UNIVERSITY OF TURKU
Turku School of Economics

TILBURG UNIVERSITY

Designing fair and compliant AI: Evaluating bias mitigation methods under data minimisation constraints

A case study on the COMPAS dataset

Information Systems Science

Master's thesis

Author:

Johannes Bekkers

Supervisors:

Dr Emiel Caron

Dr Farhan Ahmad

15.08.2025

Tilburg

The originality of this thesis has been checked in accordance with the University of Turku quality assurance system using the Turnitin Originality Check service.

Master's thesis

Subject: Information Systems Science
Author: Johannes Bekkers
ANR: 901507
SNR: 2047005
Study right number: 2406937
Title: Designing fair and compliant AI: Evaluating bias mitigation methods under data minimisation constraints
Supervisors: Dr Emiel Caron & Dr Farhan Ahmad
Number of pages: 104 pages + appendices 42 pages
Date: 15.08.2025

Abstract: The increasing integration of machine learning into high-stakes decision-making has intensified concerns about fairness, accountability, and compliance with data protection regulations. A central tension arises between the General Data Protection Regulation's data minimisation principle, which restricts access to sensitive attributes such as race, and the practical need for such data in bias mitigation and fairness auditing. This study examines whether algorithmic fairness can be achieved when bias mitigation techniques are applied under data minimisation constraints. Using the publicly available COMPAS recidivism dataset as a case study, two interpretable and widely used classifiers, Logistic Regression and Random Forest, were trained on both a dataset that included race and a data-minimised version that excluded it from model training. Models were evaluated using a variety of well-established performance and fairness metrics. In the full-data setting, bias mitigation methods requiring sensitive attributes improved fairness while maintaining performance. In the data-minimised setting, baseline models exhibited higher fairness but slightly reduced predictive performance. Mitigation options were restricted to a narrow range of post-hoc interventions, with effectiveness concentrated in a single method. This setting nonetheless achieved stronger fairness improvements than the full-data case, despite the limited toolkit. The findings highlight a trade-off in which data minimisation can provide a fairer starting point but narrows the range and reliability of available bias mitigation techniques, creating dependence on specific interventions. These insights are relevant to policymakers and practitioners seeking to balance privacy compliance with robust algorithmic fairness.

Key words: Machine Learning, Algorithmic Fairness, GDPR, Bias Mitigation, Data Minimisation, COMPAS, Ethical AI.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

AI	Artificial Intelligence
COMPAS	Correctional Offender Management Profiling for Alternative Sanctions
CRISP-DM	Cross-Industry Standard Process for Data Mining
DSS	Decision Support Systems
EGR	Exponentiated Gradient Reduction
EOD	Equalised Odds Difference
EOD-NAA	Equalised Odds Difference excluding Native American and Asian
FPR	False Positive Rate
GDPR	General Data Protection Regulation
KDE	Kernel Density Estimates
KNN	K-Nearest Neighbours
LR	Logistic Regression
ML	Machine Learning
RF	Random Forest
ROC AUC	Receiving Operating Characteristic Area Under the Curve
SR	Selection Rate
SVMs	Support Vector Machines
TPR	True Positive Rate

1 Introduction

1.1 Background

In recent years, the growing adoption of Artificial Intelligence (AI) across both public and private sectors has raised increasing concern about fairness, accountability and data protection. AI systems, often built using Machine Learning (ML), are being used to assist or automate decisions in high-stakes domains such as hiring, lending, healthcare and criminal justice. These systems promise enhanced efficiency and reduced operational costs (Davenport & Ronanki, 2018). As a result, global AI adoption has accelerated, driven by advances in algorithms, the growing availability of data, and increased computational power. According to McKinsey (2022), the share of organisations using AI in at least one business function rose from 20% in 2017 to 50% in 2022, with parallel increases in budget and capability investment. However, despite this growth, the mitigation of AI-related risks has not kept pace.

A particular concern in this context is bias in ML, which refers to any systematic influence arising from algorithmic design, modelling assumptions, data characteristics, or human decisions that causes models to produce unfair, inaccurate, or non-generalisable outcomes (Holmberg et al., 2020). While bias is often rooted in historical data, it can also be introduced through modelling choices such as feature selection, objective functions, or optimisation techniques. As a result, fairness concerns may emerge even when the data appears neutral, making it essential to evaluate the entire development pipeline of AI systems. Fairness in this context refers to the absence of prejudice or favouritism toward individuals or groups based on inherent or acquired characteristics (Mehrabi et al., 2019).

