTA, A. (2014). Automated experiments on ad privacy settings: A tale of opacity, choice, and discrimination.

[Dee, 2005] DEE, D. P. (2005). Bias and data assimilation. *Quarterly Journal of the Royal Meteorological Society: A journal of the atmospheric sciences, applied meteorology and physical oceanography*, 131(613):3323–3343.

[del Barrio *et al.*, 2020] del BARRIO, E., GORDALIZA, P. et LOUBES, J.-M. (2020). Review of mathematical frameworks for fairness in machine learning. *ArXiv*, abs/2005.13755.

[Deng *et al.*, 2009] DENG, J., DONG, W., SOCHER, R., LI, L.-J., LI, K. et FEI-FEI, L. (2009). Imagenet: A large-scale hierarchical image database. *In 2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255.

[Doak, 1992] DOAK, J. D. (1992). An evaluation of feature selection methods and their application to computer security.

[Dobbe *et al.*, 2018] DOBBE, R., DEAN, S., GILBERT, T. K. et KOHLI, N. (2018). A broader view on bias in automated decision-making: Reflecting on epistemology and dynamics. *ArXiv*, abs/1807.00553.

[Dorleon *et al.*, 2021a] DORLEON, G., BRICON-SOUF, N., MEGDICHE, I. et TESTE, O. (2021a). Absolute redundancy analysis based on features selection. *In 2021 4th International Conference on Data Science and Information Technology*, DSIT 2021, page 458–461, New York, NY, USA. Association for Computing Machinery.

[Dorleon *et al.*, 2021b] DORLEON, G., BRICON-SOUF, N., MEGDICHE, I. et TESTE, O. (2021b). Qualification du biais de données dans le processus de la science des données. *Revue des Nouvelles Technologies de l'Information*, Extraction et Gestion des Connaissances, RNTI-E-37:515–516.

[Dorleon *et al.*, 2022a] DORLEON, G., MEGDICHE, I., BRICON-SOUF, N. et TESTE, O. (2022a). Feature selection under fairness and performance constraints. *In Big Data Analytics and Knowledge Discovery: 24th International Conference, DaWaK 2022, Vienna, Austria, August 22–24, 2022, Proceedings*, page 125–130, Berlin, Heidelberg. Springer-Verlag.

[Dorleon *et al.*, 2022b] DORLEON, G., MEGDICHE, I., BRICON-SOUF, N. et TESTE, O. (2022b). Feature selection under fairness constraints. *In Proceedings of the 37th ACM/SIGAPP Symposium on Applied Computing*, SAC '22, page 1125–1127, New York, NY, USA. Association for Computing Machinery.

[Drushku *et al.*, 2019] DRUSHKU, K., ALIGON, J., LABROCHE, N., MARCEL, P. et PERALTA, V. (2019). Interest-based recommendations for business intelligence users. *Information Systems*, 86:79–93.

[Dua et Graff, 2017] DUA, D. et GRAFF, C. (2017). UCI machine learning repository.

[Dwork *et al.*, 2012] DWORK, C., HARDT, M., PITASSI, T., REINGOLD, O. et ZEMEL, R. (2012). Fairness through awareness. *In 12). Association for Computing Machinery, New York, NY, USA*, pages 214–226. Proceedings of the 3rd Innovations in Theoretical Computer Science Conference (ITCS.

[Dwork *et al.*, 2018] DWORK, C., IMMORLICA, N., KALAI, A. T. et LEISERSON, M. (2018). Decoupled classifiers for group-fair and efficient machine learning. *In* FRIEDLER, S. A. et WILSON, C., éditeurs : *Proceedings of the 1st Conference on Fairness, Accountability and Transparency*, volume 81 de *Proceedings of Machine Learning Research*, pages 119–133. PMLR.

[El Malki *et al.*, 2020] EL MALKI, N., CUGNY, R., TESTE, O. et RAVAT, F. (2020). Decwa: Density-based clustering using wasserstein distance. *In Proceedings of the 29th ACM International Conference on Information amp; Knowledge Management*, CIKM '20, page 2005–2008, New York, NY, USA. Association for Computing Machinery.

[Elyan *et al.*, 2021] ELYAN, E., MORENO-GARCIA, C. F. et JAYNE, C. (2021). Cdsmote: class decomposition and synthetic minority class oversampling technique for imbalanced-data classification. *Neural computing and applications*, 33(7):2839–2851.

[Estévez *et al.*, 2009] ESTÉVEZ, P. A., TESMER, M., PEREZ, C. A. et ZURADA, J. M. (2009). Normalized mutual information feature selection. *IEEE Transactions on neural networks*, 20(2):189–201.

[EU, 2002] EU, P. (2002). Directive 2002/58/ec of the european parliament and of the council of 12 july 2002 concerning the processing of personal data and the protection of privacy in the electronic communications sector (directive on privacy and electronic communications) (oj l 201, 31.7.2002, p. 37).

[Fang *et al.*, 2020] FANG, B., JIANG, M., CHENG, P.-y., SHEN, J. et FANG, Y. (2020). Achieving outcome fairness in machine learning models for social decision problems. *In* BESSIERE, C., éditeur : *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, pages 444–450. International Joint Conferences on Artificial Intelligence Organization. Main track.

