# FairnessDPLab: Evaluating Fairness Impact of Differential Privacy on Supervised AI Algorithms

by

**Aslı Atabek**

A Dissertation Submitted to the
Graduate School of Sciences and Engineering
in Partial Fulfillment of the Requirements for
the Degree of

Master of Science

in

Computer Science and Engineering


April 21, 2025

**FairnessDPLab: Evaluating Fairness Impact of Differential Privacy on Supervised AI Algorithms**

Koç University

Graduate School of Sciences and Engineering

This is to certify that I have examined this copy of a master's thesis by

**Aslı Atabek**

and have found that it is complete and satisfactory in all respects,
and that any and all revisions required by the final
examining committee have been made.

Committee Members:

_____

Assist. Prof. M. Emre Gürsoy (Advisor)

_____

Prof. Alptekin Küpçü

_____

Prof. Yücel Saygın

Date: _____

*Bu tezi, her zaman yanımda olan ve bana destek veren aileme adıyorum. Hayatım boyunca bana inandıkları, sabırla rehberlik ettikleri ve beni ben yapan değerleri aşıladıkları için onlara sonsuz teşekkür ederim. Ayrıca, bu yolculuk boyunca ilham kaynağım olan değerli hocalarıma ve dostlarıma, bilgiye olan tutkumun peşinden gitmem için beni cesaretlendirdikleri için minnettarım.*

# ABSTRACT

**FairnessDPLab: Evaluating Fairness Impact of Differential Privacy on Supervised AI Algorithms**

**Aslı Atabek**

**Master of Science in Computer Science and Engineering**

**April 21, 2025**

This study investigates the impact of differential privacy (DP) on fairness in supervised artificial intelligence (AI) algorithms through the development and evaluation of a benchmark platform, FairnessDPLab. The platform systematically examines the interplay between privacy preservation and fairness across three widely studied datasets: Adult, COMPAS, and German Credit. Utilizing both machine learning and deep learning models, the study explores the effects of varying privacy budgets and their implications on various fairness metrics.

The findings indicate that the balance between privacy and fairness can vary under different conditions, such as dataset characteristics or model choices. For instance, lower privacy budget values enhance privacy but may lead to undesirable changes in model accuracy and fairness measures. Deep learning models show higher sensitivity to privacy settings than traditional machine learning methods, such as logistic regression and random forests. The study also highlights the importance of carefully selecting privacy metrics, suggesting that they may need to be adapted based on the domain of application.

This work contributes to the growing body of literature on equitable AI systems by providing a robust framework for analyzing fairness under differential privacy constraints. The insights derived from this study offer practical guidelines for designing privacy-preserving AI systems that balance utility and fairness in diverse applications.

# ÖZETÇE

**FairnessDPLab: Diferansiyel Mahremiyetin Gözetimli YZ Algoritmaları Üzerindeki Adalet Etkisinin Analizi**

**Aslı Atabek**

**Bilgisayar Mühendisliği, Yüksek Lisans**

**21 Nisan 2025**

Bu tez, FairnessDPLab adlı bir karşılaştırma platformu ile gözetimli yapay zeka (YZ) algoritmalarında diferansiyel mahremiyetin adalet üzerindeki etkisini incelemektedir. Söz konusu platform, bu alanda yaygın biçimde kullanılan üç veri setini (Adult, COMPAS ve German Credit) kullanarak mahremiyet ile adalet arasındaki ilişkiyi sistematik bir şekilde değerlendirmektedir. Çalışma, makine öğrenmesi ve derin öğrenme modellerinden yararlanarak farklı mahremiyet bütçelerinin çeşitli adalet metrikleri üzerindeki sonuçlarını analiz etmektedir.

Elde edilen bulgular, mahremiyet ve adalet arasındaki dengenin veri seti ya da model tercihi gibi farklı kullanım koşullarıyla nasıl değişiklik gösterebileceğini raporlar. Örneğin, düşük verilen mahremiyet bütçe değerleri mahremiyeti artırırken, model doğruluğu ve adalet ölçümlerinde istenmeyen değişimlere yol açabilmektedir. Derin öğrenme modelleri, lojistik regresyon ve rastgele orman gibi geleneksel makine öğrenmesi yöntemlerine kıyasla mahremiyet ayarlarına karşı daha yüksek duyarlılık göstermektedir. Ayrıca, mahremiyet metriklerinin seçiminde dikkatli bir analiz yapılması gerektiği, incelenecek alana göre metrik seçiminde değişiklik yapılması gerektiği belirtilmiştir.

Bu çalışma, diferansiyel mahremiyet kısıtları altında adaleti incelemek için sağlam bir çerçeve sunarak, adil YZ sistemleri üzerine giderek büyüyen literatüre katkıda bulunmaktadır. Elde edilen analizler, farklı uygulamalarda fayda ve adaletin dengesini gözeten mahremiyet korumalı YZ sistemlerinin tasarımında pratik rehberlik sağlamaktadır.

# ACKNOWLEDGMENTS

I would like to express my deepest gratitude to my advisor, Assist. Prof. M. Emre Gürsoy, for his invaluable guidance, support, and encouragement throughout the course of this research. His expertise and insightful feedback were critical in shaping both the technical and conceptual foundations of this thesis. I also gratefully acknowledge the Security, Privacy and Data Engineering (SPADE) Research Lab at Koç University for providing a supportive research environment and valuable academic resources.

I would like to sincerely thank the members of my thesis jury Prof. Alptekin Küpçü and Prof. Yücel Saygın for their time, insightful comments, and constructive feedback, which greatly contributed to the refinement of this work.

I am also thankful to the faculty members of the Computer Engineering Department at Koç University for providing a stimulating academic environment and to all colleagues and friends who have contributed, directly or indirectly, to the completion of this work.

Finally, I would like to extend my heartfelt appreciation to my family for their unwavering support, patience, and belief in me throughout my academic journey.

# TABLE OF CONTENTS

# LIST OF TABLES

# LIST OF FIGURES

# ABBREVIATIONS

| | |
|---|---|
| AI | Artifical Intelligence |
| ML | Machine Learning |
| DL | Deep Learning |
| DP | Differential Privacy |
| NB | Naive Bayes |
| LR | Logistic Regression |
| RF | Random Forest |
| DT | Decision Tree |
| SGD | Stochastic Gradient Descent |
| PCA | Principal Component Analysis |
| RMSE | Root Mean Squared Error |
| SPD | Statistical Parity Difference |
| DI | Disparate Impact |
| EOD | Equal Opportunity Difference |
| AOD | Average Odds Difference |
| TI | Theil Index |
| FDRD | False Discovery Rate Difference |
| FNRD | False Negative Rate Difference |
| FPRD | False Positive Rate Difference |

# Chapter 1

# INTRODUCTION

Fair and responsible use of artificial intelligence (AI) has become a pressing concern as AI systems are increasingly deployed in high-stakes domains like finance, hiring, and criminal justice. In supervised learning tasks, models are typically trained on historical data to learn patterns that generalize to future predictions. However, this data often reflects systemic biases, which can inadvertently lead to discriminatory outcomes against certain demographic groups. Consequently, ensuring algorithmic fairness has become a core requirement for responsible AI deployment. Algorithmic fairness in supervised learning addresses the risk of biased or discriminatory outcomes by providing metrics and techniques to evaluate and mitigate bias in model predictions [Dwork et al., 2012, Mehrabi et al., 2019, Hardt et al., 2016, Chouldechova, 2017].

On the other hand, the privacy of individuals, whose data fuels supervised AI models, must be safeguarded. Recently, differential privacy (DP) has emerged as a gold standard approach for training models with formal privacy guarantees [Dwork et al., 2014, Abadi et al., 2016]. DP offers privacy guarantees that limit the influence of any single data point on the model's output, thereby minimizing the risk of re-identification. Yet, despite its strong theoretical appeal, applying DP in supervised learning raises important practical concerns related to model utility (accuracy) and fairness. In particular, combining privacy and fairness remains challenging.

The intersection of DP and fairness in AI is a relatively new research area, driven by the realization that these two objectives can interact in non-trivial ways [Fioretto et al., 2022]. Some recent research argues that DP promotes certain notions of fairness; therefore, the two are in alignment [Dwork et al., 2012, Khalili et al., 2021,

Jagielski et al., 2019]. In contrast, there also exist several works which argue that naively applying DP can exacerbate bias and unfairness. For instance, [Bagdasaryan et al., 2019, Pujol et al., 2020] showed that training DP models can degrade accuracy more for minority groups than for majority groups, yielding a disparate impact. Given these works, it appears that there is no clear consensus in the literature on whether DP and fairness are allies or foes.

This thesis contributes to the intersection of DP and fairness by introducing **FairnessDPLab**, a dedicated experimental platform for assessing fairness for differentially private supervised AI models. FairnessDPLab facilitates comprehensive experimentation by supporting both traditional machine learning algorithms (e.g., logistic regression, decision trees) and deep learning models under varying privacy budgets. It enables the evaluation of a wide range of fairness metrics, including well-known metrics such as statistical parity difference, equal opportunity difference, and disparate impact, across multiple benchmark datasets such as Adult, COMPAS, and German Credit. The platform integrates with existing fairness and privacy libraries (AIF360, DiffPrivLib, Opacus) and provides a modular architecture for extending to new datasets or models.

Using FairnessDPLab, the thesis then investigates how differentially private training affects model fairness under various conditions. Key variables considered include the strength of the privacy guarantee (i.e., privacy budget $\varepsilon$), model type (traditional ML vs deep learning), dataset characteristics, and hyperparameter settings. The results reveal nuanced trade-offs between privacy and fairness: while stronger privacy typically degrades accuracy, its impact on fairness is highly context-dependent. In some cases, noise introduced for privacy may mitigate biases; in others, it may amplify disparities – especially for unprivileged groups.

Ultimately, FairnessDPLab contributes a practical and open-source framework for auditing fairness under differential privacy constraints. In developing FairnessDPLab and conducting extensive experiments, the goal is to understand the interaction between fairness and privacy. This research is important because organizations deploying AI must often comply with privacy regulations (like GDPR) while also upholding standards of fairness and non-discrimination. Without clear insight into

how privacy measures affect fairness, practitioners could unintentionally harm one principle while trying to uphold the other. By providing a benchmarking tool and methodology, this thesis offers a way to navigate these dual objectives.

## *1.1 Thesis Contributions*

In short, this thesis makes the following main contributions:

- We introduce FairnessDPLab, a modular and extensible benchmarking platform that enables systematic evaluation of fairness under differential privacy constraints. The platform integrates state-of-the-art privacy-preserving algorithms and fairness metrics, and supports experimentation across various model architectures and datasets.

- Through extensive experiments on widely-used datasets (Adult, COMPAS, German Credit), we analyze how different levels of privacy (i.e., $\varepsilon$ budgets) affect both accuracy and multiple notions of fairness. We compare the fairness of traditional machine learning methods (e.g., logistic regression, random forests) and deep learning models under both private and non-private settings.

- We investigate how different fairness metrics respond to DP noise and how model type influences these outcomes. We highlight agreements and disagreements between various fairness metrics and show that some metrics are more robust to DP noise, whereas others may exhibit misleading fairness due to low accuracy.

## *1.2 Thesis Organization*

The remainder of this thesis is organized as follows. Chapter 2 provides an overview of prior research on fairness, differentially private learning, and the intersection between the two. Chapter 3 provides technical background related to fairness metrics and differentially private learning (both machine learning and deep learning). Chapter 4 presents the design and implementation of our proposed FairnessDPLab

platform. Chapter 5 provides an experimental analysis and discussion using FairnessDPLab, to empirically analyze the impacts of DP on fairness. Finally, Chapter 6 summarizes our key findings, discusses future work, and concludes this thesis.

# Chapter 2

# RELATED WORK

## *2.1 Fairness in Supervised AI*

The rapid adoption of AI in high-stakes decisions (such as credit, hiring, justice) has prompted extensive research on *algorithmic fairness* in supervised learning. Early work formalized different fairness criteria to detect and mitigate bias in model predictions [Dwork et al., 2012, Mehrabi et al., 2019]. These criteria generally fall into two categories: *individual fairness* (treating similar individuals similarly) and *group fairness* (ensuring parity of some statistic across different groups).

The notion of individual fairness, as introduced by [Dwork et al., 2012], requires that if two examples are similar (differ only in ways deemed irrelevant), they should receive similar predictions. Group fairness compares model outcomes between demographic groups defined by protected attributes (e.g., race or gender). For instance, *demographic parity* (or statistical parity) requires the overall selection rates to be similar across groups [Hardt et al., 2016]. A more nuanced notion, *equality of opportunity*, focuses on equalizing true positive rates across groups [Hardt et al., 2016]. In addition to these, many other fairness notions exist (e.g., predictive parity, calibrated equity, causal fairness), reflecting the multidimensional nature of fairness. Researchers have noted that not all fairness criteria can be satisfied simultaneously and trade-offs are often required. For example, satisfying both demographic parity and equalized odds may be impossible unless the groups have identical real outcome distributions [Chouldechova, 2017]. As a result, fairness notion selection often depends on context and values.

To navigate this landscape, several surveys have catalogued fairness definitions and bias mitigation methods. For example, Mehrabi et al. [Mehrabi et al., 2019] provide a comprehensive survey on bias and fairness in ML, and Verma & Ru-

bin [Verma and Rubin, 2018] enumerate dozens of fairness metrics, clarifying relationships between them. These works highlight trends such as the shift from preprocessing techniques (e.g., rebalancing datasets or learning fair representations) to in-processing methods (integrating fairness constraints into model training) and post-processing (adjusting outputs to meet fairness criteria). Moreover, practical toolkits have emerged to help researchers and practitioners evaluate and improve fairness. One notable example is IBM's AI Fairness 360 toolkit [Bellamy et al., 2019], which provides a library of fairness metrics (including statistical parity difference, disparate impact ratio, equal opportunity difference, etc.) and a suite of bias mitigation algorithms.

Despite substantial progress, several open challenges remain in the fairness domain. One major gap is choosing the "right" fairness metric for a given application. Current literature provides many definitions, but there is no consensus on which notion best captures fairness in complex real-world scenarios (e.g., should we prioritize equal opportunity or overall parity in lending decisions?). This ties to the broader question of how to include contextual and normative considerations into fairness criteria (what is fair may depend on legal or ethical standards beyond math). There is also ongoing debate on the relationship between fairness and model complexity (for instance, are simpler models inherently easier to make fair or to audit for fairness?). Finally, fairness often does not exist in isolation, it intersects with other objectives like accuracy, privacy, and interpretability. Balancing these sometimes competing goals (e.g., avoiding large accuracy drops when enforcing fairness, or ensuring fairness while preserving privacy as discussed later) is an unresolved problem. In summary, while a rich set of fairness notions and techniques exists, there is a need to benchmark and monitor different notions from a unified platform, which is one of the main motivations of this thesis.

## *2.2 Differential Privacy in Supervised AI*

Privacy preservation has become a cornerstone of trustworthy AI, especially as models train on sensitive personal data. *Differential Privacy (DP)* has emerged as the

gold-standard formalism for protecting individual data in statistical analysis and machine learning [Dwork et al., 2014]. DP ensures that the inclusion or removal of any single datapoint in a dataset does not significantly affect the output of an algorithm. Intuitively, a differentially private algorithm injects randomness (noise) into its computations so that an observer cannot infer whether any particular individual's data was used, within a privacy budget $\varepsilon$ that quantifies the privacy guarantee.