The risk posed by algorithmic bias becomes especially problematic when decisions affect diverse populations. Historical studies show that human judgement has always been prone to bias (Bertrand & Mullainathan, 2004; Martin, 2007), but biased AI can amplify these effects at scale (Gupta & Krishnan, 2020). For instance, biased facial recognition software and hiring tools have already caused reputational damage and legal consequences for corporations (Singer, 2018; EEOC, 2022; Reuters, 2023). As a result, there is increasing pressure on organisations to ensure fairness not only for ethical reasons but also to comply with legal and reputational expectations.

In parallel with these developments, data protection regulations such as the European General Data Protection Regulation (GDPR) have introduced additional constraints on how data can be used in AI systems. One key principle, data minimisation, requires organisations to limit data collection and processing to what is strictly necessary for the intended purpose. However, in fairness auditing and

mitigation, access to sensitive attributes such as race, gender or age is often crucial. This presents a regulatory dilemma: enforcing fairness often requires using data that privacy law aims to restrict.

This thesis explores whether algorithmic fairness can still be effectively achieved when ML systems are subject to data minimisation constraints. Using the well-known COMPAS dataset as a case study, the research focuses on supervised ML classifiers, which are algorithms that learn from labelled data to predict discrete outcomes. In this study, Logistic Regression (LR) and Random Forest (RF) are selected as proxy models for replicating complex or black-box decision systems. Logistic Regression offers high interpretability (Caraciolo, 2011), while Random Forest, an ensemble method, provides robust performance (Breiman, 2011). Both models are widely used and accessible, making them suitable for real-world deployment in both public and corporate contexts.

By comparing bias mitigation outcomes across these models both before and after removing sensitive attributes, this study seeks to assess the trade-offs between fairness, performance, and regulatory compliance. The findings aim to inform not only academic debates on algorithmic fairness, but also provide practical guidance for businesses seeking to deploy responsible AI within legal constraints.

1.2 Problem statement

As outlined above, organisations increasingly rely on algorithmic systems to support or automate high-stakes decision-making. One prominent example is the use of ML models in criminal justice to predict recidivism risk, such as in the COMPAS system widely used in the United States. However, these systems have raised serious concerns about fairness and discrimination, especially when outcomes disproportionately affect certain demographic groups.

While bias mitigation techniques have been developed to promote algorithmic fairness, data protection frameworks such as the GDPR impose constraints like data minimisation, which may limit access to the sensitive attributes often needed for such mitigation. This raises a practical and regulatory dilemma between fairness and compliance.

Therefore, this study seeks to examine whether it is possible to apply bias mitigation techniques effectively in contexts where sensitive data must be removed. The goal is to understand how both fairness and predictive performance are affected when applying bias mitigation techniques before and after data minimisation, while also considering the transparency and interpretability of the models used. This study uses interpretable proxy models: Logistic Regression and Random Forest, trained on the COMPAS dataset.

The COMPAS (Correctional Offender Management Profiling for Alternative Sanctions) dataset contains real-world data from the U.S. criminal justice system. It was originally published by ProPublica as part of an investigation into potential racial bias in algorithmic risk assessments (Angwin et al., 2016). The dataset includes demographic variables (such as race, age, and gender), criminal history, and the outcome of whether a person reoffended within two years. These characteristics make it a widely used benchmark in algorithmic fairness research and well-suited for studying the effects of bias mitigation and data minimisation in practice.

Hence, the problem statement for this study is:

Evaluate whether algorithmic fairness can still be achieved when bias mitigation techniques are applied to supervised machine learning classifiers that operate under data minimisation constraints.

1.3 Research question

The research question to address this problem statement is:

How effective are different bias mitigation methods across multiple supervised machine learning classifiers when access to sensitive data is restricted due to data minimisation?

To answer the main research question, the sub-questions are formulated below. Together, they help evaluate the effects of bias mitigation under data minimisation across different supervised ML classifiers.

Theoretical:

SQ1: How are bias and fairness defined in the context of supervised machine learning classifiers, and what are the most commonly used fairness metrics?

Answered through a literature review on algorithmic fairness and fairness metrics relevant to group-level evaluation (e.g. demographic parity, equalised odds).

SQ2: Which bias mitigation techniques are applicable to supervised machine learning classifiers, and how are they typically categorised (pre-processing, in-processing, post-processing)?