[Farahani *et al.*, 2019] FARAHANI, B., BARZEGARI, M. et ALIEE, F. S. (2019). Towards collaborative machine learning driven healthcare internet of things. *In Proceedings of the International Conference on Omni-Layer Intelligent Systems*, COINS '19, page 134–140, New York, NY, USA. Association for Computing Machinery.

[Feldman *et al.*, 2015] FELDMAN, M., FRIEDLER, S. A., MOELLER, J., SCHEIDEGGER, C. et VENKATASUBRAMANIAN, S. (2015). Certifying and removing disparate impact. *In proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*, pages 259–268.

[Friedler *et al.*, 2019] FRIEDLER, S. A., SCHEIDEGGER, C., VENKATASUBRAMANIAN, S., CHOUDHARY, S., HAMILTON, E. P. et ROTH, D. (2019). A comparative study of fairness-enhancing interventions in machine learning. *In Proceedings of the conference on fairness, accountability, and transparency*, pages 329–338.

[Friedman et Nissenbaum, 1996] FRIEDMAN, B. et NISSENBAUM, H. (1996). Bias in computer systems. 14(3):330–347.

[Ghassami *et al.*, 2018] GHASSAMI, A., KHODADADIAN, S. et KIYAVASH, N. (2018). Fairness in supervised learning: An information theoretic approach. *In 2018 IEEE International Symposium on Information Theory (ISIT)*, pages 176–180. IEEE.

[Goh *et al.*, 2016] GOH, G., COTTER, A., GUPTA, M. et FRIEDLANDER, M. P. (2016). Satisfying real-world goals with dataset constraints. *Advances in Neural Information Processing Systems*, 29.

[Golub *et al.*, 1999] GOLUB, T. R., SLONIM, D. K., TAMAYO, P., HUARD, C., GAASENBEEK, M., MESIROV, J. P., COLLER, H. A., LOH, M. L., DOWNING, J. R., CALIGIURI, M. A., BLOOMFIELD, C. D. et LANDER, E. S. (1999). Molecular classification of

cancer: class discovery and class prediction by gene expression monitoring. *Science*, 286 5439:531–7.

[Grabmeier et Lambe, 2007] GRABMEIER, J. et LAMBE, L. A. (2007). Decision trees for binary classification variables grow equally with the gini impurity measure and pearson's chi-square test. *Int. J. Bus. Intell. Data Min.*, 2:213–226.

[Gu *et al.*, 2009] GU, Q., ZHU, L. et CAI, Z. (2009). Evaluation measures of the classification performance of imbalanced data sets. *In International symposium on intelligence computation and applications*, pages 461–471.

[Gunning et Aha, 2019] GUNNING, D. et AHA, D. (2019). Darpa's explainable artificial intelligence (xai) program. *AI magazine*, 40(2):44–58.

[Gutkin *et al.*, 2009] GUTKIN, M., SHAMIR, R. et DROR, G. (2009). Slimpls: A method for feature selection in gene expression-based disease classification. *PLOS ONE*, 4(7):1–12.

[Guyon et Elisseeff, 2003] GUYON, I. et ELISSEEFF, A. (2003). An introduction to variable and feature selection. *J. Mach. Learn. Res.*, 3:1157–1182.

[Guyon *et al.*, 2008] GUYON, I., GUNN, S., NIKRAVESH, M. et ZADEH, L. A. (2008). *Feature extraction: foundations and applications*, volume 207. Springer.

[Guégan et Hassani, 2018] GUÉGAN, D. et HASSANI, B. (2018). Regulatory learning: How to supervise machine learning models? an application to credit scoring. *The Journal of Finance and Data Science*, 4(3):157–171.

[Hamerly et Elkan, 2003] HAMERLY, G. et ELKAN, C. (2003). Learning the k in k-means. *Advances in neural information processing systems*, 16.

[He *et al.*, 2008] HE, H., BAI, Y., GARCIA, E. et LI, S. (2008). Adasyn: Adaptive synthetic sampling approach for imbalanced learning. pages 1322 – 1328.

[He et Garcia, 2009] HE, H. et GARCIA, E. A. (2009). Learning from imbalanced data. *IEEE Trans. on Knowl. and Data Eng.*, 21(9):1263–1284.

[He *et al.*, 2019] HE, W., LI, H. et LI, J. (2019). Ensemble feature selection for improving intrusion detection classification accuracy. *Proceedings of the 2019 International Conference on Artificial Intelligence and Computer Science*.

[Heidari et Krause, 2018] HEIDARI, H. et KRAUSE, A. (2018). Preventing disparate treatment in sequential decision making. *In IJCAI*, pages 2248–2254.

[Hertweck *et al.*, 2021] HERTWECK, C., HEITZ, C. et LOI, M. (2021). On the moral justification of statistical parity. *In Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '21, page 747–757, New York, NY, USA. Association for Computing Machinery.

[Hoyos-Osorio *et al.*, 2021] HOYOS-OSORIO, J., ALVAREZ-MEZA, A., DAZA-SANTACOLOMA, G., OROZCO-GUTIERREZ, A. et CASTELLANOS-DOMINGUEZ, G. (2021). Relevant information undersampling to support imbalanced data classification. *Neurocomputing*, 436:136–146.