In supervised learning, DP has been applied at various stages of the modeling pipeline. A basic approach is to add noise to the *output* of a trained model or query (for example, publishing model predictions or evaluation metrics with noise). However, more sophisticated methods integrate privacy during model *training* to yield a fully private model. A landmark contribution is the Differentially Private Stochastic Gradient Descent (DP-SGD) algorithm by [Abadi et al., 2016]. In DP-SGD, gradients computed on mini-batches are *clipped* to limit any single example's influence and then perturbed with noise. By carefully calibrating the noise to the clipping threshold and tracking the cumulative privacy loss across iterations (using techniques like the moment accountant), DP-SGD enables training deep neural networks with provable $(\varepsilon, \delta)$-DP guarantees. This method has become the *de facto standard* for private deep learning, allowing complex models (CNNs, RNNs, etc.) to learn from sensitive data while limiting memorization of individual examples. For instance, Google and Apple's teams have leveraged DP-SGD and related approaches to train models on user data (keyboard, location, etc.) without compromising privacy.

Besides gradient perturbation, researchers have developed DP versions of many *classical machine learning algorithms*. These typically rely on adding noise at different points of the training procedure to satisfy DP. For example, Chaudhuri et al. [Chaudhuri et al., 2011] proposed an objective perturbation method for logistic regression: before solving the optimization for model weights, they add a small random noise term to the loss function, which guarantees DP under certain convexity conditions. This *perturbed objective* approach yields a private logistic classifier with only a minor change to the learning algorithm. Similarly, there are DP adaptations of Naive Bayes classifiers [Ji et al., 2014] and other methods – often implemented by adding noise to sufficient statistics during training. For tasks like decision tree

splitting, the *Exponential Mechanism* (a DP technique for privately selecting a top candidate) is used to choose splits in a way that favors good splits while preserving privacy. Overall, these contributions illustrate that a wide range of ML models can be trained under DP by careful algorithmic modifications. Many of these techniques have been consolidated into *open-source libraries*: for instance, IBM's DiffPrivLib and Facebook's Opacus provide ready-to-use DP versions of common classifiers.

Applying DP in supervised learning invariably introduces a *utility-privacy trade-off*. The randomness (noise) required for privacy can degrade model accuracy and utility. For example, a smaller privacy budget $\varepsilon$ (stronger privacy) means more noise is added at each step, which can blur important patterns in data and lead to higher error rates. This trade-off has been observed widely: models trained with strict DP guarantees often have lower accuracy than non-private models trained on the same data. Finding ways to *mitigate the utility loss* is an active area of research. Some strategies include: tuning the noise distribution and clipping norms carefully; using *privacy accounting* methods (like Rényi DP or advanced composition) that let us spend the privacy budget more efficiently over many training iterations [Gong et al., 2020]; and leveraging public or less-sensitive data to pre-train models or guide private training (thus reducing the burden on private data). There is also work on adaptive clipping – dynamically adjusting how much to clip gradients per iteration or per layer to preserve as much signal as possible for learning.

The foundations of differential privacy are well-documented in literature [Dwork et al., 2014]. Dwork and Roth's tutorial is a canonical reference defining DP and basic mechanisms (Laplace, Gaussian noise for numeric queries, and the exponential mechanism for categorical outputs). Several surveys focus on *differentially private machine learning*. For example, [Gong et al., 2020] offer a comprehensive survey of DP in ML, categorizing techniques by the type of DP mechanism and learning paradigm. They review methods based on adding Laplace/Gaussian noise versus those using advanced mechanisms like objective perturbation, as well as use-case specific adaptations. Another survey by Ji et al. [Ji et al., 2014] explores the interplay between machine learning algorithms and DP, including theoretical results on the limits of learning with privacy. These surveys and others highlight common

challenges such as scaling DP to deep learning and improving model accuracy.

### *2.3 Differential Privacy and Fairness in Supervised AI*

The intersection of differential privacy and fairness in AI is a relatively new research area, driven by the realization that these two objectives can interact in complex ways [Fioretto et al., 2022]. On the surface, both aims seem aligned with ethical AI: privacy seeks to protect individuals from undue harm by limiting information disclosure, and fairness seeks to protect groups/individuals from biased decisions. However, recent work indicates that *DP and fairness can either align or conflict* depending on the context. In this section, we examine both perspectives and survey key findings on how privacy-preserving learning affects fairness, and vice versa.

Some researchers argue that ensuring privacy may implicitly promote certain notions of fairness. The intuition is that differential privacy (DP), by treating any two individuals (or datasets) the same up to some noise, enforces a form of uniformity that resonates with fairness. In fact, Dwork et al. (2012) drew a formal connection between DP and individual fairness, showing that a classifier that is $\varepsilon$-differentially private is also $\varepsilon$-individually fair under an appropriate metric [Dwork et al., 2012]. In other words, if an algorithm's output does not change much when one person's data is swapped with another's, it implies similar individuals are treated similarly — aligning with the individual fairness principle [Fioretto et al., 2022]. This result suggests a deep link: a DP mechanism can be interpreted as one that limits how much outputs can differ for any two distinct inputs, which is analogous to saying it limits how differently two individuals can be treated. Beyond individual fairness, there are specific scenarios where privacy mechanisms yield fair outcomes. [Khalili et al., 2021] demonstrate that using a DP Exponential Mechanism for selecting candidates can produce fair selections under certain conditions on the data distribution. Intuitively, if the data for each group is similar in quality (having comparable averages and variances of qualifications), the randomness injected by the DP selection process ends up giving groups a more equal chance of being selected. These aligned results are encouraging, as they hint that with careful design, one might achieve privacy and

fairness simultaneously. Indeed, a line of work on "fair private learning" attempts to incorporate fairness constraints into DP model training, aiming for the best of both worlds. For example, recent methods like FairDP [Jagielski et al., 2019] train separate models for each group with DP noise and then combine them to ensure no group is unduly hurt by DP noise. Such approaches report improved trade-offs, indicating that privacy-preserving models can be tuned to maintain or even improve fairness metrics.

In contrast to the optimistic view above, many studies have revealed that naively applying differential privacy can exacerbate bias and unfairness in supervised learning [Fioretto et al., 2022]. The root cause is that the noise added for privacy is not "fairness-aware" – it can disproportionately impact underrepresented or disadvantaged groups. For instance, [Bagdasaryan et al., 2019] showed that training deep models with DP-SGD can degrade accuracy more for minority groups than for majority groups. In their experiments, models trained on text data with DP had a significant performance drop on the African-American English subgroup, while the drop for the white English subgroup was smaller. This happens because the DP noise effectively acts like random perturbations to the training process; if one group's data pattern is harder to learn or less represented, the noise can overwhelm the signal for that group, leading to worse model predictions for that group. In other words, privacy comes at a higher cost for the minority group – thus, DP has a disparate impact on model accuracy [Bagdasaryan et al., 2019]. Another example is in decision or resource allocation tasks: [Pujol et al., 2020] found that when census data is privatized with DP noise and then used to distribute funds to districts, some districts (often those with smaller populations or more sensitive attributes) receive less than their fair share. The DP mechanism introduced random error in population counts, which led to systematic under-allocation for certain minority districts. These findings illustrate that applying privacy blindly can undermine fairness. As a result, a key insight from recent research is that DP can amplify existing inequities: if a model already has higher error on a minority class, adding DP noise may worsen that gap [Fioretto et al., 2022]. There are also theoretical works [Cummings et al., 2019] that examine conditions under which fairness and privacy are fundamentally

incompatible – for example, they showed that under some setups, any non-trivial DP guarantee will prevent achieving perfect equalized odds fairness, and vice versa.

In conclusion, the interplay between differential privacy and fairness is a controversial and double-edged sword. On one side, some works claim that DP and fairness are compatible, e.g., a well-designed approach can protect individual privacy and yield fairer outcomes. On the other side, some works claim that employing DP can unintentionally hurt minority groups and lead to unfair outcomes. Thus, there is a need for a *comprehensive empirical framework* to understand when privacy and fairness are compatible and when they conflict. This thesis aims to contribute to this emerging need by systematically evaluating how DP impacts fairness across different models and datasets, thereby offering insights and tools (e.g., FairnessDPLab) to navigate the privacy-fairness trade-off in supervised AI.

# Chapter 3

# BACKGROUND AND PRELIMINARIES

## *3.1 Problem Setting and Notation*

Consider a tabular dataset $\mathcal{D}$ which consists of a set of attributes (features) and a class label. One or more attributes are designated as *protected* attributes, meaning that they are associated with bias or fairness concerns. We denote by $D$ a random variable which represents privileged and unprivileged groups with respect to the protected attribute(s). We use $D = 1$ to represent the privileged group and $D = 0$ to represent the unprivileged group. Furthermore, we denote by random variable $Y$ the true outcome. $Y = 1$ indicates a positive (favorable) outcome whereas $Y = 0$ indicates a negative (unfavorable) outcome.

We exemplify these concepts using the sample bank loan dataset $\mathcal{D}$ given in Table 3.1. The set of attributes are {Loan Term, Job, Income, Race, Gender} and the class label is Approval Status. Race and Gender are designated as the protected attributes. Individuals with Race = "White" and Gender = "Male" are in the privileged group ($D = 1$), while remaining individuals are in the unprivileged group ($D = 0$). The positive outcome $Y = 1$ corresponds to Approval Status = "Approved". The negative outcome $Y = 0$ corresponds to Approval Status = "Rejected".

A supervised AI model $\mathcal{M}$ can be trained using $\mathcal{D}$, e.g., to predict if a new individual's request for a bank loan will be approved or rejected. Let $\widehat{Y}$ denote the outcome predicted by $\mathcal{M}$. The number of true positives, true negatives, false positives, and false negatives can be measured as:

- True Positives ($TP$): $Y = 1$ and $\widehat{Y} = 1$
- True Negatives ($TN$): $Y = 0$ and $\widehat{Y} = 0$

Table 3.1: Sample bank loan dataset $\mathcal{D}$ to illustrate protected attributes, privileged groups, and types of outcomes.

| Loan Term | Job | Income | Race | Gender | Approval Status |
|---|---|---|---|---|---|
| Long-Term | Engineer | \$70,000 | White | Male | Approved |
| Short-Term | Secretary | \$38,000 | White | Female | Rejected |
| Short-Term | Artist | \$92,000 | African-American | Male | Approved |
| Long-Term | Musician | \$43,000 | White | Male | Approved |
| Short-Term | Teacher | \$48,000 | White | Female | Rejected |
| Short-Term | Engineer | \$65,000 | Asian | Female | Approved |
| Long-Term | Machinist | \$54,000 | African-American | Male | Rejected |

- False Positives ($FP$): $Y = 0$ and $\widehat{Y} = 1$
- False Negatives ($FN$): $Y = 1$ and $\widehat{Y} = 0$

Afterwards, True Positive Rate (TPR), False Positive Rate (FPR), False Discovery Rate (FDR), and False Negative Rate (FNR) can be measured as follows.

$$TPR = \frac{TP}{TP + FN} \qquad FPR = \frac{FP}{FP + TN}$$

$$FDR = \frac{FP}{FP + TP} \qquad FNR = \frac{FN}{TP + FN}$$

For TPR, FPR, FDR and FNR, we use the subscript notation (e.g., $TPR_{D=1}$ or $FPR_{D=0}$) to mean that they are calculated using the privileged or unprivileged group only. For example, $FDR_{D=1}$ measures $FDR$ only for the privileged group; whereas $FNR_{D=0}$ measures $FNR$ only for the unprivileged group.

## *3.2 Fairness Notions*

Fairness in supervised AI can be measured using different fairness notions, each capturing a unique perspective on evaluation. In this section, we present the key fairness notions used in supervised learning.

**Statistical Parity Difference (SPD):** Statistical parity (also known as demographic parity) states that the likelihood of favorable outcome should be equal

regardless of whether an individual is in the privileged group or not [Dwork et al., 2012, Kusner et al., 2017]. For example, there should be equal likelihood to approve a bank loan request from a male client (privileged) versus a female client (unprivileged). The difference in statistical parity, i.e., difference in the likelihoods, can be calculated as:

$$SPD \;=\; \Pr[Y = 1|D = 0] - \Pr[Y = 1|D = 1]$$

According to SPD, highest fairness is achieved when SPD = 0. As SPD deviates from 0, unfairness increases.

**Disparate Impact (DI):** Disparate impact (DI) is similar to SPD; however, it measures fairness using a ratio of likelihoods rather than a difference [Chouldechova, 2017].

$$DI \;=\; \frac{\Pr[Y = 1|D = 0]}{\Pr[Y = 1|D = 1]}$$

Perfect fairness is achieved when DI = 1. When $DI$ is greater than 1, there is a positive bias towards the unprivileged group. When $DI$ is less than 1, there is a positive bias towards the privileged group.

**Consistency (CO):** Consistency is an individual fairness metric that measures how similar the labels are for similar instances [Zemel et al., 2013]. Let $(x_i, y_i) \in \mathcal{D}$ be a record with features $x_i$ and label $y_i$. Let $\Psi(x_i, k)$ denote the $k$-nearest neighbors of $x_i$ in $\mathcal{D}$. Then, consistency is defined as:

$$1 - \frac{1}{n}\sum_{i=1}^{n} |y_i - \frac{1}{k}\sum_{x_j \in \Psi(x_i,k)} y_j|$$

If all $k$-nearest neighbors of $x_i$ have the same label as $y_i$ (i.e., $y_j = y_i$), then consistency is maximized (equal to 1).

**False Discovery Rate Difference (FDRD):** FDR measures the ratio of false positives among all positive outcomes. FDRD calculates FDR separately for the privileged and unprivileged groups, i.e., $FDR_{D=1}$ and $FDR_{D=0}$. Then, the difference between them is computed as:

$$FDRD = FDR_{D=0} - FDR_{D=1}$$

When FDRD is close to 0, the privileged and unprivileged groups have similar false positive ratios, therefore the metric concludes that the two groups are treated more fairly.

**False Negative Rate Difference (FNRD):** FNR measures the ratio of false negatives among all instances that originally have true positive label (i.e., $FN+TP$). Similar to FDRD, FNRD metric calculates FNR separately for the privileged and unprivileged groups, i.e., $FNR_{D=1}$ and $FNR_{D=0}$. Then, FNRD is equal to the difference between the two:

$$FNRD = FNR_{D=0} - FNR_{D=1}$$

FNRD close to 0 implies that the privileged and unprivileged groups are treated more fairly.

**False Positive Rate Difference (FPRD):** This metric is also called "predictive equality difference". Perfect predictive equality requires the false positive rates (FPR) to be equal for the privileged and unprivileged groups. Thus, FPRD metric calculates FPR separately for the privileged and unprivileged groups, i.e., $FPR_{D=1}$ and $FPR_{D=0}$. Then, FPRD is equal to the difference between the two:

$$FPRD = FPR_{D=0} - FPR_{D=1}$$

Perfect FPRD is achieved when FPRD = 0. For example, in the context of bank loans, FPRD = 0 indicates that the probability of a white male (privileged) applicant being incorrectly predicted as "approved" is equal to the probability of an unprivileged applicant being incorrectly predicted as "approved".