Answered through a review and tabulation of existing bias mitigation methods.

Model experimentation:

SQ3: How do different supervised classifiers (Logistic Regression, Random Forest) perform in terms of fairness and performance before applying bias mitigation?

Answered by training and testing each model on the full dataset and evaluating baseline fairness and performance metrics.

SQ4: What is the impact of applying selected bias mitigation methods on the fairness and performance of each classifier?

Answered by applying bias mitigation techniques and evaluating their effects on both fairness and performance metrics.

Compliance and constraint evaluation:

SQ5: To what extent does the removal of sensitive attributes, in line with data minimisation principles, impact classifier fairness and performance, and how effective are bias mitigation techniques that do not rely on sensitive attribute access (e.g., calibration, regularisation) in addressing these disparities?

Answered by training and evaluating models on the data-minimised dataset, then applying and evaluating bias mitigation techniques applicable under these constraints, and comparing these results against the full-data and sensitive-attribute-aware mitigation conditions.

1.4 Research relevance

1.4.1 Academic relevance

Algorithmic fairness and responsible artificial intelligence are prominent topics in academic research. While various studies have proposed methods for identifying and reducing bias in ML systems, relatively few have evaluated these methods under data minimisation constraints. Many fairness interventions rely on access to sensitive attributes, yet privacy laws may prohibit the use of these attributes in real-world deployment.

This thesis contributes to the academic discourse by empirically testing bias mitigation methods across multiple supervised learning models in both unrestricted and restricted data settings. It provides a structured comparison of performance and fairness outcomes when models are trained without access to sensitive demographic information. Additionally, by focusing on interpretable

models and using the COMPAS dataset as a case study, this thesis offers a grounded and reproducible contribution to ML research. The COMPAS dataset contains real-world risk assessment data from the U.S. criminal justice system. It includes demographic information (such as race, gender, and age), criminal history, and whether an individual reoffended within two years. Its public availability and frequent use in fairness research make it a reliable benchmark for evaluating algorithmic bias, enabling replication and comparison across studies.

1.4.2 Business relevance

As ML becomes increasingly embedded in organisational decision-making, companies face growing pressure to ensure that these systems are both fair and legally compliant. From recruitment and credit assessments to fraud detection and customer analytics, algorithmic bias presents reputational, financial, and legal risks. At the same time, organisations operating within the European Union are required to comply with data protection regulations such as the GDPR, which includes the principle of data minimisation. This principle restricts the collection and use of personal data to what is strictly necessary, often limiting access to sensitive attributes like race, gender, or age. These attributes are frequently used in fairness interventions.

This situation creates a practical challenge: organisations must ensure fairness and transparency in their algorithmic systems while limiting the use of the data that is often required to audit and mitigate bias. By exploring the effectiveness of bias mitigation techniques when sensitive data is unavailable, this research addresses a relevant and timely business problem. It aims to provide practical insights for organisations that wish to develop and deploy ML models that are fair, interpretable, and compliant with privacy regulations.

1.5 Research methods

This study adopts a mixed-methods approach, combining a literature review and an empirical experimental design, to evaluate the effectiveness of bias mitigation techniques under data minimisation constraints. The study follows the CRISP-DM framework, which provides a systematic structure for data preparation, model training, bias mitigation, and evaluation (Schröer et al., 2021). A detailed description of the methodology is provided in Chapter 3.

The research methods were tailored to address each sub-question (SQ) and, collectively, the main research question.

Theoretical understanding (SQ1 & SQ2): These questions were addressed through a comprehensive literature review. SQ1 defined bias and fairness in the context of supervised machine learning classifiers and identified relevant group-level fairness metrics (e.g., demographic parity, equalised odds). SQ2 categorised bias mitigation techniques into pre-processing, in-processing, and post-processing approaches. Together, these findings established the theoretical foundation for the empirical investigation.

Model experimentation (SQ3, SQ4 & SQ5): These questions were addressed through an experimental design using two supervised classifiers, Logistic Regression and Random Forest, applied to the COMPAS dataset. Models were trained and evaluated on both a full version (retaining all features, including race) and a data-minimised version in which race was removed from the training features to reflect GDPR data minimisation requirements. For the full dataset, explicit bias mitigation techniques were applied, including pre-processing (reweighing), in-processing (Exponentiated Gradient Reduction (EGR)), and post-processing (threshold optimiser). For the data-minimised dataset, where direct access to race was restricted, the analysis focused on inherent bias and the effectiveness of mitigation techniques that do not rely on sensitive attributes (e.g., calibration and regularisation). All models were assessed using key fairness and performance metrics to determine the impact of classifier choice and data minimisation constraints.