[Hsu *et al.*, 2011] HSU, H.-H., HSIEH, C.-W. et LU, M.-D. (2011). Hybrid feature selection by combining filters and wrappers. *Expert Systems with Applications*, 38(7):8144–8150.

[Hu *et al.*, 2009] HU, S., LIANG, Y., MA, L.-T. et HE., Y. (2009). Msmote: Improving classification performance when training data is imbalanced. 2009 second international workshop on computer science and engineering 2 (2009). pages 13–17.

[Huang *et al.*, 2008] HUANG, G., MATTAR, M., BERG, T. et LEARNED-MILLER, E. (2008). Labeled faces in the wild: A database forstudying face recognition in unconstrained environments. *Tech. rep.*

[Huda *et al.*, 2018] HUDA, S., LIU, K., ABDELRAZEK, M., IBRAHIM, A., ALYAHYA, S., AL-DOSSARI, H. et AHMAD, S. (2018). An ensemble oversampling model for class imbalance problem in software defect prediction. *IEEE access*, 6:24184–24195.

[Ingold et Soper, 2016] INGOLD, D. et SOPER, S. (2016). Amazon does not consider the race of its customers.

[Iosifidis *et al.*, 2019] IOSIFIDIS, V., FETAHU, B. et NTOUTSI, E. (2019). Fae: A fairness-aware ensemble framework. 2019 ieee international conference on big data (big data) (2019). pages 108–110.

[Iosifidis et Ntoutsi, 2019] IOSIFIDIS, V. et NTOUTSI, E. (2019). Adafair: Cumulative fairness adaptive boosting. CIKM '19, page 781–790, New York, NY, USA. Association for Computing Machinery.

[Jensen et Neville, 2002] JENSEN, D. et NEVILLE, J. (2002). Linkage and autocorrelation cause feature selection bias in relational learning. *In ICML*, volume 2, pages 259–266.

[Ji *et al.*, 2020] JI, D., SMYTH, P. et STEYVERS, M. (2020). Can i trust my fairness metric? assessing fairness with unlabeled data and Bayesian inference. *In* LAROCHELLE, H., RANZATO, M., HADSELL, R., BALCAN, M. F. et LIN, H., éditeurs : *Advances*

in Neural Information Processing Systems, pages 18600–18612. Vol. 33. Curran Associates, Inc.

[Jiang *et al.*, 2019] JIANG, B., WU, X., YU, K. et CHEN, H. (2019). Joint semi-supervised feature selection and classification through bayesian approach. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):3983–3990.

[Jiang *et al.*, 2021] JIANG, Z., HAN, X., FAN, C., YANG, F., MOSTAFAVI, A. et HU., X. (2021). Generalized demographic parity for group fairness. *In International Conference on Learning Representations*.

[Jin *et al.*, 2006] JIN, X., XU, A., BIE, R. et GUO, P. (2006). Machine learning techniques and chi-square feature selection for cancer classification using sage gene expression profiles. *In International workshop on data mining for biomedical applications*, pages 106–115. Springer.

[Jing, 2001] JING, Y. (2001). A new test for the stable clustering hypothesis. *The Astrophysical Journal Letters*, 550:L125 – L128.

[Jović *et al.*, 2015] JOVIĆ, A., BRKIĆ, K. et BOGUNOVIĆ, N. (2015). A review of feature selection methods with applications. *In 2015 38th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, pages 1200–1205.

[Joyce, 2003] JOYCE, J. (2003). Bayes' theorem.

[Kamiran et Calders, 2011] KAMIRAN, F. et CALDERS, T. (2011). Data pre-processing techniques for classification without discrimination. *Knowledge and Information Systems*, 33.

[Kamishima *et al.*, 2012] KAMISHIMA, T., AKAHO, S., ASOH, H. et SAKUMA, J. (2012). Fairness-aware classifier with prejudice remover regularizer. *In Proceedings of the 2012th European Conference on Machine Learning and Knowledge Discovery in Databases - Volume Part II*, ECMLPKDD'12, page 35–50, Berlin, Heidelberg. Springer-Verlag.

[Khalid *et al.*, 2014] KHALID, S., KHALIL, T. et NASREEN, S. (2014). A survey of feature selection and feature extraction techniques in machine learning. *In 2014 science and information conference*, pages 372–378. IEEE.

[Khalili *et al.*, 2021] KHALILI, M. M., ZHANG, X., ABROSHAN, M. et SOJOUDI, S. (2021). Improving fairness and privacy in selection problems. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(9):8092–8100.

[Kim *et al.*, 2019] KIM, M. P., GHORBANI, A. et ZOU, J. (2019). Multiaccuracy: Black-box post-processing for fairness in classification. *In Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*, pages 247–254.

[Kostrzewa et Brzeski, 2017] KOSTRZEWA, D. et BRZESKI, R. (2017). The data dimensionality reduction in the classification process through greedy backward feature elimination. *In International Conference on Man–Machine Interactions*, pages 397–407. Springer.

[Kotsiantis *et al.*, 2006] KOTSIANTIS, S., KANELLOPOULOS, D., PINTELAS, P. *et al.* (2006). Handling imbalanced datasets: A review. *GESTS international transactions on computer science and engineering*, 30(1):25–36.