**Theil Index (TI):** Recall that $(x_i, y_i) \in \mathcal{D}$ is a record with features $x_i$ and true label $y_i$. Let $\hat{y}_i$ denote the predicted label of this record, and let $b_i = \hat{y}_i - y_i + 1$ denote the benefit of the record. Then, Theil Index is mathematically defined as:

$$TI = \frac{1}{n} \sum_{i=1}^{n} \frac{b_i}{\mu} \ln \frac{b_i}{\mu}$$

where $n$ is the cardinality of the dataset and $\mu$ denotes the mean benefit of the records:

$$\mu = \frac{1}{n} \sum_{i=1}^{n} b_i$$

For the Theil Index (TI), a value of 0 implies perfect fairness. Higher the value of TI, higher the amount of unfairness.

**Equal Opportunity Difference (EOD):** Equal opportunity measures the difference between the TPRs of the privileged and unprivileged groups. Recall that $TPR_{D=1}$ and $TPR_{D=0}$ denote the TPR of the privileged and unprivileged group respectively. Then, Equal Opportunity Difference (EOD) is defined as:

$$EOD = TPR_{D=0} - TPR_{D=1}$$

Perfect EOD is achieved when EOD = 0. In the context of bank loans, EOD = 0 indicates that equal proportion of deserving individuals from the privileged and unprivileged groups will be predicted as "approved".

**Average Odds Difference (AOD):** Finally, Average Odds Difference (AOD) takes into account both the TPRs and FPRs of the privileged and unprivileged groups:

$$AOD = \frac{1}{2}\Big[(FPR_{D=0} - FPR_{D=1}) + (TPR_{D=0} - TPR_{D=1})\Big]$$

The output of this metric is interpreted as follows: A value of 0 implies both privileged and unprivileged groups have equal benefit. A value less than 0 implies a higher benefit for the privileged group, whereas a value greater than 0 implies a higher benefit for the unprivileged group.

### *3.3 Differential Privacy*

Differential Privacy (DP) provides a formal framework for quantifying the privacy guarantees of an algorithm operating on sensitive data [Dwork et al., 2014]. Consider a tabular dataset $\mathcal{D}$, where each row corresponds to an individual's data. Two datasets $\mathcal{D}$ and $\mathcal{D}'$ are called neighboring datasets if they differ in exactly one record. This means that one dataset can be obtained from the other by the addition or removal of an individual. Let $\mathcal{A}$ represent a randomized algorithm, and let $Range(\mathcal{A})$ denote the set of possible outcomes produced by $\mathcal{A}$. An algorithm $\mathcal{A}$ is said to satisfy $(\varepsilon, \delta)$-Differential Privacy, abbreviated as $(\varepsilon, \delta)$-DP, if for any two neighboring

datasets $\mathcal{D}$ and $\mathcal{D}'$, and for all possible outcomes $S \in Range(\mathcal{A})$, the following inequality holds:

$$\Pr[\mathcal{A}(\mathcal{D}) = S] \leq e^{\varepsilon} \cdot \Pr[\mathcal{A}(\mathcal{D}') = S] + \delta \tag{3.1}$$

In this equation, $\varepsilon$ and $\delta$ are the privacy parameters. When $\delta = 0$, the algorithm is said to satisfy $\varepsilon$-DP, which provides a stricter notion of privacy. The parameter $\varepsilon$ (often called the privacy loss parameter or the privacy budget) controls the amount of privacy leakage allowed by the algorithm. Smaller values of $\varepsilon$ indicate that the algorithm provides better privacy protection but may also lead to less accurate results. The parameter $\delta$ allows for a small probability of failure in privacy guarantees, meaning that with probability at most $\delta$, the algorithm may fail to be $\varepsilon$-DP. In many cases, $\delta$ is chosen to be a very small value, such as $10^{-3}$ or smaller.

## *3.4 Differentially Private Machine Learning (DPML)*

In the context of supervised AI, DP can be applied to algorithms that train models on sensitive datasets [Gong et al., 2020]. Let $\mathcal{A}$ denote a machine learning algorithm (such as logistic regression or decision tree). When $\mathcal{A}$ is designed to satisfy $(\varepsilon, \delta)$-DP, the resulting machine learning model is said to be trained under differential privacy, and this is referred to as Differentially Private Machine Learning (DPML). In DPML, the goal is to ensure that the presence or absence of any single individual's data record in the training set has a limited impact on the resulting model. This is typically achieved by introducing randomness into the training process, such as adding noise to the gradients or the output model parameters.

In this thesis, we work with 4 DPML algorithms that are implemented in IBM's DiffPrivLib library [Holohan et al., 2019]: Naive Bayes, Logistic Regression, Decision Tree, and Random Forest. Below, we briefly explain each model.

**Naive Bayes (NB):** Naive Bayes (NB), proposed in [Chan et al., 1979], is one of the most fundamental ML models for classification. Our work adapts the differentially private version of NB originally proposed in [Vaidya et al., 2013]. This approach adds calibrated Laplace noise (depending on whether the attribute is nu-

meric or categorical) when computing the Bayesian posterior probabilities necessary for constructing the NB model.

NB is based on Bayes' theorem, with the assumption that the features are conditionally independent given the target class. For each class $C_j$ and given a set of features $a_1, a_2, \ldots, a_n$, NB classifier calculates the posterior probability of each class and selects the class with the highest probability. The posterior probability is computed using Bayes' theorem where $P(a_1, a_2, \ldots, a_n \mid C_j)$ is likelihood and $P(C_j)$ is prior probability:

$$P(C_j \mid a_1, a_2, \ldots, a_n) = \frac{P(a_1, a_2, \ldots, a_n \mid C_j)P(C_j)}{P(a_1, a_2, \ldots, a_n)}$$

Since the denominator $P(a_1, a_2, \ldots, a_n)$ is the same for all classes and does not affect the final decision, this expression can be simplified to:

$$P(C_j \mid a_1, a_2, \ldots, a_n) \propto P(C_j) \prod_{i=1}^{n} P(a_i \mid C_j)$$

In Gaussian NB, it is assumed that each feature $a_i$ follows a Gaussian distribution for each class $C_j$. The likelihood for a continuous feature $a_i$ given class $C_j$ is given by the Gaussian probability density function:

$$P(a_i \mid C_j) = \frac{1}{\sqrt{2\pi\sigma_j^2}} \exp\left(-\frac{(a_i - \mu_j)^2}{2\sigma_j^2}\right)$$

where $\mu_j$ is the mean of the feature values for class $C_j$ and $\sigma_j^2$ is the variance of the feature values for class $C_j$.

Combining the prior probability $P(C_j)$ of the class and the product of the Gaussian likelihoods for each feature, the aim is to predict the class $C_j$ that maximizes the following expression:

$$\arg\max_{C_j} \left(\log P(C_j) + \sum_{i=1}^{n} \log P(a_i \mid C_j)\right)$$

where $P(C_j)$ is the prior probability of class $C_j$, and $P(a_i \mid C_j)$ is the Gaussian likelihood of feature $a_i$ given class $C_j$. By using the logarithms of probabilities, we convert the product of likelihoods into a sum, which simplifies the computations and avoids potential numerical underflow.

In short, Gaussian Naive Bayes calculates the posterior probability for each class based on the assumption that the features are normally distributed and conditionally independent. The class with the highest posterior probability is chosen as the prediction. The differentially private version of Naive Bayes modifies the standard Gaussian Naive Bayes by introducing noise to the model parameters (mean and variance) to protect the privacy of individual data points. The sensitivity of the mean $\mu_j$ of feature $a_i$ in class $C_j$ is computed as the maximum possible change in the mean if any single data point were added or removed from the dataset. If the feature values lie within the range $[\min(a_i), \max(a_i)]$, the sensitivity of the mean is:

$$\Delta\mu_j = \frac{\max(a_i) - \min(a_i)}{n}$$

where $n$ is the number of training examples in class $C_j$.

Similarly, the sensitivity of the variance $\sigma_j^2$ is determined, and Laplacian noise is added to both $\mu_j$ and $\sigma_j^2$ to preserve privacy. The updated mean $\mu_j'$ and variance $\sigma_j'^2$ are computed as follows:

$$\mu_j' = \mu_j + \text{Lap}\left(\frac{\Delta\mu_j}{\varepsilon}\right) \qquad \sigma_j'^2 = \sigma_j^2 + \text{Lap}\left(\frac{\Delta\sigma_j^2}{\varepsilon}\right)$$

where $\varepsilon$ is the privacy budget. The new parameters $\mu_j'$ and $\sigma_j'^2$ are then used in the Gaussian probability computations to ensure privacy while maintaining the functionality of the Naive Bayes classifier.

**Logistic Regression (LR):** Differentially private LR models are built by utilizing the differentially private empirical risk minimization method originally proposed in [Chaudhuri et al., 2011]. This method perturbs the objective function of LR proposed in [Yu et al., 2011] by adding Laplace noise. The perturbed objective function is solved using the L-BFGS optimization algorithm and $\ell_2$ norm penalty.

LR is a standard binary classification model. It predicts the probability that $y \in \{0, 1\}$ given $x \in \mathbb{R}^n$ via the *sigmoid function*:

$$\sigma(z) \;=\; \frac{1}{1 + e^{-z}}, \quad P(y = 1 \mid x) \;=\; \sigma\big(w^\top x\big).$$

We learn the parameters $w$ by minimizing the *regularized negative log-likelihood*:

$$\min_w \; \sum_{i=1}^{n} \log\big(1 + e^{-\,y_i\, w^\top x_i}\big) \;+\; \frac{\lambda}{2}\,\|w\|^2,$$

where the first term measures data fit and the second term penalizes large weights, controlling overfitting.

There are two common methods to achieve differential privacy in LR:

- Output Perturbation: This method adds noise directly to the final learned parameters $w$ to protect the privacy of individual data points. The noise is typically drawn from a Laplace distribution, where the amount of noise is calibrated based on the sensitivity of the LR model.

- Objective Perturbation: Instead of adding noise to the output, this method perturbs the objective function of LR by adding noise to the regularized empirical risk minimization problem. This noise is added before the optimization process, and then the perturbed objective function is minimized [Chaudhuri et al., 2011].

In this thesis, we construct differentially private LR by perturbing the objective function with a noise term proportional to the $\ell_2$ norm of the data. The optimization process still uses the L-BFGS-B solver, but DP is guaranteed through the addition of noise to the gradient. For objective perturbation of the logistic regression loss, we use:

$$\mathcal{L}_{DP}(w) \;=\; \sum_{i=1}^{n} \log\!\big(1 + e^{-\,y_i\, w^\top x_i}\big) \;+\; \frac{1}{2}\,\|w\|^2 \;+\; \frac{1}{n}\, b^\top w,$$

where $b$ is a noise vector (e.g., Laplace-distributed).

**Random Forest (RF):** We utilize the implementation of differentially private random decision forests from DiffPrivLib, which was originally proposed in [Fletcher and Islam, 2017]. While achieving DP, this method utilizes the Exponential Mechanism with reduced sensitivity, which provides higher accuracy compared to prior methods that retrieve perturbed counts of class labels.

Random forest (RF) builds multiple decision trees during training and outputs the class that is the mode of the classes from individual trees. The decision trees are constructed using a random subset of features at each split, reducing variance and improving prediction accuracy [Breiman, 2001]. A single decision tree is a

binary structure in which internal nodes represent tests on features, and the leaves represent class labels. Given a dataset $\{(x_i, y_i)\}_{i=1}^{n}$, where $x_i$ are the features and $y_i$ are the class labels, each tree is trained on a random sample from the data. For classification, the prediction is made by taking a majority vote from all the trees in the forest. This improves the model's robustness against overfitting and leads to better generalization.

A decision tree recursively partitions the feature space by maximizing an impurity measure such as Gini impurity or information gain. For a node $t$ with data $D_t$, the impurity function $I(t)$ is defined as:

$$I(t) = 1 - \sum_{k=1}^{K} p_k^2,$$

where $p_k$ is the proportion of samples belonging to class $k$ in the node, and $K$ is the number of classes. The goal at each split is to minimize the weighted impurity across child nodes.

The final random forest prediction for a new data point $x$ is given by:

$$\hat{y} = \text{mode}\left(\{T_m(x)\}_{m=1}^{M}\right),$$

where $T_m(x)$ represents the prediction of the $m$-th tree in the ensemble, and $M$ is the total number of trees. The random forest's overall prediction $\hat{y}$ is simply the most common (i.e., the majority vote, or mode) among all the tree outputs.

The differentially private version of random forests builds upon the traditional random forest approach by incorporating DP. In this model, DP is achieved through the use of the Exponential Mechanism and smooth sensitivity. In a differentially private random forest, each decision tree is built using random samples of data, but the queries (such as those used to split nodes) are modified to ensure privacy. Instead of querying exact class counts, the exponential mechanism is used to output the most frequent class label in each leaf node with high probability, thus reducing the sensitivity of the query. The privacy budget $\varepsilon$ is split across all trees, and each query consumes a portion of this budget.

The use of the Exponential Mechanism is as follows. For each leaf node, rather than directly counting class labels, the mechanism selects the most frequent label

with probability proportional to the following scoring function:

$$u(c, x) = \begin{cases} 1, & \text{if } c = \text{argmax}_k n_k, \\ 0, & \text{otherwise,} \end{cases}$$

where $n_k$ is the count of class $k$ in the leaf node $x$. This mechanism ensures that the output class label is private while maintaining the accuracy of the classifier.

To further improve utility, smooth sensitivity is applied. This method allows for a more efficient use of the privacy budget by adding less noise when the dataset has low sensitivity (i.e., when small changes in the dataset do not significantly impact the result). The smooth sensitivity $S^*(f, x)$ for the most frequent label query is given by:

$$S^*(f, x) = \max_{k=0,1,\ldots,n} e^{-k\varepsilon} S_k(x),$$

where $S_k(x)$ is the local sensitivity of the query.

**Decision Tree (DT):** Differentially private decision trees are natural building blocks of RFs. Thus, we can use the same methodology as RF, but instead of constructing an RF, we can construct a single DT. A single decision tree is constructed by recursively partitioning the data based on feature splits that maximize a chosen criterion such as Gini impurity or information gain. The resulting tree consists of internal nodes representing feature-based decisions and leaf nodes representing class labels or predicted values.

Differentially private decision trees are built using similar principles as in differentially private random forests, but instead of building an ensemble, we focus on constructing a single tree. The core idea remains to ensure that the process of choosing splits at each node does not reveal sensitive information about individual data points. The Exponential Mechanism is used to select the best feature and split point at each node with high probability, ensuring that DP is preserved. However, because there is only one tree (as opposed to an ensemble of trees in RF), the privacy budget $\varepsilon$ is not divided across multiple trees, allowing the full privacy budget to be used for selecting splits in this single tree.

## *3.5 Differentially Private Deep Learning (DPDL)*

Next, we focus on how differentially private deep learning (DPDL) can be performed. We first provide an overview of how Stochastic Gradient Descent (SGD) works. Then, we explain how the SGD algorithm is adapted to achieve differential privacy.