1.5.1 Research scope

This study focuses on evaluating the effectiveness of bias mitigation techniques in the context of supervised machine learning classifiers, specifically under data minimisation constraints as outlined by the GDPR. The research is limited to two widely used model types: Logistic Regression and Random Forest. These models are used as proxy classifiers to approximate decision-making systems similar to proprietary or black-box algorithms employed in real-world settings.

The COMPAS dataset is selected as the case study due to its relevance in fairness-related research and the presence of sensitive demographic features such as race, gender, and age.

The scope of bias mitigation includes methods applied across different classifiers, with an investigation into both:

- Sensitive-attribute-aware methods: These methods (e.g., reweighing, EGR, and threshold optimiser) require access to sensitive attributes to function, representing a scenario where such data is available for direct fairness intervention.

- Unaware methods: These methods (e.g., calibration and regularisation) do not rely on explicit access to sensitive attributes, representing a scenario where data minimisation is strictly enforced.

Data minimisation is operationalised in this study by removing or excluding sensitive features (e.g., race, age, gender) from the training and prediction process, in order to simulate realistic GDPR-compliant scenarios. Only structured data from the COMPAS dataset is used; unstructured data and external datasets are not considered.

The study does not aim to develop new bias mitigation methods but rather to evaluate existing ones under constrained data conditions. It also does not attempt to audit the original COMPAS algorithm, but rather to explore how similar decision systems may behave under fairness interventions. The findings are intended to inform both academic understanding and practical decision-making in organisational contexts that rely on automated classification systems while adhering to data protection regulations.

1.6 Thesis outline

This thesis is structured as follows.

Chapter 2 introduces the key theoretical concepts relevant to this study. It defines algorithmic bias, fairness, and data minimisation, and explores their intersections in the context of supervised ML. The chapter also includes a structured overview of existing bias mitigation methods, introduces the concept and types of classifiers, and discusses GDPR's implications for ML models.

Chapter 3 presents the detailed research methodology. It expands on the approach summarised in Chapter 1, explaining the research design, data preparation, evaluation metrics, and experimental setup in full. It also justifies the specific selection of classifiers, bias mitigation techniques, and tools used in the analysis.

Chapter 4 presents the experimental results from applying bias mitigation methods across different supervised classifiers using the full dataset. This chapter evaluates how effective these methods are when sensitive attributes are available.

Chapter 5 applies data minimisation by removing sensitive attributes from the training data and retraining the models. It re-evaluates model performance and fairness after applying bias mitigation methods suitable for these conditions, and includes a comparative analysis of models within this data-minimised context.

Chapter 6 provides a direct comparative analysis of the empirical findings from both the full and data-minimised datasets. It quantifies the impact of sensitive attribute removal on baseline model performance and assesses the comparative effectiveness of the applied bias mitigation strategies across these two data conditions, addressing the core research question regarding the necessity of sensitive attributes for achieving fairness.

Chapter 7 discusses the results of the study. It highlights the key findings, considers their implications for both practice and theory, and examines the limitations that may have influenced the results. The chapter concludes by outlining suggestions for future research that could build on or address gaps identified in this study.

Chapter 8 summarises the answers to the research questions, outlines the study's contributions, highlights its limitations, and offers recommendations for both practitioners and future research.

Several chapters refer to the Appendices, where implementation code is provided. All references are included in the text.

2 Introduction to bias mitigation

Organisations across various sectors, from criminal justice to finance and healthcare, rely on decision-making processes and decision support systems (DSS) to navigate complex environments and optimise outcomes (Turban et al., 2005). Effective decision-making is vital, and the quality of these decisions hinges on several critical properties, including accuracy, transparency, and crucially, unbiasedness and fairness (European Commission, 2019). Historically, human biases and organisational factors such as data availability, regulatory pressures, and competing goals have significantly influenced decision-making quality (Arnott, 2006).