[Kraiem *et al.*, 2021] KRAIEM, I. B., GHOZZI, F., PÉNINOU, A., ROMAN-JIMENEZ, G. et TESTE, O. (2021). Human-interpretable rules for anomaly detection in time-series. *In EDBT*.

[Krawczyk *et al.*, 2021] KRAWCZYK, B., BELLINGER, C., CORIZZO, R. et JAPKOWICZ, N. (2021). Undersampling with support vectors for multi-class imbalanced data classification. *In 2021 International Joint Conference on Neural Networks (IJCNN)*, pages 1–7. IEEE.

[Kumar et Minz, 2014] KUMAR, V. et MINZ, S. (2014). Feature selection: a literature review. *SmartCR*, 4(3):211–229.

[Lahoti *et al.*, 2020] LAHOTI, P., BEUTEL, A., CHEN, J., LEE, K., PROST, F., THAIN, N., WANG, X. et CHI, E. (2020). Fairness without demographics through adversarially reweighted learning. *Advances in neural information processing systems*, 33(2020): 728–740.

[Lavalle *et al.*, 2020] LAVALLE, A., MATÉ, A. et TRUJILLO, J. (2020). An approach to automatically detect and visualize bias in data analytics.

[Lejeune *et al.*, 2020] LEJEUNE, C., MOTHE, J., SOUBKI, A. et TESTE, O. (2020). Shape-based outlier detection in multivariate functional data. *Knowledge-Based Systems*, 198:105960.

[Leung, 2007] LEUNG, K. M. (2007). Naive bayesian classifier. *Polytechnic University Department of Computer Science/Finance and Risk Engineering*, 2007:123–156.

[Li *et al.*, 2017] LI, J., CHENG, K., WANG, S., MORSTATTER, F., TREVINO, R. P., LIU, H. et TANG, J. (2017). 39 feature selection: A data perspective. *Feature Selection: A Data Perspective ACM Comput. Surv*, 9.

[Liaw *et al.*, 2002] LIAW, A., WIENER, M. *et al.* (2002). Classification and regression by randomforest. *R news*, 2(3):18–22.

[Liu *et al.*, 2017] LIU, Z., TANG, D., CAI, Y., WANG, R. et CHEN, F. (2017). A hybrid method based on ensemble welm for handling multi class imbalance in cancer microarray data. *Neurocomputing*, 266:641–650.

[Lloyd, 2018] LLOYD, K. (2018). Bias amplification in artificial intelligence systems. *arXiv preprint arXiv:1809.07842*.

[Lohia *et al.*, 2019] LOHIA, P. K., RAMAMURTHY, K. N., BHIDE, M., SAHA, D., VARSHNEY, K. R. et PURI, R. (2019). Bias mitigation post-processing for individual and group fairness. *In Icassp 2019-2019 ieee international conference on acoustics, speech and signal processing (icassp)*, pages 2847–2851. IEEE.

[Lu *et al.*, 2017] LU, H., CHEN, J., YAN, K., JIN, Q., XUE, Y. et GAO, Z. (2017). A hybrid feature selection algorithm for gene expression data classification. *Neurocomputing*, 256:56–62.

[Luong *et al.*, 2011] LUONG, B. T., RUGGIERI, S. et TURINI, F. (2011). K-nn as an implementation of situation testing for discrimination discovery and prevention. *In Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '11, page 502–510, New York, NY, USA. Association for Computing Machinery.

[Mao, 2002] MAO, K. (2002). Fast orthogonal forward selection algorithm for feature subset selection. *IEEE Transactions on Neural Networks*, 13(5):1218–1224.

[Martínez *et al.*, 2021] MARTÍNEZ, N., BERTRÁN, M., PAPADAKI, A., RODRIGUES, M. R. D. et SAPIRO, G. (2021). *Blind Pareto Fairness and Subgroup Robustness*. In ICML.

[Mary *et al.*, 2019] MARY, J., CALAUZÈNES, C. et KAROUI, N. E. (2019). Fairness-aware learning for continuous attributes and treatments. *In* CHAUDHURI, K. et SALAKHUTDINOV, R., éditeurs : *Proceedings of the 36th International Conference on Machine Learning*, volume 97 de *Proceedings of Machine Learning Research*, pages 4382–4391. PMLR.

[Mehrabi *et al.*, 2019] MEHRABI, N., MORSTATTER, F., SAXENA, N., LERMAN, K. et GALSTYAN, A. (2019). A Survey on Bias and Fairness in Machine Learning.

[Mehrabi *et al.*, 2021] MEHRABI, N., MORSTATTER, F., SAXENA, N., LERMAN, K. et GALSTYAN, A. (2021). A survey on bias and fairness in machine learning. *ACM Computing Surveys (CSUR)*, 54(6):1–35.

[Meil, 2003] MEIL, M. (2003). *Comparing Clusterings by the Variation of Information*. In COLT.

[Mishler *et al.*, 2021] MISHLER, A., KENNEDY, E. H. et CHOULDECHOVA, A. (2021). Fairness in risk assessment instruments: Post-processing to achieve counterfactual equalized odds. *In Proceedings of the 2021 ACM Conference on Fairness*, pages 386–400, and Transparency. Accountability.