**Stochastic Gradient Descent (SGD).** Deep learning models are typically trained using an optimization algorithm called Stochastic Gradient Descent (SGD). In SGD, the goal is to minimize a loss function $L(\theta)$, where $\theta$ represents the model parameters. The loss function $L$ computes the difference between the predicted and true values for a given dataset, and it is minimized by adjusting the model parameters $\theta$ iteratively using gradients computed on random mini-batches of data [Abadi et al., 2016].

At each iteration $t$, the gradient of the loss $\nabla L$ with respect to the model parameters $\theta_t$ is computed as:

$$g_t = \nabla_\theta L(\theta_t, B_t)$$

where $B_t$ is a mini-batch of examples sampled from the dataset. The model parameters are updated using the following rule:

$$\theta_{t+1} = \theta_t - \eta g_t$$

where $\eta$ is the learning rate that controls the step size of the update.

**Differentially Private SGD (DP-SGD).** The main idea behind Differentially Private SGD (DP-SGD) is to ensure that the updates to the model parameters are insensitive to any individual data point in the training dataset [Abadi et al., 2016]. The DP-SGD algorithm works by clipping the gradient of each individual example in a mini-batch to limit its influence on the model update and then adding noise to the average gradient to mask the contribution of any specific data point. The steps are as shown in Algorithm 1.

After computing gradients, each gradient is clipped so that its norm does not exceed a fixed bound $C$. This clipping step limits the sensitivity of the gradient function — i.e., the maximum change in the gradient due to a single data point is

**Algorithm 1** Differentially Private SGD

**Input:** $\{x_1, \ldots, x_N\}$, $\mathcal{L}(\theta) = \frac{1}{N}\sum_i \mathcal{L}(\theta, x_i)$

**Params:** $\eta_t, \sigma$, $L$, $C$ (learning rate, noise scale, group size, grad norm bound)

Initialize $\theta_0$ randomly

**for** $t = 1$ to $T$ **do**

  Sample $L_t$ with prob $L/N$

  **for** $i \in L_t$ **do**

    $g_t(x_i) = \nabla_{\theta_t}\mathcal{L}(\theta_t, x_i)$ (compute gradient)

    $\overline{g}_t(x_i) = \frac{g_t(x_i)}{\max(1, \|g_t(x_i)\|_2/C)}$ (clip gradient)

    $\widetilde{g}_t \leftarrow \frac{1}{L}\Big(\sum_i \overline{g}_t(x_i) + \mathcal{N}(0, \sigma^2 C^2 I)\Big)$ (add noise)

  $\theta_{t+1} \leftarrow \theta_t - \eta_t\, \widetilde{g}_t$ (descent)

**Output:** $\theta_T$ and overall privacy cost $(\varepsilon, \delta)$ via privacy accounting

bounded by $C$. Next, the clipped gradients are averaged and Gaussian noise sampled from $\mathcal{N}\big(0,\ \sigma^2 C^2 I\big)$ is added, where $\sigma$ is the noise scale. The scale $\sigma$ is chosen based on the desired privacy guarantee; specifically, it is calibrated such that each update is $\big(O(q\,\varepsilon),\ q\,\delta\big)$ is differentially private, where:

$$q \;=\; \frac{L}{N}, \quad \text{with } L \text{ being the group size and } N \text{ the total number of examples,}$$

and $T$ is the total number of training steps.

Under the moments accountant [Abadi et al., 2016], the noise scale must satisfy

$$\sigma \;\geq\; c_2\, \frac{\sqrt{T\,\log\!\big(1/\delta\big)}}{q\,\varepsilon},$$

where $c_2$ is a constant from the analysis. This condition ensures that, when composing the privacy loss over all $T$ updates, the overall training process achieves $(\varepsilon, \delta)$-DP.

The privacy accountant is used to track the cumulative privacy loss. This method keeps track of the *log moments* of the privacy loss for each update and sums them over all steps. Since the accountant leverages higher-moment analysis, it yields a significantly tighter bound on the total privacy loss than standard composition

theorems. As a result, the overall $(\varepsilon, \delta)$ guarantee remains within acceptable limits even after many training steps [Abadi et al., 2016].

**Opacus: A practical framework for DPDL.** Opacus [Yousefpour et al., 2021] is a PyTorch-based library for implementing DP-SGD efficiently in deep learning models. The key component of Opacus is the `PrivacyEngine`, which wraps around existing PyTorch models, optimizers, and data loaders, automatically managing the addition of noise and clipping of gradients to ensure compliance with $(\varepsilon, \delta)$-DP. Furthermore, a challenge in DPDL is maintaining computational efficiency when applying DP mechanisms. Traditional DP-SGD approaches require computing per-sample gradients and processing them individually, which can introduce significant computational overhead. Opacus addresses this issue by using a vectorized approach to compute per-sample gradients more efficiently. This method leverages PyTorch's underlying tensor operations to reduce the runtime and memory overhead associated with DP-SGD. In addition to this optimization, Opacus supports features such as Poisson sampling and adaptive batch sizes, which further improve the performance of DP-SGD in deep learning models. In this thesis, we use Opacus to implement DPDL.

# Chapter 4

# DESIGN AND IMPLEMENTATION OF FAIRNESSDPLAB

A key contribution of this thesis is the design and implementation of FairnessDPLab, a comprehensive platform for benchmarking the impact of differential privacy (DP) on fairness in supervised machine learning (ML) and deep learning models. The purpose of this platform is to evaluate and measure how privacy-preserving mechanisms influence fairness metrics like statistical parity, equalized odds, and disparate impact across different datasets and ML and DL algorithms. The platform is also designed to enable customization of various models and privacy parameters, providing a flexible environment for testing different combinations of privacy levels and model setups. The implementation and documentation of FairnessDPLab are publicly available at: `https://github.com/aatabek/FairnessDPLab`.

## *4.1 Architectural Overview*

The architectural overview of FairnessDPLab is given in Figure 4.1. As illustrated in Figure 4.1, the architecture of FairnessDPLab is modular, designed around core components that interact seamlessly to facilitate fairness assessment and DP experimentation. Each component performs a specific role in the pipeline, allowing for flexibility and extensibility in terms of datasets, algorithms, and evaluation metrics. The four core components of FairnessDPLab are as follows.

**Data Ingestion Module** is responsible for loading and preprocessing datasets. FairnessDPLab comes ready with multiple datasets, which are frequently used in the fairness literature for experimentation (e.g., COMPAS, German Credit, Adult). The module ensures that each dataset can be configured with a specified target variable and sensitive attribute for fairness analysis. To ease fairness evaluation,

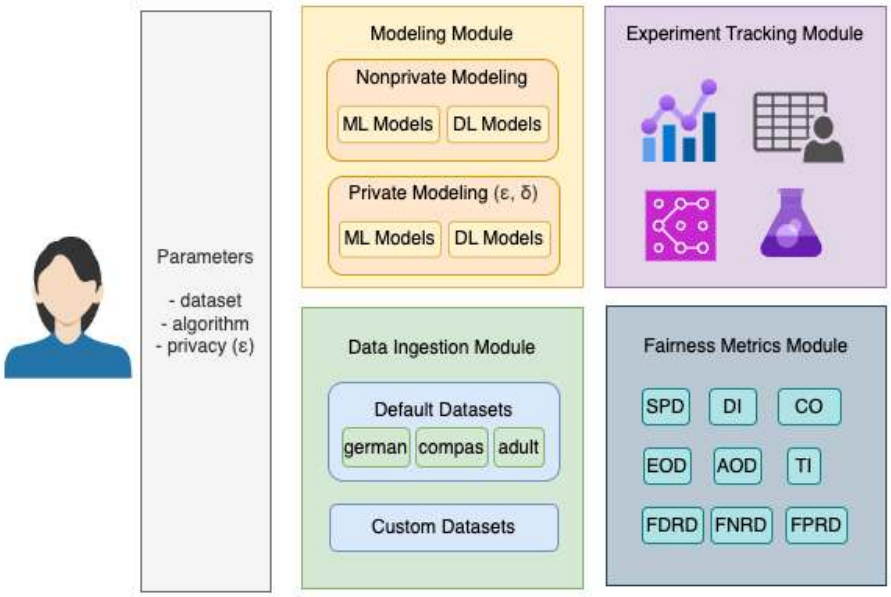


Figure 4.1: Overview of the FairnessDPLab architecture.

data is structured to be compliant with `AIF360` library. Additionally, the module allows for custom datasets to be integrated.

**Modeling Module** supports various machine learning algorithms (NB, LR, RF, DT) and deep learning through SGD and DP-SGD. Each model can be configured to run with or without differential privacy. For DPML and DPDL, this module uses libraries such as `Diffprivlib` and `Opacus` to ensure privacy.

**Fairness Metrics Module** employs the `AIF360` library to calculate various fairness metrics, including Statistical Parity Difference, Disparate Impact, Equal Opportunity Difference, and others. These metrics provide insights into how fairly the model treats different groups within the dataset.

**Experiment Tracking Module** stores and logs experimental results, including the configuration, model performance, and fairness metric outcomes. This enables tracking of different runs and comparing the effect of DP on fairness metrics across various experiments.

The design of FairnessDPLab supports two key principles:

- **Modularity**: Each core component operates independently and can be re-

Table 4.1: Key Characteristics of Adult Dataset

| **Characteristic** | **Value** |
|---|---|
| Number of Instances | 48,842 |
| Number of Attributes | 14 |
| Protected Attributes | race, sex |
| Target Variable | income |

placed or modified without affecting the overall functionality, allowing users to extend the platform for additional algorithms, datasets, or metrics.

- **Reproducibility**: The Experiment Tracking Module ensures that each experiment's configurations and results are recorded, allowing for reproducible research and experiments to be conducted with many repetitions to achieve statistical significance.

## *4.2 Data Ingestion*

The Data Ingestion Module in FairnessDPLab is responsible for loading and preprocessing datasets for both machine learning (ML) and deep learning (DL). The platform comes with three default datasets: Adult, COMPAS, and German Credit. Each dataset is carefully prepared to ensure consistency and fairness evaluation across experiments. Additional datasets can be integrated into FairnessDPLab in the future. The existing datasets are briefly summarized as follows.

**Adult.** The Adult dataset, also known as the "Census Income" dataset, is used to predict whether an individual's income exceeds \$50K/year based on census data [Kohavi and Becker, 1996]. As shown in Table 4.1, the Adult dataset contains 48,842 instances with 14 attributes. The primary protected attributes are race and sex, making it suitable for studying fairness in income prediction models across different demographic groups.

**COMPAS.** The COMPAS (Correctional Offender Management Profiling for Alternative Sanctions) dataset is used to predict recidivism. Table 4.2 summarizes the

Table 4.2: Key Characteristics of COMPAS Dataset

| **Characteristic** | **Value** |
|---|---|
| Number of Instances | 7,214 |
| Number of Attributes | 28 |
| Protected Attributes | race, sex |
| Target Variable | two year recid |

Table 4.3: Key Characteristics of German Credit Dataset

| **Characteristic** | **Value** |
|---|---|
| Number of Instances | 1,000 |
| Number of Attributes | 20 |
| Protected Attributes | sex, age |
| Target Variable | credit risk |

COMPAS dataset, which contains 7,214 instances with 28 attributes. The primary protected attributes are race and sex, making it particularly relevant for studying fairness in criminal justice predictions [Bao et al., 2021].

**German Credit.** The German Credit dataset is used for credit risk assessment [Hofmann, 1994]. As detailed in Table 4.3, the German Credit dataset contains 1,000 instances with 20 attributes. The primary protected attributes are sex and age, making it valuable for studying fairness in financial decision-making models.

*Preprocessing for Machine Learning*

For ML experiments, FairnessDPLab utilizes the AIF360 library to preprocess the datasets. This ensures that the data is in a format suitable for fairness analysis. The preprocessing steps include:

1. Loading the dataset using AIF360's built-in functions.
2. Specifying the protected attribute(s) for fairness evaluation.
3. Splitting the data into training and testing sets.

For example, the Adult dataset is processed using the following code:

```
dataset_orig = load_preproc_data_adult(['sex'])
dataset_orig_train, dataset_orig_test = dataset_orig.split
    ([0.7], shuffle=True)
```

This preprocessing step ensures that the data is structured correctly for AIF360's fairness metric calculations and that the protected attributes are properly identified.

*Preprocessing for Deep Learning*

Deep learning models require data in a specific tensor format, which differs from the AIF360 structure used in ML experiments. To accommodate this, FairnessDPLab implements a custom preprocessing pipeline for DL:

1. Loading the raw dataset from its source.
2. Applying necessary transformations (e.g., one-hot encoding for categorical variables, standardization for numerical variables).
3. Converting the processed data into PyTorch tensors [Paszke et al., 2019].
4. Creating DataLoader objects for efficient batching during training.

For instance, the Adult dataset preprocessing for DL can be done in the following manner:

```
numeric_transformer = Pipeline(steps=[('scaler',
    StandardScaler())])
categorical_transformer = Pipeline(steps=[('onehot',
    OneHotEncoder(handle_unknown='ignore'))])
preprocessor = ColumnTransformer(
  transformers=[
    ('num', numeric_transformer, numeric_features),
    ('cat', categorical_transformer, categorical_features)])
features = preprocessor.fit_transform(features)
```

*Custom Dataset Integration*

FairnessDPLab also supports the integration of custom datasets. Users can define their own dataset class, inheriting from AIF360's StandardDataset [Bellamy et al., 2019], and implement the necessary preprocessing steps. This flexibility allows researchers to extend the platform's capabilities to new domains and datasets while maintaining consistency in the remainder of the system.

## 4.3 Model Training and Evaluation

FairnessDPLab implements a comprehensive framework for training and evaluating both traditional machine learning (ML) and deep learning (DL) models, with and without differential privacy (DP). The platform's structure allows for a systematic comparison of various algorithms across different fairness and privacy settings.

*Training Setup*

The training process in FairnessDPLab is designed to accommodate both ML and DL algorithms, with separate implementations for non-private and differentially private versions of each algorithm. This structure allows for direct comparison between standard and privacy-preserving models. For each algorithm, there are two main functions: one for non-private and the other for private modeling. The function signatures are exemplified below. Note that the signatures are almost identical, with the main difference being the introduction of the privacy budget $\varepsilon$ in the DP version.

```
1    def algorithm(dataset_orig_train, dataset_orig_test,
       privileged_groups, unprivileged_groups)
```

```
1    def dp_algorithm(dataset_orig_train, dataset_orig_test,
       privileged_groups, unprivileged_groups, epsilon)
```

All ML and DL private and nonprivate model trainings happen inside the functions above. For ML models, the training process follows this general pattern: data preprocessing, model initialization (with or without privacy), model fitting, prediction

on test data. For DL models, the process is more complex, involving: data preprocessing and conversion to PyTorch tensors, model definition, loss function and optimizer setup, training loop (multiple epochs), prediction on test data.