In recent decades, the integration of AI and ML systems has profoundly transformed DSS, offering unprecedented capabilities for data analysis and predictive modelling. This transformation, however, has also introduced new forms of bias and unfairness, which can arise at scale and in less transparent ways (Barocas & Selbst, 2016; Goddard et al., 2011). These systems are powerful examples of modern DSS, designed to enhance efficiency and objectivity, but while often perceived as neutral tools, ML algorithms can inadvertently reproduce or even intensify structural inequalities and human biases when trained on historical data, developed with flawed assumptions, or deployed without adequate oversight (Angwin et al., 2016; Noble, 2018).

This challenge has brought fairness and unbiasedness to the forefront of academic and regulatory debates concerning algorithmic decision-making. At the same time, data protection and privacy concerns are escalating. Regulations like the GDPR mandate principles such as data minimisation, which can directly conflict with efforts to mitigate bias because fairness interventions often require access to sensitive attributes to measure and correct disparities (Hardt et al., 2016; Tran & Fioretto, 2023).

This chapter provides the theoretical foundation for the thesis by first establishing the context of quality decision-making and the role of AI/ML. It then explores the importance of early problem framing, including how organisations define fairness and bias in relation to their objectives, and how stakeholder engagement shapes these definitions. Next, it examines definitions and categories of bias and fairness in ML, reviews key mitigation techniques, and considers how data minimisation under the GDPR constrains fairness interventions. The chapter also discusses widely used classifiers and the challenges of deploying models in production environments, including the need for continuous monitoring to maintain fairness and reliability.

To support this discussion, Figure 1 outlines the broader environment in which ML models operate when integrated into DSS. It illustrates how these models are embedded within organisational, legal, and societal structures, shaped by constraints such as the GDPR, fairness expectations, business goals, data limitations, and historical patterns. The figure also shows how model outputs result in decisions, which in turn have real-world consequences for the people affected. This context sets the stage for the theoretical concepts explored in the remainder of the chapter.