[Mitchell *et al.*, 2021] MITCHELL, S., POTASH, E., BAROCAS, S., D'AMOUR, A. et LUM, K. (2021). Algorithmic fairness: Choices, assumptions, and definitions. *Annual Review of Statistics and Its Application*, 8(1):141–163.

[Mitchell, 1980] MITCHELL, T. M. (1980). *The need for biases in learning generalizations*. Department of Computer Science, Laboratory for Computer Science Research ....

[Moghimi *et al.*, 2018] MOGHIMI, A., YANG, C. et MARCHETTO, P. M. (2018). Ensemble feature selection for plant phenotyping: A journey from hyperspectral to multispectral imaging. *IEEE Access*, 6:56870–56884.

[Mohammed *et al.*, 2020] MOHAMMED, R., RAWASHDEH, J. et ABDULLAH, M. (2020). Machine learning with oversampling and undersampling techniques: overview study and experimental results. *In 2020 11th international conference on information and communication systems (ICICS)*, pages 243–248. IEEE.

[Mostert *et al.*, 2018] MOSTERT, W., MALAN, K. M. et ENGELBRECHT, A. P. (2018). Filter versus wrapper feature selection based on problem landscape features. *Proceedings of the Genetic and Evolutionary Computation Conference Companion*.

[Muthukrishnan et Rohini, 2016] MUTHUKRISHNAN, R. et ROHINI, R. (2016). Lasso: A feature selection technique in predictive modeling for machine learning. *In 2016 IEEE international conference on advances in computer applications (ICACA)*, pages 18–20. IEEE.

[Myung, 2003] MYUNG, I. J. (2003). Tutorial on maximum likelihood estimation. *Journal of Mathematical Psychology*, 47(1):90–100.

[Ndirangu *et al.*, 2019] NDIRANGU, D., MWANGI, W. et NDERU, L. (2019). An ensemble model for multiclass classification and outlier detection method in data mining. *Journal of Information Engineering and Applications*.

[Nie *et al.*, 2010] NIE, F., HUANG, H., CAI, X. et DING, C. (2010). Efficient and robust feature selection via joint _{2,1}-norms minimization. *In Proceedings of the 23rd International Conference on Neural Information Processing Systems - Volume 2*, NIPS'10, page 1813–1821, Red Hook, NY, USA. Curran Associates Inc.

[Nordhausen, 2009] NORDHAUSEN, K. (2009). The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Second Edition by Trevor Hastie, Robert Tibshirani, Jerome Friedman. *International Statistical Review*, 77(3):482–482.

[Ntoutsi *et al.*, 2020] NTOUTSI, E., FAFALIOS, P., GADIRAJU, U., IOSIFIDIS, V., NEJDL, W., VIDAL, M.-E., RUGGIERI, S., TURINI, F., PAPADOPOULOS, S., KRASANAKIS, E. *et al.* (2020). Bias in data-driven artificial intelligence systems—an introductory survey. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 10(3):e1356.

[Oneto et Chiappa, 2020] ONETO, L. et CHIAPPA, S. (2020). Fairness in machine learning. *In Recent Trends in Learning From Data*, pages 155–196. Springer.

[Oneto *et al.*, 2019] ONETO, L., DONINI, M., ELDERS, A. et PONTIL, M. (2019). Taking advantage of multitask learning for fair classification. AIES '19, New York, NY, USA. Association for Computing Machinery.

[Osoba et IV, 2017] OSOBA, O. A. et IV, W. W. (2017). *An Intelligence in Our Image: The Risks of Bias and Errors in Artificial Intelligence*. RAND Corporation, Santa Monica, CA.

[Pal, 2005] PAL, M. (2005). Random forest classifier for remote sensing classification. *International journal of remote sensing*, 26(1):2005.

[Paparrizos *et al.*, 2011] PAPARRIZOS, I., CAMBAZOGLU, B. B. et GIONIS, A. (2011). Machine learned job recommendation. *In Proceedings of the Fifth ACM Conference on Recommender Systems*, RecSys '11, page 325–328, New York, NY, USA. Association for Computing Machinery.

[Park *et al.*, 2021] PARK, S., HWANG, S., KIM, D. et BYUN, H. (2021). Learning disentangled representation for fair facial attribute classification via fairness-aware information alignment. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(3):2403–2411.

[Peng *et al.*, 2005] PENG, H., LONG, F. et DING, C. (2005). Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(8):1226–1238.

[Pessach et Shmueli, 2020] PESSACH, D. et SHMUELI, E. (2020). Algorithmic fairness.

[Petersen *et al.*, 2021] PETERSEN, F., MUKHERJEE, D., SUN, Y. et YUROCHKIN, M. (2021). Post-processing for individual fairness. *Advances in Neural Information Processing Systems*, 34.

[Pleiss *et al.*, 2017] PLEISS, G., RAGHAVAN, M., WU, F., KLEINBERG, J. et WEINBERGER, K. Q. (2017). On fairness and calibration. *Advances in neural information processing systems*, 30:2017.