*Non-Private Machine Learning*

FairnessDPLab standardizes the model training pipelines across all machine learning algorithms (NB, LR, DT, RF) by encapsulating the training and prediction processes within uniformly structured functions. As an example, consider the `logistic_regression` function. This function begins by extracting features and labels directly from FairnessDPLab training and test datasets. Next, an LR classifier is initialized with the solver set to `liblinear` and trained on the training data. Once trained, the classifier predicts labels on the test set, and FairnessDPLab helper function `create_dataset_pred` creates prediction results. Using the prediction results, FairnessDPLab function called `show_metric_results` is invoked to assess classification performance and fairness metric results. Finally, the function returns the training data, test data, and the classification metric results, in a format that ensures consistency and reproducibility across different experiments within FairnessDPLab.

```
1  def logistic_regression(dataset_orig_train,
2                          dataset_orig_test,
3                          unprivileged_groups,
4                          privileged_groups):
5      # Extract features and labels from FairnessDPLab training
           and test datasets
6      X_train, y_train = dataset_orig_train.features,
           dataset_orig_train.labels
7      X_test, y_test = dataset_orig_test.features,
           dataset_orig_test.labels
8
9      # Train and predict using sklearn functions
10     nonprivate_clf = LogisticRegression(solver="liblinear")
11     nonprivate_clf.fit(X_train, y_train)
```

```
    nonprivate_y_pred = nonprivate_clf.predict(X_test)
    # Generate a prediction dataset using FairnessDPLab
       helper
    dataset_pred = create_dataset_pred(
                        dataset_orig=dataset_orig_test,
                        y_pred=nonprivate_y_pred)

    # Calculate fairness and classification metrics using
       FairnessDPLab helper
    classified_metric_pred = show_metric_results(
        dataset_orig_test,
        dataset_pred,
        unprivileged_groups,
        privileged_groups)

    return X_train, y_train, X_test, y_test,
       classified_metric_pred
```

*Differentially Private Machine Learning*

For DPML, the same four algorithms are provided, each following a standardized modeling pipeline to maintain consistency throughout the platform. To illustrate this approach, we again take Logistic Regression as an example. This implementation closely mirrors its non-private counterpart, with the key distinction being the use of a differentially private classifier. Specifically, the model is initialized with a privacy parameter epsilon ($\varepsilon$), which controls the strength of the DP guarantee. To account for the randomness involved in stochastic processes, classical machine learning models are retrained $n$ times (with $n$ defaulting to 20) with different `random_state` values, providing robustness against variations in initialization and model fitting. This repeated retraining over multiple random states and a range of $\varepsilon$ values captures variability in model performance and fairness metrics, ensuring more robust and reliable experimental results.

```
def dp_logistic_regression(dataset_orig_train,
                           dataset_orig_test,
                           unprivileged_groups,
                           privileged_groups,
                           random_state,
                           epsilon=1):
    # Extract features and labels from FairnessDPLab training and test datasets
    X_train, y_train = dataset_orig_train.features, dataset_orig_train.labels
    X_test, y_test = dataset_orig_test.features, dataset_orig_test.labels

    # Train and predict using diffprivlib functions
    private_clf = dp.LogisticRegression(
                           random_state=random_state,
                           epsilon=epsilon)
    private_clf.fit(X_train, y_train)
    private_y_pred = private_clf.predict(X_test)
    # Generate a prediction dataset using FairnessDPLab helper
    dataset_pred = create_dataset_pred(
                           dataset_orig=dataset_orig_test,
                           y_pred=private_y_pred)

    # Calculate fairness and classification metrics using FairnessDPLab helper
    classified_metric_pred = show_metric_results(
        dataset_orig_test,
        dataset_pred,
        unprivileged_groups,
        privileged_groups)
```

```
28
29      return X_train, y_train, X_test, y_test,
            classified_metric_pred
```

*Non-Private Deep Learning*

For the non-private deep learning model, FairnessDPLab employs a simple feed-forward neural network designed to perform binary classification. The code snippet below initializes a network with one hidden layer of size 32, followed by a ReLU activation, and an output layer with two units (corresponding to the two classes). This architecture enables the network to capture non-linear relationships in the data while remaining computationally lightweight. The model is trained using a cross-entropy loss function (commonly used for classification tasks) and optimized with SGD at a learning rate of 0.01.

```
1  # Define the Simple Neural Network
2  def initialize_model(num_features):
3      model = nn.Sequential(
4          nn.Linear(num_features, 32),
5          nn.ReLU(),
6          nn.Linear(32, 2)
7      )
8      return model
9
10 # Define an optimizer
11 def initialize_optimizer(model):
12     optimizer = optim.SGD(model.parameters(), lr=0.01)
13     return optimizer
```

*Differentially Private Deep Learning*

To incorporate DP into the deep learning model, FairnessDPLab utilizes the Opacus library's `PrivacyEngine`, which injects noise into the gradients during the training

process. The DP-enhanced version maintains the same neural network architecture and overall training procedure as its non-private counterpart. The key modification lies in the model initialization, where the model parameters, optimizer, and data loader are wrapped with DP mechanisms that perform gradient clipping and noise addition. These privacy operations are controlled explicitly by the privacy budget parameters $\varepsilon$ and $\delta$. The following code snippet demonstrates the DPDL model's implementation and training procedure:

```
# Private deep learning training function with differential privacy
def dp_deep_learning(
    train_df,
    test_df,
    train_loader,
    test_loader,
    features_train,
    features_test,
    label,
    target_epsilon,
    sensitive_attribute,
    target_delta=1e-5,
    num_epochs=50,
):
    model = initialize_model(features_train.shape[1])
    optimizer = initialize_optimizer(model)
    privacy_engine = PrivacyEngine()
    criterion = nn.CrossEntropyLoss()

    # Privacy engine setup for differential privacy
    privacy_engine = PrivacyEngine()
    model, optimizer, train_loader = privacy_engine.
        make_private_with_epsilon(
            module=model,
```

```
            optimizer=optimizer,
            data_loader=train_loader,
            target_epsilon=target_epsilon,  # Privacy budget
            target_delta=target_delta, # Failure probability
            epochs=num_epochs,
            max_grad_norm=1.0  # Gradient clipping threshold
    )
    ...
    # train loop and evaluation
    ...

    return predictions
```

### *4.4 Fairness Assessment*

The Fairness Metrics Module in FairnessDPLab evaluates the trained model's performance in terms of a comprehensive set of fairness metrics. This module utilizes the `ClassificationMetric` class from the AIF360 library to compute various fairness and performance metrics. The core of this assessment is implemented in the `create_metric_entry` function, which computes: accuracy, balanced accuracy, SPD, DI, consistency, AOD, EOD, TI, FDRD, FNRD, and FPRD. These metrics provide a comprehensive view of the model's fairness across various dimensions, allowing researchers to assess different aspects of algorithmic fairness. The function also captures the algorithm type, privacy parameter $\varepsilon$, and random state, enabling thorough analysis of how these factors influence fairness outcomes. The metrics are stored in a structured DataFrame format, allowing for easy comparison and analysis across different experimental conditions and model configurations. By computing these metrics for both non-private and differentially private models, FairnessDPLab facilitates a nuanced comparison of how DP impacts not only model performance but also various fairness criteria.

### *4.5 Experiment Tracking*

The Experiment Tracking Module in FairnessDPLab is designed to systematically log all configurations, DP settings, parameters, and fairness metric results for each experiment. This comprehensive logging facilitates future analysis, comparison, and reproducibility of results. The core of this module is implemented through the `create_metric_entry` function and the use of pandas DataFrames for data storage. For each experiment, the `create_metric_entry` function stores:

1. Experiment identifier
2. ML or DL algorithm used
3. Epsilon value for differential privacy (with `infinity` indicating no privacy)
4. Random state for reproducibility
5. Fairness and performance metric results

The logged results are stored in a pandas DataFrame, initialized at the beginning of the experimental process:

```
classification_metrics_df =
   initialize_classification_metrics_df()
```

Throughout the experimental process, new entries are added to this DataFrame:

```
classification_metrics_df = classification_metrics_df.append(
    pd.Series(create_metric_entry(...), index=
       classification_metrics_df.columns),
    ignore_index=True)
```

# Chapter 5

# EXPERIMENTAL ANALYSIS AND DISCUSSION

In this chapter, we conduct a comprehensive set of experiments using the FairnessDPLab platform to empirically analyze the impacts of DP on fairness. Adult, German Credit, and COMPAS datasets were used in the experimentation, each utilizing **Gender** as the protected attribute. In all three datasets, the privileged group was defined as **Male** and the unprivileged group is **Female**, as aligned with existing fairness literature. All datasets were used in their original form with an 80%-20% train-test split ratio.

For each dataset, four different DPML models (LR, NB, DT, RF) were trained under varying privacy budgets $\varepsilon \in \{10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 10^{0}, 2, 10^{1}\}$. In addition, one DPDL model per dataset was trained with a fixed $\delta = 10^{-5}$ and varying $\varepsilon \in \{10^{-2}, 10^{-1}, 10^{0}, 2, 10^{1}\}$. Across these experiments, a total of **1710 model trainings** were performed. The fairness of each model was evaluated using eight established group and individual fairness metrics explained in Chapter 3 and included in FairnessDPLab, namely: SPD, DI, FDRD, FNRD, FPRD, TI, EOD, and AOD. This resulted in a total of **13,680 fairness metric computations** across all experiments.

In FairnessDPLab, each model has hyperparameters optimized for its non-private and DP implementations. The hyperparameters in the non-private and DP implementations are used consistently to ensure consistent comparisons between non-private and DP versions of the algorithms. For LR, `liblinear` solver was used. For RF, the number of trees is 10, and the default tree depth is 5. For DT, the default tree depth is again 5. For deep learning, considering the tabular nature of the datasets, a neural network composed of two layers was used: an input layer feeding into a hidden layer with 32 neurons activated by a ReLU function, and an output layer with two neurons corresponding to binary class predictions. This model uses

Table 5.1: Accuracy of differentially private ML and DL models under varying $\varepsilon$

| **Dataset** | **Model** | $\varepsilon = 10^{-4}$ | $\varepsilon = 10^{-3}$ | $\varepsilon = 10^{-2}$ | $\varepsilon = 10^{-1}$ | $\varepsilon = 10^{0}$ | $\varepsilon = 2$ | $\varepsilon = 10^{1}$ | **No DP** |
|---|---|---|---|---|---|---|---|---|---|
| Adult | Decision Tree | 0.65 | 0.76 | 0.77 | 0.78 | 0.78 | 0.78 | 0.78 | 0.78 |
| | Logistic Regression | 0.54 | 0.54 | 0.54 | 0.58 | 0.63 | 0.65 | 0.65 | 0.82 |
| | Naive Bayes | 0.53 | 0.68 | 0.72 | 0.77 | 0.80 | 0.80 | 0.79 | 0.79 |
| | Random Forest | 0.61 | 0.76 | 0.77 | 0.77 | 0.77 | 0.77 | 0.77 | 0.77 |
| | Deep Learning | – | – | 0.53 | 0.83 | 0.83 | 0.83 | 0.83 | 0.83 |
| German Credit | Decision Tree | 0.48 | 0.51 | 0.59 | 0.72 | 0.74 | 0.74 | 0.74 | 0.74 |
| | Logistic Regression | 0.55 | 0.55 | 0.55 | 0.58 | 0.66 | 0.63 | 0.74 | 0.75 |
| | Naive Bayes | 0.50 | 0.49 | 0.54 | 0.57 | 0.65 | 0.66 | 0.63 | 0.65 |
| | Random Forest | 0.56 | 0.57 | 0.59 | 0.68 | 0.74 | 0.74 | 0.74 | 0.75 |
| | Deep Learning | – | – | 0.67 | 0.71 | 0.71 | 0.71 | 0.71 | 0.71 |
| COMPAS | Decision Tree | 0.52 | 0.54 | 0.60 | 0.64 | 0.64 | 0.64 | 0.64 | 0.64 |
| | Logistic Regression | 0.49 | 0.49 | 0.51 | 0.57 | 0.67 | 0.66 | 0.67 | 0.67 |
| | Naive Bayes | 0.50 | 0.50 | 0.50 | 0.58 | 0.65 | 0.66 | 0.66 | 0.66 |
| | Random Forest | 0.51 | 0.51 | 0.56 | 0.64 | 0.66 | 0.66 | 0.66 | 0.66 |
| | Deep Learning | – | – | 0.57 | 0.68 | 0.69 | 0.69 | 0.69 | 0.69 |

the cross-entropy loss function and is optimized using SGD with a learning rate of 0.01. The training process runs for 50 epochs. In DP-SGD, `max_grad_norm` is set by default to 1.

## *5.1 Impacts of Privacy on Model Accuracy*

Table 5.1 presents the accuracy values for all ML models (DT, LR, NB, RF) and the DL model under varying $\varepsilon$. The column labeled *No DP* in the table corresponds to the non-private model. It can be observed that as privacy gets stricter (i.e., $\varepsilon$ is decreased), models' accuracy values become lower. When $\varepsilon = 10$, most models' accuracy becomes equivalent to the non-private version. For several models, even $\varepsilon = 2$ and $\varepsilon = 1$ are sufficient to achieve accuracy values similar to their non-private counterparts. However, on almost all models, as $\varepsilon \leq 10^{-2}$, there is a noticeable discrepancy between the private models' accuracy and the accuracy of the non-private version.

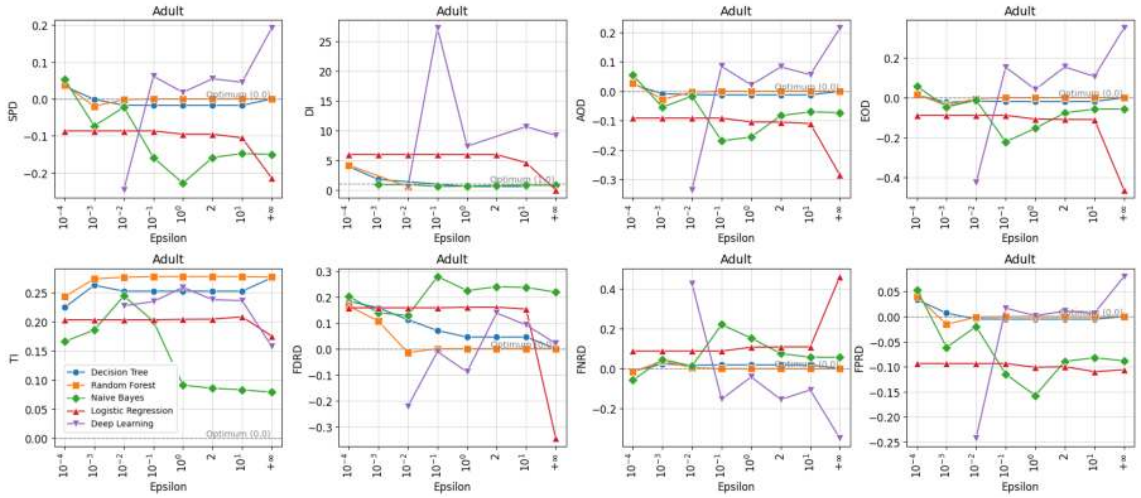


Figure 5.1: Fairness results across varying $\varepsilon$ values for the Adult dataset.