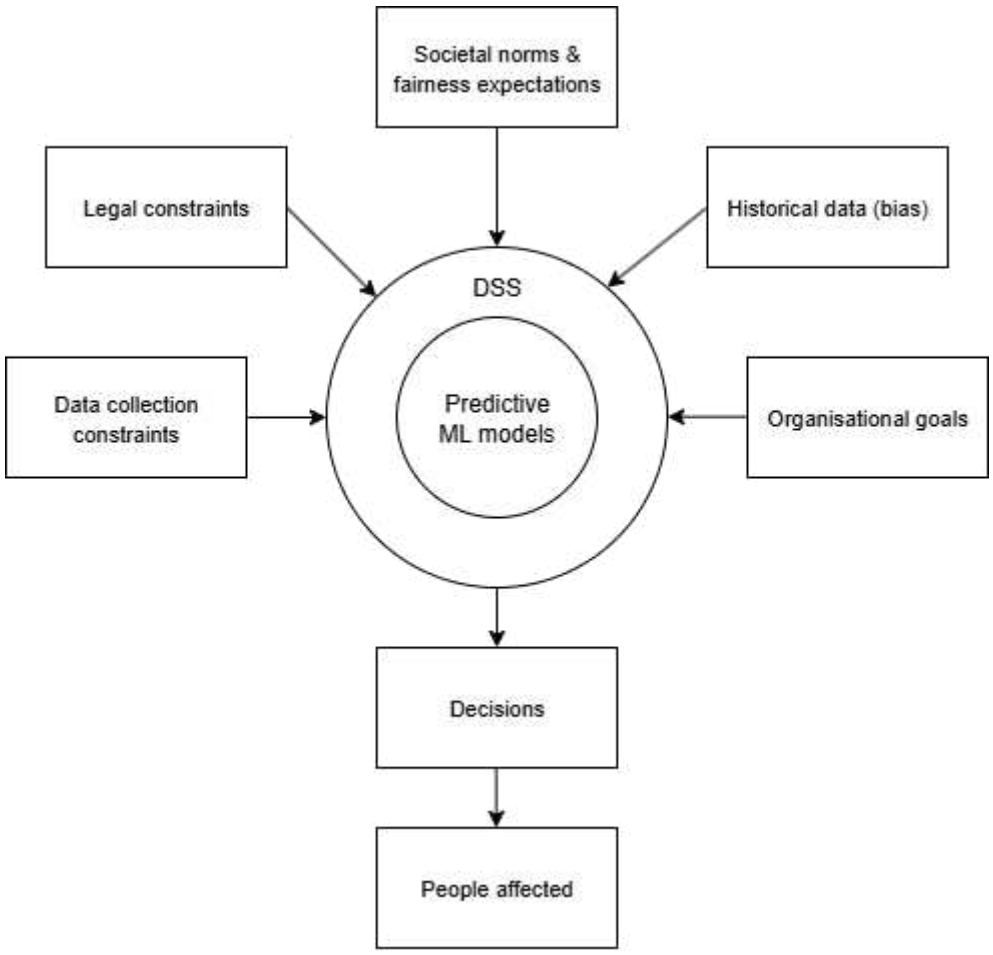

Figure 1. ML models embedded within DSS and their surrounding influences

2.1 The role of COMPAS in fairness research

The COMPAS dataset is one of the most frequently used case studies in algorithmic fairness research. It was originally published by ProPublica as part of a 2016 investigative report that examined potential racial bias in algorithmic risk assessments within the U.S. criminal justice system (Angwin et al., 2016). The COMPAS algorithm itself is proprietary and not publicly available, meaning independent evaluations have relied on the risk score outputs included in the published dataset. The dataset includes demographic variables such as race, sex, and age, as well as criminal history and a

binary outcome indicating whether a defendant reoffended within two years. These features make it well suited for evaluating both model performance and fairness interventions.

ProPublica's analysis concluded that the COMPAS algorithm was racially biased. Specifically, it found that black defendants were almost twice as likely to be falsely labelled as high risk, while white defendants were more likely to be incorrectly classified as low risk. In response, the company that developed COMPAS, Northpointe, argued that the tool was fair because it was calibrated. Calibration, in this context, means that within each risk score category, the observed rate of recidivism was approximately equal across racial groups. This public disagreement revealed a critical tension in the field of algorithmic fairness: it is possible for a model to be calibrated while still producing unequal error rates across groups.

This conflict became a catalyst for formal research into fairness metrics. The main issue identified by ProPublica, disparities in false positive and false negative rates between racial groups, directly led to the development of group fairness metrics such as *equalised odds* and *equal opportunity* (Hardt et al., 2016). These metrics require that models exhibit similar error rates across protected groups. At the same time, subsequent work demonstrated that, under certain conditions, it is mathematically impossible for a classifier to satisfy both calibration and equalised odds simultaneously if base rates differ between groups. This result is now widely referred to as the impossibility theorem of fairness (Chouldechova, 2017; Kleinberg et al., 2017).

These findings had a significant impact on the field. The COMPAS controversy did not simply highlight the existence of bias in a specific system; it also shifted the conversation from general concerns about fairness to a structured, mathematical framework for evaluating and comparing fairness criteria. The case remains foundational in the literature and is often cited as the moment when algorithmic fairness became a rigorous and formalised area of study (Barocas & Selbst, 2016). In addition to its historical significance, the COMPAS dataset continues to serve as a standard benchmark in machine learning fairness research. Its availability and relevance to high-stakes decision-making make it useful for assessing both fairness-aware modelling techniques and the effects of legal constraints such as data minimisation.

2.2 Business understanding

To understand the different phases involved in bias mitigation, this theoretical background is guided by the CRISP-DM framework, a widely adopted standard for organising ML workflows (Chapman

et al., 2000; Schröer et al., 2021). CRISP-DM defines the sequence of phases as *Business Understanding*, *Data Understanding*, *Data Preparation*, *Modelling*, *Evaluation*, and *Deployment*.

The first of these, *Business Understanding*, is particularly critical in the context of fairness-aware ML. The success of any machine learning project depends heavily on how the problem is initially framed. Before any technical work begins, organisations must determine what constitutes fairness and bias within the context of their decision-making objectives. These definitions are shaped by business priorities, stakeholder expectations, and legal or regulatory obligations (Holstein et al., 2023). Failing to establish a shared understanding can lead to models that optimise for narrow technical goals while perpetuating harm or undermining trust.

Fairness is rarely a purely technical construct; it is influenced by organisational values, the needs of affected communities, and broader societal norms. As Barocas, Hardt, and Narayanan (2019) emphasise, fairness is a socio-technical challenge that requires careful consideration of the application context, stakeholder values, and potential societal impacts. Different definitions of fairness exist because the concept is complex and context-dependent, and these decisions must be agreed upon before technical solutions are sought.