[Powers, 2020] POWERS, D. M. (2020). Evaluation: from precision, recall and f-measure to roc, informedness, markedness and correlation. preprint, arXiv.

[Pudil *et al.*, 1994] PUDIL, P., NOVOVICOVÁ, J. et KITTLER, J. (1994). Floating search methods in feature selection. *Pattern Recognit. Lett.*, 15:1119–1125.

[Qin et Tang, 2019] QIN, Z. T. et TANG, J. (2019). Deep reinforcement learning with applications in transportation. *In Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '19, page 3201–3202, New York, NY, USA. Association for Computing Machinery.

[Quinlan, 2014] QUINLAN, J. R. (2014). *C4. 5: programs for machine learning*. Elsevier.

[Raileanu et Stoffel, 2004] RAILEANU, L. E. et STOFFEL, K. (2004). Theoretical comparison between the gini index and information gain criteria. *Annals of Mathematics and Artificial Intelligence*, 41(1):77–93.

[Raizada et Lee, 2013] RAIZADA, R. et LEE, Y. (2013). Smoothness without smoothing: Why gaussian naive bayes is not naive for multi-subject searchlight studies. *PloS one*, 8:e69566.

[Raschka, 2016] RASCHKA, S. (2016). Mlxtend.

[Raschka, 2018] RASCHKA, S. (2018). Model evaluation, model selection, and algorithm selection in machine learning. *arXiv preprint arXiv:1811.12808*.

[Räz, 2021] RÄZ, T. (2021). Group fairness: Independence revisited. *In Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '21, page 129–137, New York, NY, USA. Association for Computing Machinery.

[Reddy *et al.*, 2020] REDDY, G. T., REDDY, M. P. K., LAKSHMANNA, K., KALURI, R., RAJPUT, D. S., SRIVASTAVA, G. et BAKER, T. (2020). Analysis of dimensionality reduction techniques on big data. *IEEE Access*, 8:54776–54788.

[Ren *et al.*, 2008] REN, J., QIU, Z., FAN, W., CHENG, H. et YU, P. S. (2008). Forward semi-supervised feature selection. *In PAKDD*.

[Ring et Eskofier, 2016] RING, M. et ESKOFIER, B. M. (2016). An approximation of the gaussian rbf kernel for efficient classification with svms. *Pattern Recogn. Lett.*, 84(C):107–113.

[Ristanoski *et al.*, 2013] RISTANOSKI, G., LIU, W. et BAILEY, J. (2013). Discrimination aware classification for imbalanced datasets. *In Proceedings of the 22nd ACM international conference on Information & Knowledge Management*, pages 1529–1532.

[Rokach et Maimon, 2005] ROKACH, L. et MAIMON, O. (2005). Decision trees. *In Data mining and knowledge discovery handbook*, pages 165–192. Springer.

[Romei et Ruggieri, 2013] ROMEI, A. et RUGGIERI, S. (2013). A multidisciplinary survey on discrimination analysis. *The Knowledge Engineering Review*, 29:582 – 638.

[Salazar *et al.*, 2021] SALAZAR, T., SANTOS, M. S., ARAÚJO, H. et ABREU, P. H. (2021). Fawos: Fairness-aware oversampling algorithm based on distributions of sensitive attributes. *IEEE Access*, 9:81370–81379.

[Salzberg, 1994] SALZBERG, S. (1994). C4.5: Programs for machine learning by j. ross quinlan. morgan kaufmann publishers, inc., 1993. *Machine Learning*, 16:235–240.

[Samek et Müller, 2019] SAMEK, W. et MÜLLER, K.-R. (2019). Towards explainable artificial intelligence. *In Explainable AI: interpreting, explaining and visualizing deep learning*, pages 5–22. Springer.

[Sánchez-Marono *et al.*, 2007] SÁNCHEZ-MARONO, N., ALONSO-BETANZOS, A. et TOMBILLA-SANROMÁN, M. (2007). Filter methods for feature selection–a comparative study. *In International Conference on Intelligent Data Engineering and Automated Learning*, pages 178–187. Springer.

[Schapire, 2013] SCHAPIRE, R. E. (2013). Explaining adaboost. *In Empirical inference. , , Heidelberg*, pages 37–52. Springer, Berlin.

[Shameem et Ferdous, 2009] SHAMEEM, M.-U.-S. et FERDOUS, R. (2009). An efficient k-means algorithm integrated with jaccard distance measure for document clustering. 2009 first asian himalayas international conference on internet (2009). pages 1–6.

[Sha'Abani *et al.*, 2020] SHA'ABANI, M., FUAD, N., JAMAL, N. et ISMAIL, M. (2020). knn and svm classification for eeg: a review. *InECCE2019*, pages 555–565.

[Shortliffe *et al.*, 1979] SHORTLIFFE, E., BUCHANAN, B. et FEIGENBAUM, E. (1979). Knowledge engineering for medical decision making: A review of computer-based clinical decision aids. *Proceedings of the IEEE*, 67(9):1207–1224.

[Singh et Joachims, 2018] SINGH, A. et JOACHIMS, T. (2018). Fairness of exposure in rankings. *In Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery Data Mining*, pages 2219–2228.