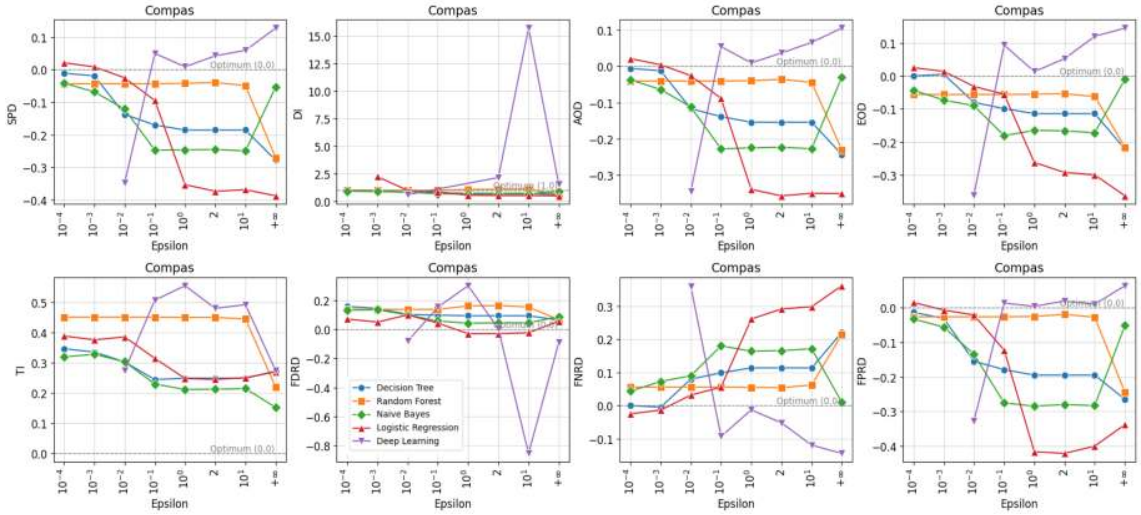


Figure 5.2: Fairness results across varying $\varepsilon$ values for the COMPAS dataset.

## *5.2 Impacts of Privacy on Model Fairness*

Next, we investigate how the strength of privacy protection (i.e., the value of $\varepsilon$) affects various privacy notions. In essence, we aim to answer the following question: How does differential privacy, with changing $\varepsilon$, affect fairness?

Figures 5.1, 5.2, and 5.3 display the results of all fairness metrics for the Adult, COMPAS, and German Credit datasets, respectively. In the figures, $+\infty$ corresponds to the case without DP (non-private training). Furthermore, Figure 5.4

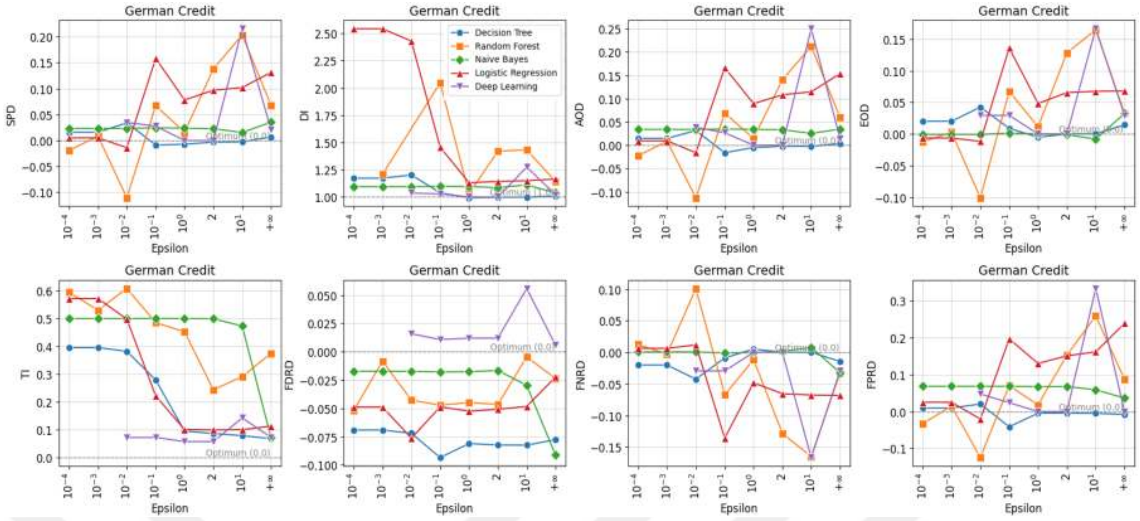


Figure 5.3: Fairness results across varying $\varepsilon$ values for the German Credit dataset.

illustrates the relationship between $\varepsilon$ and various fairness metrics, specifically for the DL model.

A central observation is how stricter privacy levels (i.e., smaller $\varepsilon$) often introduce more noise and thus affect group-level fairness. In particular, results for very small $\varepsilon$ values (e.g., $\varepsilon = 10^{-2}$), exhibit a larger gap in selection rates between the unprivileged (Female) and privileged (Male) groups because of the heightened noise. As $\varepsilon$ increases, we typically observe a trend toward the baseline (No DP) model, indicating that reduced noise allows the model to capture more faithful patterns in the data. Similarly, DI values can deviate significantly from 1.0 when $\varepsilon$ is extremely low, reflecting larger disparities in how privileged versus unprivileged groups are treated. However, as $\varepsilon$ grows, DI often draws closer to the non-private baseline, suggesting a rebalancing effect in the decision boundary once the noise level is reduced.

When examining metrics that consider conditional probabilities, such as Average Odds Difference (AOD) and Equal Opportunity Difference (EOD), we see a somewhat nuanced picture. For smaller $\varepsilon$, the model's ability to properly calibrate true positive and false positive rates across groups might degrade, leading to higher AOD values (i.e., larger differences in performance across groups). As $\varepsilon$ increases, these conditional rates generally become more similar to the No DP case, indicating an improvement in how consistently the model treats privileged versus unprivileged

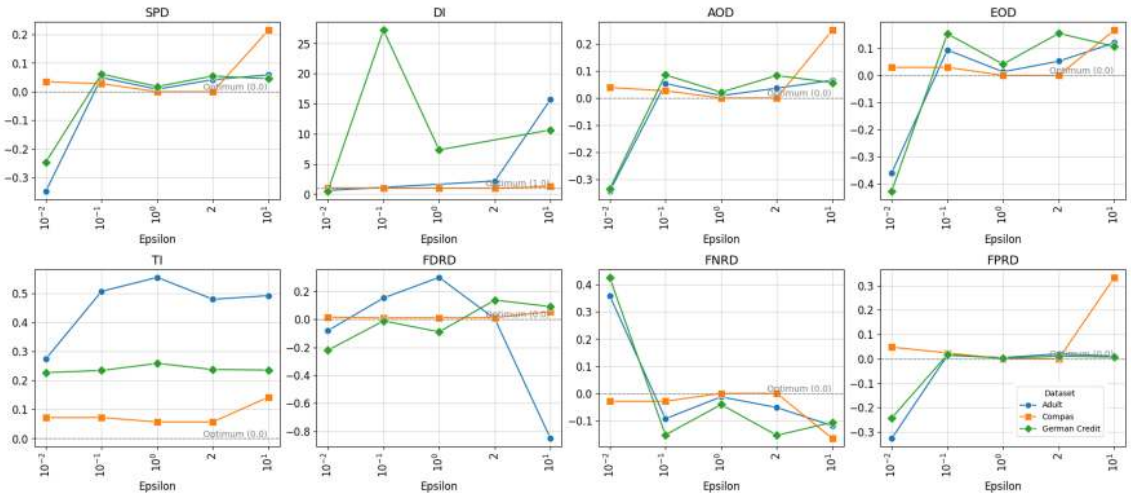


Figure 5.4: Fairness results of the DL model across varying $\varepsilon$ values for all datasets.

groups. For EOD, with more stringent privacy constraints, we often observe a noticeable gap in true positive rates for Male and Female groups. As DP noise diminishes with larger $\varepsilon$, EOD trends closer to zero, pointing toward more fairness between the groups.

For the Theil Index, in the presence of large noise (small $\varepsilon$), classification outcomes may inadvertently cluster in ways that inflate the Theil Index. As noise is reduced (larger $\varepsilon$), outcomes tend to distribute more in line with underlying data patterns, and the Theil Index typically decreases, often converging near the No DP baseline. This suggests that excessively strong privacy constraints can exacerbate outcome inequality.

The analysis of per-group error rates, such as FDRD, FNRD, and FPRD, provides further insight into how DP impacts the model's fairness. When $\varepsilon$ is smaller, the inability to accurately learn from the data can disproportionately increase false discoveries for one group over another. As $\varepsilon$ grows, FDRD often shrinks, highlighting a more uniform performance in false positive predictions. FNRD and FPRD capture whether one group is more likely to be misclassified (negatively or positively) than the other. They generally mirror the trends observed in AOD and EOD: small $\varepsilon$ can amplify misclassification differences, and larger $\varepsilon$ yields closer alignment between the unprivileged and privileged groups.

Overall, our results demonstrate a recurring theme: stricter privacy guarantees (smaller $\varepsilon$) come at the cost of potentially higher disparities across fairness metrics, whereas relaxed privacy (larger $\varepsilon$) tends to mitigate those disparities, often approaching the performance of the No DP baseline. This highlights the inherent trade-off between maintaining robust privacy and achieving fair, unbiased outcomes. In practical terms, while setting $\varepsilon$ to an extremely low value may protect sensitive information more rigorously, it risks amplifying biases due to excessive noise in gradient updates. Conversely, more permissive $\varepsilon$ values can reduce the noise and yield fairer outcomes, but reduce the strength of privacy guarantees.

Importantly, as shown in Table 5.1, high model accuracy does not necessarily imply fair treatment across groups. Even when utility remains relatively stable across different $\varepsilon$ values, fairness metrics can vary significantly. This indicates that practitioners must consider fairness trade-offs explicitly, even in the absence of a clear utility trade-off, reinforcing the need to jointly evaluate privacy, utility, and fairness in DP-based model development.

### *5.3 Impacts of Model Type on Fairness*

To analyze the impact of using different models, we conducted a comparative study using five algorithms: Decision Tree, Logistic Regression, Random Forest, Naive Bayes, and a Deep Learning model. All benchmarking datasets are employed. We report results with three key fairness metrics: Statistical Parity Difference (SPD), Average Odds Difference (AOD), and Theil Index (TI). These metrics were selected to provide complementary perspectives on fairness. SPD examines group-level disparities in selection rates, AOD captures disparities in conditional error rates, and TI measures overall inequality in outcomes.

Figure 5.5 displays the fairness metrics (SPD, AOD, TI) for each algorithm on the Adult dataset. RF and DT exhibit relatively stable *group-level* fairness for SPD and AOD. Their curves remain closer to zero across most $\varepsilon$ values, suggesting minimal disparities in SPD and AOD between privileged and unprivileged groups. However, this group-level advantage contrasts with their higher Theil Index values.

A high TI indicates greater *individual-level* inequality in the distribution of predicted outcomes, implying that consistent group-level fairness may mask finer-grained inequities at the individual level.

On the other hand, NB shows more fluctuation in SPD and AOD, at times dipping noticeably below zero (indicating potential disadvantage for the unprivileged group) before moving closer to zero as privacy constraints ease. Interestingly, NB achieves some of the lowest TI values at moderate and larger $\varepsilon$, suggesting that once privacy-related noise is reduced, it distributes outcomes relatively evenly across individuals.

LR maintains moderately negative SPD and AOD values across most $\varepsilon$ settings, indicating a consistent but not zero-centered disparity for the unprivileged group. Nevertheless, this model appears more stable than DL in handling privacy noise. The Theil Index for LR falls roughly in the mid-range, suggesting neither particularly high nor particularly low outcome inequality at the individual level.

DL displays the most pronounced volatility in SPD and AOD, particularly under tight privacy (smaller $\varepsilon$). This oscillation indicates that while the model may capture complex data relationships, it is especially sensitive to gradient noise introduced by DP. Its TI curve also fluctuates more than those of simpler models, underscoring a trade-off between higher representational capacity and robustness to privacy noise.

Further analysis using the COMPAS dataset, illustrated in Figure 5.6, reinforces an important caution: models with poor predictive utility (i.e., accuracy close to random guessing) may misleadingly appear fair. In the region highlighted in yellow, multiple models under strict privacy constraints exhibit SPD and AOD values close to zero. However, Table 5.1 shows that these models perform with accuracies around 50%, indicating that their seemingly fair outcomes may stem from random predictions rather than meaningful learning. This emphasizes that fairness metrics must be interpreted in conjunction with model utility to avoid misrepresenting the fairness-performance trade-off.

In summary, Figure 5.5 highlights a nuanced relationship between model complexity, DP noise, and different notions of fairness. RF, for instance, shows favorable group-level fairness but fares worse in individual-level fairness, while NB manages

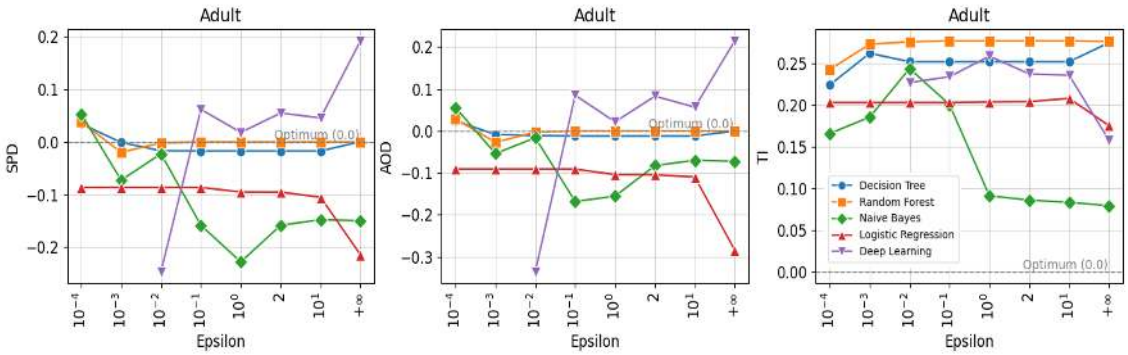


Figure 5.5: Fairness metrics (SPD, AOD, TI) across different model types on the Adult dataset.

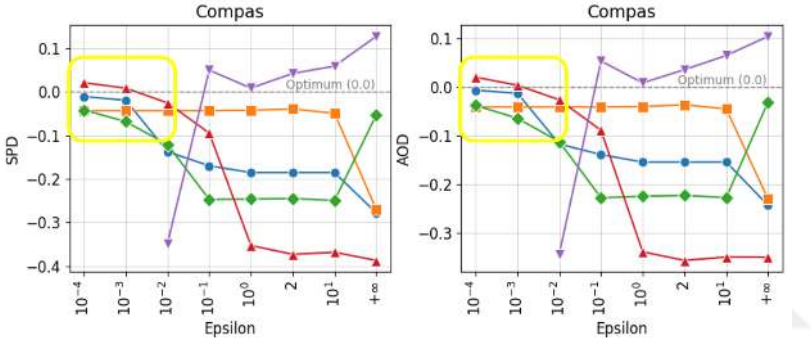


Figure 5.6: Close-up of SPD and AOD metrics on COMPAS dataset under tight privacy settings. The highlighted region shows misleading fairness under low-accuracy conditions.

to reduce individual-level disparities once privacy constraints loosen. LR remains relatively stable but with a persistent bias across SPD and AOD. DL, although powerful, is least stable under stringent privacy budgets. These observations emphasize the importance of considering multiple fairness dimensions and the inherent trade-offs in choosing both privacy parameters and model types.