[Singhi et Liu, 2006] SINGHI, S. K. et LIU, H. (2006). Feature subset selection bias for classification learning. *In Proceedings of the 23rd international conference on Machine learning*, pages 849–856.

[Suresh et Guttag, 2019] SURESH, H. et GUTTAG, J. V. (2019). A framework for understanding unintended consequences of machine learning. *ArXiv*, abs/1901.10002.

[Sutton, 2005] SUTTON, C. D. (2005). Classification and regression trees, bagging, and boosting. *Handbook of statistics*, 24:303–329.

[Tangirala, 2020] TANGIRALA, S. (2020). Evaluating the impact of gini index and information gain on classification using decision tree classifier algorithm. *International Journal of Advanced Computer Science and Applications*, 11(2):612–619.

[Thanh *et al.*, 2011] THANH, B. L., RUGGIERI, S. et TURINI, F. (2011). k-nn as an implementation of situation testing for discrimination discovery and prevention. *In KDD*.

[Tharmakulasingam *et al.*, 2020] THARMAKULASINGAM, M., TOPAL, C., FERNANDO, A. et LA RAGIONE, R. (2020). Backward feature elimination for accurate pathogen recognition using portable electronic nose. *In 2020 IEEE International Conference on Consumer Electronics (ICCE)*, pages 1–5. IEEE.

[Ullah *et al.*, 2017] ULLAH, A., QAMAR, U., KHAN, F. H. et BASHIR, S. (2017). Dimensionality reduction approaches and evolving challenges in high dimensional data. *In Proceedings of the 1st International Conference on Internet of Things and Machine Learning*, IML '17, New York, NY, USA. Association for Computing Machinery.

[Venkatesh et Anuradha, 2019] VENKATESH, B. et ANURADHA, J. (2019). A review of feature selection and its methods. *Cybernetics and Information Technologies*, 19:26 – 3.

[Verma et Rubin, 2018] VERMA, S. et RUBIN, J. (2018). Fairness definitions explained. *In Proceedings of the International Workshop on Software Fairness*, FairWare '18, page 1–7, New York, NY, USA. Association for Computing Machinery.

[Viloria *et al.*, 2020] VILORIA, A., LEZAMA, O. B. P. et MERCADO-CARUZO, N. (2020). Unbalanced data processing using oversampling: Machine learning. *Procedia Computer Science*, 175:108–113.

[Visalakshi et Radha, 2014] VISALAKSHI, S. et RADHA, V. (2014). A literature review of feature selection techniques and applications: Review of feature selection in data mining. *In 2014 IEEE International Conference on Computational Intelligence and Computing Research*, pages 1–6. IEEE.

[von Luxburg, 2010] von LUXBURG, U. (2010). Clustering stability: An overview. *Found. Trends Mach. Learn.*, 2(3):235–274.

[Wadsworth *et al.*, 2018] WADSWORTH, C., VERA, F. et PIECH, C. (2018). Achieving fairness through adversarial learning: an application to recidivism prediction. *ArXiv*, abs/1807.00199.

[Wan *et al.*, 2020] WAN, H., WANG, H., SCOTNEY, B., LIU, J. et NG, W. W. Y. (2020). Within-class multimodal classification. *Multimedia Tools Appl.*, 79(39–40): 29327–29352.

[Wang et Chen, 2021] WANG, H.-H. et CHEN, C.-P. (2021). Comparison of chi-square test and representative decision tree in features that influence vehicle style. *International Journal of Machine Learning and Computing*.

[Wang *et al.*, 2020] WANG, M., TAO, X. et HAN, F. (2020). A new method for redundancy analysis in feature selection. *In 2020 3rd International Conference on Algorithms, Computing and Artificial Intelligence*, ACAI 2020, New York, NY, USA. Association for Computing Machinery.

[Wang *et al.*, 2017] WANG, P., LI, Y., CHEN, B., HU, X., YAN, J., XIA, Y. et YANG, J. (2017). Proportional hybrid mechanism for population based feature selection algorithm. *International Journal of Information Technology & Decision Making*, 16(05):1309–1338.

[Wang et Wang, 2014] WANG, T. et WANG, D. (2014). Why amazon's ratings might mislead you: The story of herding effects. *Big Data*, 2:196–204.

[Webb *et al.*, 2010] WEBB, G. I., KEOGH, E. et MIIKKULAINEN, R. (2010). Naïve bayes. *Encyclopedia of machine learning*, 15:713–714.

[Wolf et Shashua, 2003] WOLF et SHASHUA (2003). Feature selection for unsupervised and supervised inference: the emergence of sparsity in a weighted-based approach. *In Proceedings Ninth IEEE International Conference on Computer Vision*, pages 378–384 vol.1.

[Xu *et al.*, 2010] XU, Z., KING, I., LYU, M. R.-T. et JIN, R. (2010). Discriminative semi-supervised feature selection via manifold regularization. *IEEE Transactions on Neural networks*, 21(7):1033–1047.

[Yan *et al.*, 2020] YAN, S., te KAO, H. et FERRARA, E. (2020). Fair class balancing: Enhanc- ing model fairness without observing sensitive attributes. *In 20). Association for Computing Machinery, New York, NY, USA*. Proceedings of the 29th ACM International Conference on Information amp; Knowledge Manage- ment (CIKM.

[Yao et Wang, 2021] YAO, B. et WANG, L. (2021). An improved under-sampling imbalanced classification algorithm. *In 2021 13th International Conference on Measuring Technology and Mechatronics Automation (ICMTMA)*, pages 775–779. IEEE.

[Ye *et al.*, 2021] YE, Z., CHEN, Y. et ZHENG, H. (2021). Understanding the effect of bias in deep anomaly detection. *In* ZHOU, Z.-H., éditeur : *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, pages 3314–3320. International Joint Conferences on Artificial Intelligence Organization. Main Track.

[Yeom *et al.*, 2018] YEOM, S., DATTA, A. et FREDRIKSON, M. (2018). Hunting for discriminatory proxies in linear regression models. *In Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS'18, page 4573–4583, Red Hook, NY, USA. Curran Associates Inc.

[Yeom et Tschantz, 2021] YEOM, S. et TSCHANTZ, M. C. (2021). Avoiding disparity amplification under different worldviews. *In Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '21, page 273–283, New York, NY, USA. Association for Computing Machinery.

[You *et al.*, 2014] YOU, W., YANG, Z. et JI, G. (2014). Feature selection for high-dimensional multi-category data using pls-based local recursive feature elimination. *Expert Systems with Applications*, 41(4):1463–1475.

[Younsi *et al.*, 2019] YOUNSI, F.-Z., BOUNEKKAR, A., HAMDADOU, D. et BOUSSAID, O. (2019). Integration of multiple regression model in an epidemiological decision support system. *International Journal of Information Technology Decision Making*, 18.

[Yu et Liu, 2004] YU, L. et LIU, H. (2004). Efficient feature selection via analysis of relevance and redundancy. *J. Mach. Learn. Res.*, 5:1205–1224.

[Yu et Zhao, 2018] YU, S. et ZHAO, H. (2018). Rough sets and laplacian score based cost-sensitive feature selection. *PLOS ONE*, 13(6):1–23.

[Zafar *et al.*, 2017] ZAFAR, M. B., VALERA, I., GOMEZ RODRIGUEZ, M. et GUMMADI, K. P. (2017). Fairness beyond disparate treatment & disparate impact: Learning classification without disparate mistreatment. *In Proceedings of the 26th international conference on world wide web*, pages 1171–1180.

[Zemel *et al.*, 2013] ZEMEL, R., WU, Y., SWERSKY, K., PITASSI, T. et DWORK, C. (2013). Learning fair representations. *In International conference on machine learning*, pages 325–333. PMLR.

[Zeng *et al.*, 2009] ZENG, X., CHEN, Y.-W. et TAO, C. (2009). Feature selection using recursive feature elimination for handwritten digit recognition. *In 2009 Fifth International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, pages 1205–1208. IEEE.

[Zhang, 2004] ZHANG, H. (2004). The optimality of naïve bayes. *In In FLAIRS2004 conference*.

[Zhang *et al.*, 2020] ZHANG, J., LIN, Y., JIANG, M., LI, S., TANG, Y. et TAN, K. C. (2020). Multi-label feature selection via global relevance and redundancy optimization. *In IJCAI*, pages 2512–2518.

[Zhang *et al.*, 2016] ZHANG, L., WU, Y. et WU., X. (2016). A causal framework for discovering and removing direct and indirect discrimination. *CoRR abs/*, 7509:2016.

[Zhang *et al.*, 2018] ZHANG, S., CHENG, D., HU, R. et DENG, Z. (2018). Supervised feature selection algorithm via discriminative ridge regression. *World Wide Web*, 21(6):1545–1562.

[Zhang, 2016] ZHANG, Z. (2016). Introduction to machine learning: k-nearest neighbors. *Annals of translational medicine*, 4(11).

[Zhao et Liu,] ZHAO, Z. et LIU, H. *Semi-supervised Feature Selection via Spectral Analysis*, pages 641–646.

[Zhao et Liu, 2007] ZHAO, Z. et LIU, H. (2007). Spectral feature selection for supervised and unsupervised learning. *In Proceedings of the 24th International Conference on Machine Learning*, ICML '07, page 1151–1157, New York, NY, USA. Association for Computing Machinery.

[Zheng *et al.*, 2021] ZHENG, M., LI, T., ZHENG, X., YU, Q., CHEN, C., ZHOU, D., LV, C. et YANG, W. (2021). Uffdfr: Undersampling framework with denoising, fuzzy c-means clustering, and representative sample selection for imbalanced data classification. *Information Sciences*, 576:658–680.

[Zhu *et al.*, 2007] ZHU, Z., ONG, Y. et DASH, M. (2007). Wrapper–filter feature selection algorithm using a memetic framework. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 37:70–76.

[Zhuang et Dai, 2006] ZHUANG, L. et DAI, H. (2006). Reducing performance bias for unbalanced text mining. *In Proceedings of the Sixth IEEE International Conference on Data Mining - Workshops*, ICDMW '06, page 770–774, USA. IEEE Computer Society.

[Žliobaitė, 2017] ŽLIOBAITĖ, I. (2017). Measuring discrimination in algorithmic decision making. *Data Mining and Knowledge Discovery*, 31:1060–1089.