## *5.4 Impacts of Varying Datasets*

To investigate the impacts of different datasets on fairness, we utilized three datasets: Adult, German Credit, and COMPAS. Similar to previous section, we selected three key fairness metrics: Statistical Parity Difference (SPD), Average Odds Difference

(AOD), and Theil Index (TI). Figure 5.7 compares fairness outcomes across the three different datasets: German Credit (using RF), COMPAS (using NB), and Adult (using LR), evaluated under various privacy budgets ($\varepsilon$). High-performing models were specifically chosen for each dataset to minimize model-related biases, thereby emphasizing the inherent impact of dataset characteristics on fairness.

The differences observed in SPD highlight how intrinsic dataset properties affect fairness. The German Credit dataset maintains SPD values consistently close to the optimal zero level, suggesting inherently balanced group selection rates. In contrast, the Adult dataset reveals a consistent but moderate negative disparity, indicating a persistent bias against the unprivileged group irrespective of privacy levels. The COMPAS dataset exhibits the most striking behavior: its initially neutral fairness rapidly deteriorates as privacy constraints are relaxed. This indicates that inherent biases embedded within the COMPAS dataset emerge more prominently when DP noise is minimized, reflecting substantial intrinsic bias related to the dataset itself rather than the chosen model.

For AOD, similar dataset-dependent patterns emerge. The German Credit dataset continues to display minimal disparities in conditional error rates across protected groups, underscoring inherent stability with respect to fairness. Adult dataset maintains moderate disparity consistently across privacy levels, suggesting that underlying biases related to demographic features remain stable. COMPAS, however, shows a dramatic increase in unfairness under lower-noise conditions, highlighting again the strong intrinsic biases in the conditional error rates of the dataset itself.

The Theil Index (TI), measuring individual-level fairness, further emphasizes these dataset-driven disparities. At strict privacy levels (small $\varepsilon$), all datasets exhibit relatively high TI values, indicative of widespread inequality at the individual prediction level due to noise-induced randomness. However, as privacy constraints relax, differences driven by inherent data characteristics become clear. The German Credit dataset rapidly improves individual fairness with decreasing noise, suggesting that its structure inherently allows fairer individual-level predictions. The Adult dataset also shows improvement, though more moderately, indicating stable but less optimal individual-level fairness. Conversely, the COMPAS dataset shows

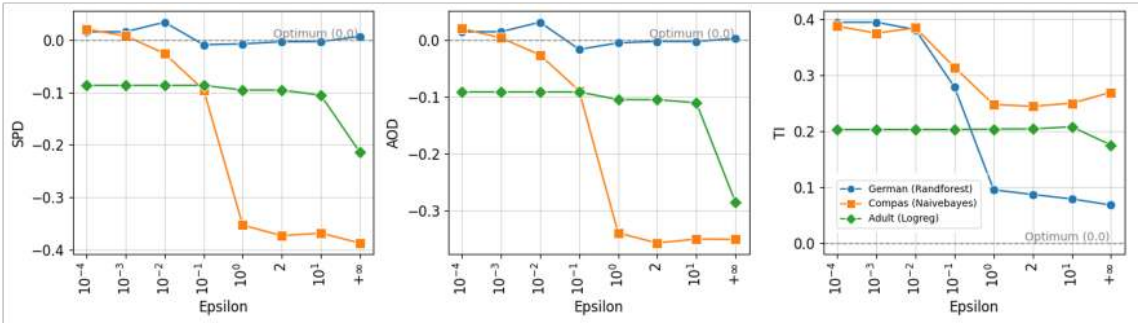


Figure 5.7: Fairness metrics (SPD, AOD, TI) across different datasets (Adult, German Credit, COMPAS) under varying $\varepsilon$ values.

limited improvement in TI, stabilizing at a higher inequality level, emphasizing that individual-level fairness is significantly constrained by the dataset's intrinsic structural bias.

These findings emphasize the need for dataset-specific fairness evaluations. For datasets with significant disparities, such as COMPAS, additional preprocessing or fairness-enhancing techniques may be required to address inherent biases. Moreover, the stability of fairness metrics in the Adult dataset suggests that it may be better suited for applications prioritizing fairness under privacy constraints. Overall, these results confirm that fairness outcomes under DP are profoundly influenced by the inherent characteristics and biases of the underlying data.

## *5.5 Impacts of Hyperparameter Tuning on Fairness*

In this experiment, we investigate how tuning a key hyperparameter, the maximum tree depth, impacts fairness in a Decision Tree (DT) model. The privacy budget $\varepsilon = 1$ is kept static for this experiment. We vary the tree depth from 2 up to 21, keeping all other parameters (e.g., splitting criterion, minimum samples per split) fixed. At each chosen tree depth, we train the DT model on the same training set and evaluate on the same test set in order to isolate the effect of changing depth.

Figure 5.8 illustrates how fairness metrics and accuracy evolve as the maximum tree depth increases. As the tree grows deeper, it gains the capacity to learn more complex decision boundaries. While this often results in improved classification

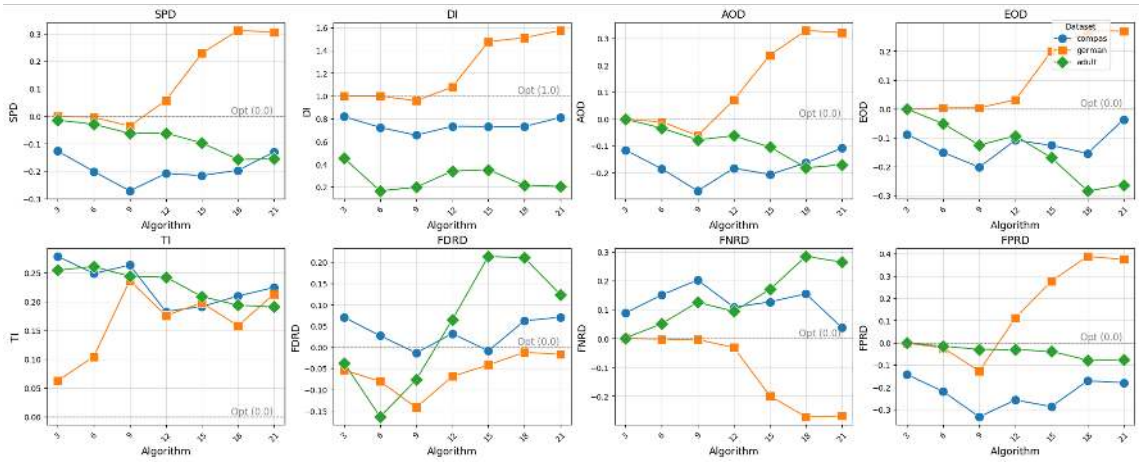


Figure 5.8: Fairness metrics (y-axis) versus tree depth (x-axis) on the Adult, German Credit and COMPAS datasets. Higher tree depth increases accuracy while reducing fairness across most metrics.

performance, Table 5.2 shows that increasing tree depth does not consistently lead to significant accuracy gains across all datasets. In fact, for datasets like German Credit, accuracy fluctuates and in some cases even decreases at deeper levels.

Despite this limited or inconsistent utility gain, fairness metrics, such as SPD, AOD, and EOD, consistently exhibit a *decline* in fairness with increasing depth. This suggests that model complexity can adversely affect fairness even when accuracy remains unchanged. The underlying reason is that deeper trees produce more specialized leaf nodes, enabling highly specific rules that may overfit to dominant patterns in the training data. These decision rules can inadvertently favor one demographic group over another, leading to disparities in prediction outcomes, particularly in terms of error rates and selection rates between privileged and unprivileged groups. Overall, these findings support the intuition that tuning hyperparameters for higher accuracy can negatively affect group fairness. Depending on the application domain, practitioners may need to constrain model complexity if fairness is a priority.

Table 5.2: DT accuracy under varying depths for $\varepsilon = 1$

| **Dataset** | **Default** | **Depth 3** | **Depth 6** | **Depth 9** | **Depth 12** | **Depth 15** | **Depth 18** | **Depth 21** |
|---|---|---|---|---|---|---|---|---|
| Adult | 0.76 | 0.76 | 0.76 | 0.77 | 0.77 | 0.78 | 0.78 | 0.79 |
| German | 0.68 | 0.76 | 0.68 | 0.62 | 0.64 | 0.64 | 0.66 | 0.64 |
| COMPAS | 0.63 | 0.58 | 0.62 | 0.63 | 0.68 | 0.65 | 0.66 | 0.65 |

## *5.6 Relationships Between Different Fairness Metrics*

To explore how various fairness metrics interrelate, we devised an experimental procedure comprising several steps. First, for each dataset, we perform a random split into training and test subsets using a 60%-40% split ratio. Next, we calculate the relevant fairness metrics based on the derived training sets and ML models trained on those sets. We repeat this entire process 20 times, where each repetition uses a distinct random partition, and we record the fairness metric outcomes for each iteration. Ultimately, we determine the pairwise correlations among these recorded metric values.

To quantify correlation, we employ three distinct correlation measures: Pearson correlation, Spearman correlation, and Kendall tau. The purpose of using more than one correlation definition is to ensure that our conclusions do not hinge on the choice of correlation measure. In other words, if all three correlation measures support the same patterns, we can be more confident in our findings.

Figure 5.9 presents correlation outcomes for three fairness measures, namely Disparate Impact (DI), Consistency (CO), and Statistical Parity Difference (SPD). Based on Figure 5.9, we notice a strong positive link between SPD and DI. Intuitively, both rely on comparing $\Pr[Y = 1 \mid D = 0]$ and $\Pr[Y = 1 \mid D = 1]$, although one focuses on their ratio and the other on their difference. Meanwhile, Consistency (CO) does not appear to correlate substantially with DI, or SPD, suggesting it captures a different fairness perspective. This observation aligns with expectations because CO is an *individual-level* fairness metric, whereas DI and SPD are *group-level* metrics.

Figure 5.10 shows correlation findings for six other fairness metrics: Average

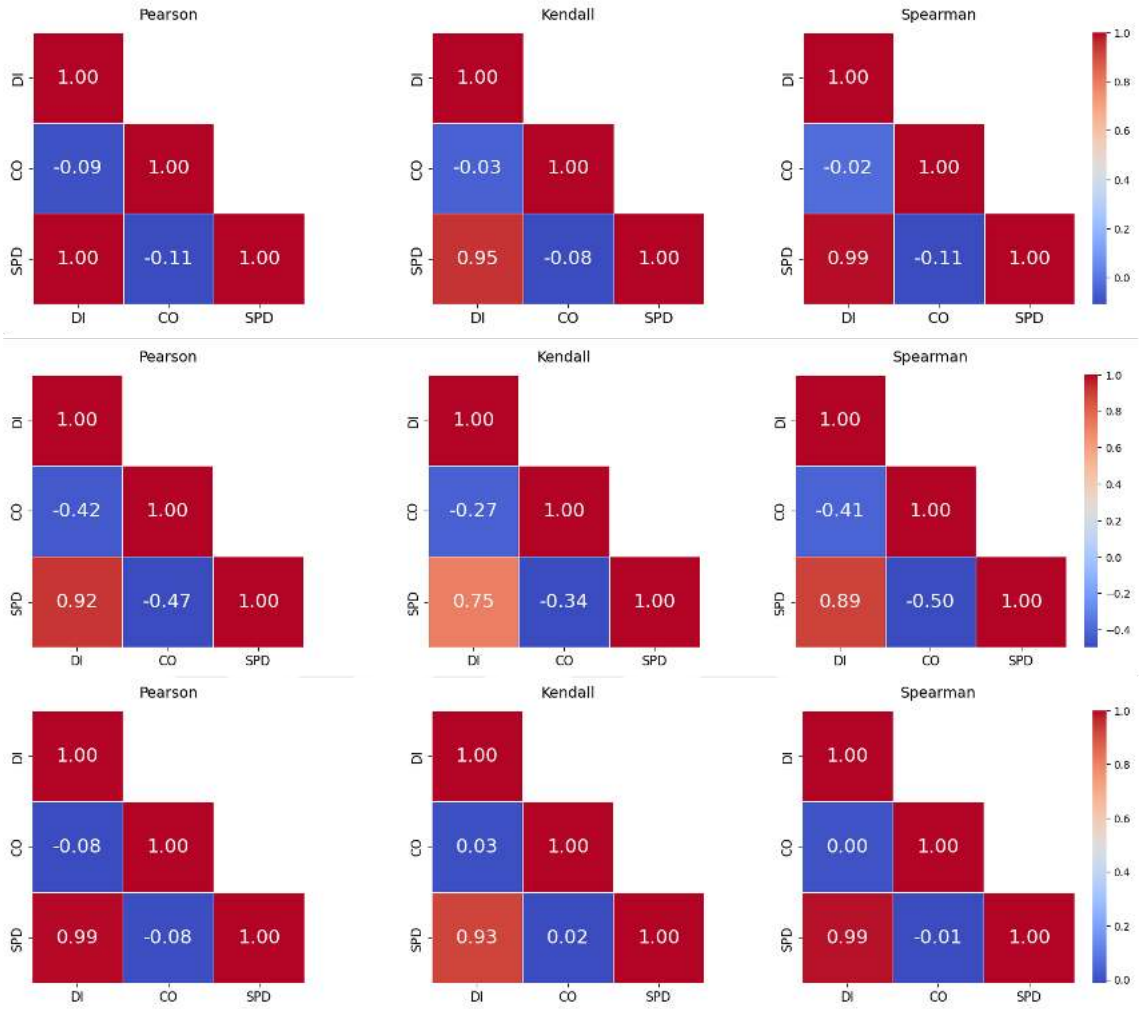


Figure 5.9: Correlations of three fairness metrics (DI, CO, and SPD), according to three correlation definitions: Pearson, Kendall tau, Spearman. German Credit dataset on the first row, Adult dataset on the second row, COMPAS on the third row.

Odds Difference (AOD), Equal Opportunity Difference (EOD), Theil Index (TI), False Discovery Rate Difference (FDRD), False Negative Rate Difference (FNRD), and False Positive Rate Difference (FPRD). In this figure, we used LR as the model. First, we discern a strong positive correlation between EOD and AOD, as well as between FPRD and AOD. This makes sense because EOD measures a difference in TPRs, FPRD concerns a difference in FPRs, and AOD tracks differences in both TPRs and FPRs. Consequently, any adjustment in the TPR gap influences both EOD and AOD, while an adjustment in the FPR gap has ramifications for

both FPRD and AOD. Second, EOD and FPRD also display a positive correlation, although not to the same degree as EOD–AOD or FPRD–AOD. Since EOD and FPRD do not inherently quantify the same type of rate difference, this result is intriguing. Our hypothesis is that increasing a model's propensity to predict positive leads to a simultaneous increase in TPR and FPR, thus affecting both EOD and FPRD; decreasing that propensity similarly lowers TPR and FPR.

Next, we observe robust negative correlations between FNRD and AOD, as well as between FNRD and EOD. Since EOD and AOD hinge on TPR, while FNRD evaluates FNR, this strong inverse relationship seems natural. By definition, TPR equals $\frac{TP}{TP+FN}$, while FNR equals $\frac{FN}{TP+FN}$. When a true label $Y$ is 1, the classifier's prediction $\widehat{Y}$ can be 1 (yielding a true positive) or 0 (yielding a false negative). TPR and FNR are thus bound to be inversely correlated.

As for the remaining metric combinations, we do not see compelling evidence of either positive or negative correlation. Hence, some fairness metrics appear independent or uncorrelated with one another.

Finally, we look at how consistent our results are across the three correlation measures (Pearson, Spearman, and Kendall). We notice that the outcomes for Pearson and Spearman frequently resemble each other more closely than those for Kendall. While the discrepancies are especially notable in the Adult dataset, the principal trends remain unchanged across German and COMPAS as well. Overall, regardless of the dataset or correlation measure, we consistently identify certain pairs of fairness metrics with robust correlation and others that are essentially unrelated. This highlights two key insights: (i) by their nature, some fairness metrics exhibit strong correlation while others do not, and (ii) examining correlations across more than one definition promotes greater reliability and depth in interpreting the results.

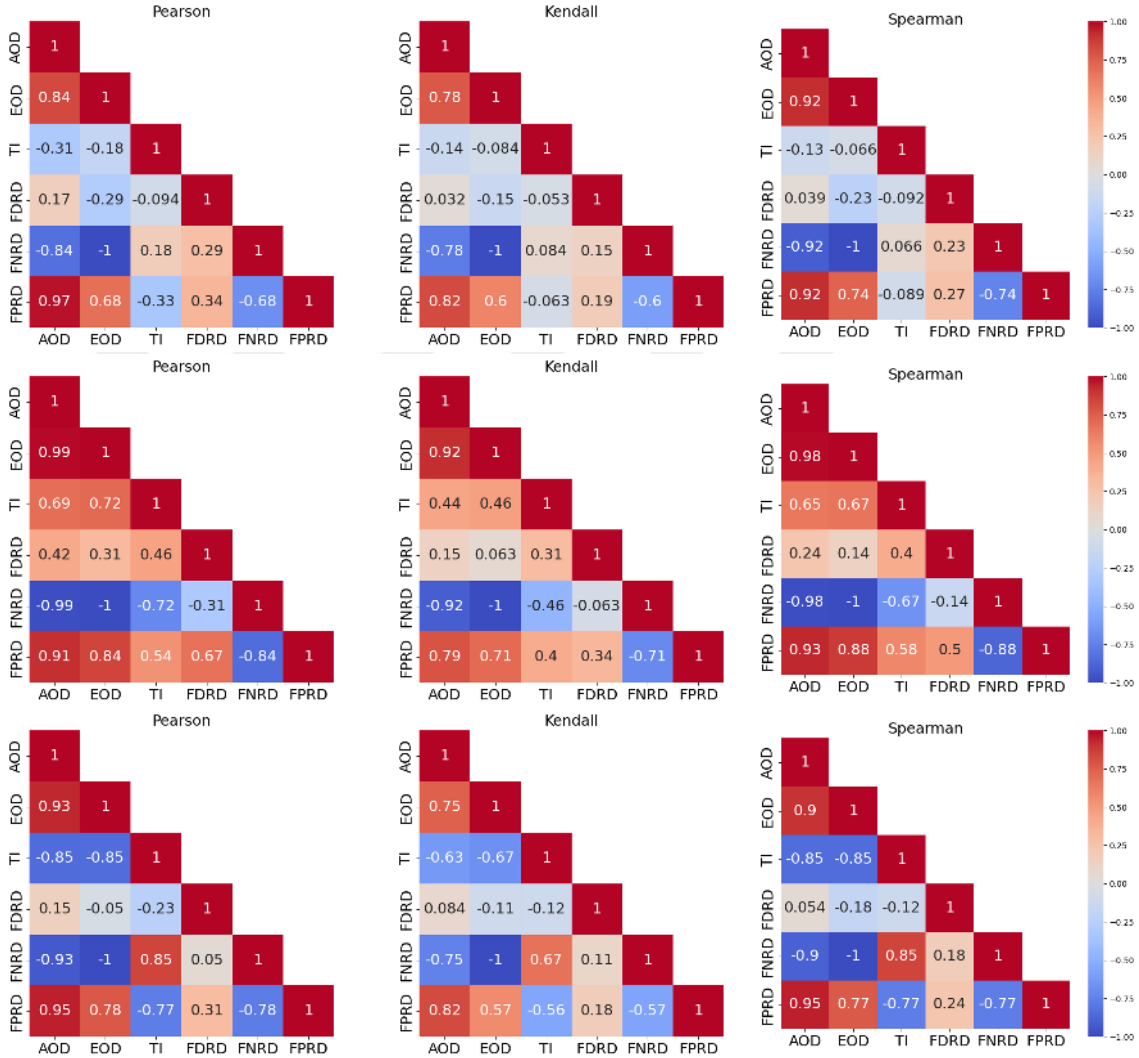


Figure 5.10: Correlations of six fairness metrics (AOD, EOD, TI, FDRD, FNRD, FPRD) across three correlation definitions: Pearson, Kendall tau, and Spearman. LR model is used. The first row corresponds to the German dataset, the second row to the Adult dataset, and the third row to the COMPAS dataset.

# Chapter 6

# CONCLUSION

In this thesis, we set out to explore the impact of DP on algorithmic fairness and to provide the community with a means to quantify and understand this impact. The core problem addressed by the thesis was the lack of clarity on how DP model training methods influence fairness outcomes, considering the conflicting opinions and findings present in the literature. We responded to this problem by designing and implementing **FairnessDPLab**, a comprehensive evaluation platform to empirically analyze models trained with DP across multiple datasets, models, and fairness criteria. Through experiments utilizing FairnessDPLab, we gained valuable insights into the privacy-fairness trade-off. We observed how tightening privacy (lowering the privacy budget $\varepsilon$) can affect model performance and various fairness metrics, and how those effects differ depending on context. This contribution fills a vital gap in the literature by offering empirical evidence and a practical toolkit for researchers and practitioners concerned with building AI systems that are both private and fair.

## *6.1 Key Findings*

The key findings of this thesis can be summarized as follows:

- **A single fairness metric is not sufficient:** Fairness is a multi-faceted concept, and our experiments confirmed that relying on any one metric can be misleading. We found cases where a model might appear fair by one measure (for instance, having balanced positive prediction rates between groups) while exhibiting bias according to another measure (such as disparate error rates). This underscores the need to evaluate multiple fairness metrics in parallel, especially under differentially private training.

- **DP influences fairness in varying ways:** The effect of DP on fairness is not uniform across all scenarios. Different privacy parameter settings (e.g., varying $\varepsilon$ values) can alter fairness outcomes in ways that depend on the underlying dataset and model. For example, according to our results, some datasets (like Adult) maintained relatively stable fairness results even under strict privacy, whereas others (like COMPAS) showed significant disparities as the privacy budget tightened. Likewise, simpler models (e.g., LR) and more complex models (DL) reacted differently to the introduction of noise. This variability means there is no one-size-fits-all answer – the fairness impact of DP must be assessed on a case-by-case basis. Practitioners should be aware that increasing privacy protection can sometimes amplify or sometimes mitigate bias, and thus, careful tuning and evaluation are necessary.

- **FairnessDPLab provides a structured evaluation framework:** A major contribution of this thesis is the FairnessDPLab platform itself, which offers a systematic way to evaluate trade-offs between privacy, accuracy, and fairness. By using this framework, one can efficiently run controlled experiments to see how adjusting the privacy budget or switching the model type affects different fairness metrics. This empowers users to make informed decisions – for instance, choosing an acceptable privacy level that does not unduly sacrifice fairness, or selecting a model that best balances both. In providing this tool, the thesis contributes a practical benchmarking methodology that others can build upon for future research and development of fair and privacy-preserving AI systems.

## *6.2 Future Research Directions*

Looking ahead, our work opens up several promising avenues for future research. First, one important direction is to move from evaluating trade-offs to actively optimizing for both fairness and privacy. Future research could develop new training algorithms or optimization techniques that incorporate fairness constraints or objectives directly into the differentially private learning process. Such multi-objective

approaches (balancing accuracy, privacy, and fairness) could enable models to maintain equitable outcomes while satisfying DP. Second, another key direction is the practical deployment of fairness-aware DP in real-world applications. This involves taking the insights and tools (such as FairnessDPLab) beyond a research setting and integrating them into industry and government AI pipelines. Third, our study focused on a set of well-known group fairness metrics and classification tasks; future work can broaden this scope. This includes exploring new fairness metrics or definitions tailored to DP. Moreover, examining intersectional fairness (considering multiple protected attributes at once) under DP would be a valuable extension.

# BIBLIOGRAPHY

[Abadi et al., 2016] Abadi, M., Chu, A., Goodfellow, I., McMahan, H. B., Mironov, I., Talwar, K., and Zhang, L. (2016). Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*.

[Bagdasaryan et al., 2019] Bagdasaryan, E., Poursaeed, O., and Shmatikov, V. (2019). *Differential privacy has disparate impact on model accuracy*. Curran Associates Inc., Red Hook, NY, USA.

[Bao et al., 2021] Bao, M., Zhou, A., Zottola, A. S., Brubach, B., Desmarais, S., Horowitz, S. A., Lum, K., and Venkatasubramanian, S. (2021). It's compaslicated: The messy relationship between rai datasets and algorithmic fairness benchmarks. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 1)*.

[Bellamy et al., 2019] Bellamy, R. K., Dey, K., Hind, M., Hoffman, S. C., Houde, S., Kannan, K., Lohia, P., Martino, J., Mehta, S., Mojsilovic, A., et al. (2019). Ai fairness 360: An extensible toolkit for detecting and mitigating algorithmic bias. *IBM Journal of Research and Development*, 63(4/5):4:1–4:15.

[Breiman, 2001] Breiman, L. (2001). Random forests. *Machine Learning*, 45(1):5–32.

[Chan et al., 1979] Chan, T. F., Golub, G. H., and LeVeque, R. J. (1979). Updating formulae and a pairwise algorithm for computing sample variances. Technical report, Stanford University, Stanford, CA, USA.

[Chaudhuri et al., 2011] Chaudhuri, K., Monteleoni, C., and Sarwate, A. D. (2011).

Differentially private empirical risk minimization. *Journal of Machine Learning Research*, 12:1069–1109.

[Chouldechova, 2017] Chouldechova, A. (2017). Fair prediction with disparate impact: A study of bias in recidivism prediction instruments. *Big Data*, 5(2):153–163.

[Cummings et al., 2019] Cummings, R., Gupta, V., Kimpara, D., and Morgenstern, J. (2019). On the compatibility of privacy and fairness. In *Adjunct Publication of the 27th Conference on User Modeling, Adaptation and Personalization*, UMAP'19 Adjunct, page 309–315, New York, NY, USA. Association for Computing Machinery.

[Dwork et al., 2012] Dwork, C., Hardt, M., Pitassi, T., Reingold, O., and Zemel, R. (2012). Fairness through awareness. In *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference*, pages 214–226.

[Dwork et al., 2014] Dwork, C., Roth, A., et al. (2014). The algorithmic foundations of differential privacy. *Foundations and Trends in Theoretical Computer Science*, 9(3–4):211–407.

[Fioretto et al., 2022] Fioretto, F., Tran, C., Van Hentenryck, P., and Zhu, K. (2022). Differential privacy and fairness in decisions and learning tasks: A survey. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence*, IJCAI-2022, page 5470–5477. International Joint Conferences on Artificial Intelligence Organization.

[Fletcher and Islam, 2017] Fletcher, S. and Islam, M. Z. (2017). Differentially private random decision forests using smooth sensitivity. *Expert Systems with Applications*, 78:16–31.

[Gong et al., 2020] Gong, M., Xie, Y., Pan, K., Feng, K., and Qin, A. (2020). A survey on differentially private machine learning [review article]. *IEEE Computational Intelligence Magazine*, 15(2):49–64.

[Hardt et al., 2016] Hardt, M., Price, E., and Srebro, N. (2016). Equality of opportunity in supervised learning. In *Advances in Neural Information Processing Systems*.

[Hofmann, 1994] Hofmann, H. (1994). Statlog (german credit data) data set. `https://archive.ics.uci.edu/ml/datasets/statlog+(german+credit+data)`. UCI Machine Learning Repository.

[Holohan et al., 2019] Holohan, N., Braghin, S., Mac Aonghusa, P., and Levacher, K. (2019). Diffprivlib: the IBM differential privacy library. *ArXiv e-prints*, 1907.02444 [cs.CR].

[Jagielski et al., 2019] Jagielski, M., Kearns, M., Mao, J., Oprea, A., Roth, A., Sharifi-Malvajerdi, S., and Ullman, J. (2019). Differentially private fair learning.

[Ji et al., 2014] Ji, S. et al. (2014). Differentially private naive bayes classification. In *IEEE International Conference on Data Mining*.

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

[Kohavi and Becker, 1996] Kohavi, R. and Becker, B. (1996). Adult data set. `https://archive.ics.uci.edu/ml/datasets/adult`. UCI Machine Learning Repository.

[Kusner et al., 2017] Kusner, M. J., Loftus, J., Russell, C., and Silva, R. (2017). Counterfactual fairness. *Advances in Neural Information Processing Systems*, 30.

[Mehrabi et al., 2019] Mehrabi, N., Morstatter, F., Saxena, N., Lerman, K., and Galstyan, A. (2019). A survey on bias and fairness in machine learning. *arXiv preprint arXiv:1908.09635*.

[Paszke et al., 2019] Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al. (2019). Pytorch: An imperative style, high-performance deep learning library. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché Buc, F., Fox, E., and Garnett, R., editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc.

[Pujol et al., 2020] Pujol, D., McKenna, R., Kuppam, S., Hay, M., Machanavajjhala, A., and Miklau, G. (2020). Fair decision making using privacy-protected data. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, FAT* '20, page 189–199, New York, NY, USA. Association for Computing Machinery.

[Vaidya et al., 2013] Vaidya, J., Shafiq, B., Basu, A., and Hong, Y. (2013). Differentially private naive bayes classification. In *Proceedings of the 2013 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT) - Volume 01*, WI-IAT '13, page 571–576, USA. IEEE Computer Society.

[Verma and Rubin, 2018] Verma, S. and 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.

[Yousefpour et al., 2021] Yousefpour, A., Shilov, I., Sablayrolles, A., Testuggine, D., Prasad, K., Malek, M., Nguyen, J., Ghosh, S., Bharadwaj, A., Zhao, J., et al. (2021). Opacus: User-friendly differential privacy library in pytorch. In *Thirty-fifth Conference on Neural Information Processing Systems (NeurIPS 2021), Sydney, Australia*.

[Yu et al., 2011] Yu, H.-F., Huang, F.-L., and Lin, C.-J. (2011). Dual coordinate descent methods for logistic regression and maximum entropy models. *Machine Learning*, 85(1-2):41–75.

[Zemel et al., 2013] Zemel, R., Wu, Y., Swersky, K., Pitassi, T., and Dwork, C. (2013). Learning fair representations. In *International Conference on Machine Learning*, pages 325–333. PMLR.