Partnership on AI (2024) further argues that engaging diverse stakeholders, particularly those from marginalised communities, is essential for identifying and mitigating biases, foreseeing risks, and ensuring AI systems are equitable and relevant. Early choices about which outcomes to optimise, what data to collect, and which groups to prioritise will significantly shape the success of later mitigation strategies. Many problems that appear downstream, such as biased training data or unsuitable fairness metrics, often originate from incomplete or inconsistent problem definitions at this stage.

2.3 Bias and fairness

Bias in ML refers to any systematic influence arising from algorithmic design, modelling assumptions, data characteristics, or human decisions that causes models to produce unfair, inaccurate, or non-generalisable outcomes (Holmberg et al., 2020). Such bias is particularly concerning in high-stakes contexts, where unequal treatment of individuals or groups can result in social, ethical, and legal harm. Consequently, the concept of fairness is almost always intrinsically linked to people, with the goal of preventing harm to human groups defined by protected attributes (Mehrabi et al., 2019). Fairness in ML is thus a normative goal aimed at counteracting these biases. Although there is no single universally accepted definition, a widely cited description frames fairness

as the absence of prejudice or favouritism toward an individual or group based on their inherent or acquired characteristics (Mehrabi et al., 2019).

A wide range of biases have been identified in the literature. Although there is no unified taxonomy, eight recurring types of bias have been noted across scholarly work:

- Social bias arises from existing inequities embedded in society. When models are trained on data that reflect historical or societal inequalities, those patterns may be reinforced. For instance, underrepresentation of women in leadership roles can result in skewed search engine outputs unless actively corrected (Mehrabi et al., 2019; Olteanu et al., 2019; Zarya, 2018; Suresh & Guttag, 2018).

- Representation bias occurs when the training data is not representative of the broader population. This may be due to outdated data collection, biased sampling, or limited data availability. As a result, certain groups may be over- or underrepresented, reducing model accuracy for those populations (Baer, 2019; Lan et al., 2010).

- Measurement bias refers to the use of flawed features or labels that do not accurately reflect the intended variables. Protected attributes such as race or gender may be directly or indirectly encoded in other variables, producing discriminatory outcomes even if explicitly excluded from the dataset (Mullainathan & Obermeyer, 2017; Corbett-Davies et al., 2023; van Giffen et al., 2022; Angwin et al., 2016).

- Label bias emerges when training examples are assigned incorrect or culturally specific labels that deviate from their true classifications. For example, image classification models may reflect Western assumptions about events like weddings, leading to systematic mislabelling of culturally distinct examples (Barocas & Selbst, 2016; Baer, 2019).

- Technical bias is introduced through design decisions, such as how abstract concepts are formalised into code or how hyperparameters are chosen. This includes statistical assumptions embedded in models that systematically favour certain outputs over others (Friedman & Nissenbaum, 1996).

- Evaluation bias arises when benchmark datasets used to assess model performance are themselves unrepresentative. As a result, a model may perform well in testing but fail in real-world contexts where the data distribution is more diverse (Suresh et al., 2018; Suresh & Guttag, 2021).

- Deployment bias occurs when models are used outside the scope of their original design. For example, predictive models built to assess risk may instead be applied to determine sentencing length, despite lacking validation for such use (Chouldechova, 2017; Collins, 2018).

- Feedback bias happens when a model's predictions influence the future data on which it is retrained. This creates self-reinforcing cycles in applications such as recommendation systems, where certain content is repeatedly shown and thus increasingly clicked, regardless of actual user preference (Bellamy et al., 2018; Baeza-Yates, 2018).

In the CRISP-DM framework, the *Data Understanding* phase provides an opportunity to evaluate the likelihood of biases being present in the dataset before modelling begins (Schröer et al., 2021). While not all forms of bias originate from the data itself, several types, such as representation, measurement, and label bias, can be identified at this stage.

To assess fairness in light of these biases, scholars have proposed a range of formal fairness metrics. These include demographic parity, which checks whether different groups receive positive predictions at equal rates, and equalised odds, which requires similar false positive and false negative rates across groups (Hardt et al., 2016). However, these metrics are often mathematically incompatible, and no single metric fully captures the multidimensional nature of fairness (Binns, 2018).

Although the primary focus of this thesis is on fairness, a model's performance remains an important baseline measure in machine learning. Maintaining adequate predictive performance ensures that fairness interventions do not render models ineffective in practice. Standard performance metrics such as accuracy score, F1 score, and ROC AUC (Receiver Operating Area Under the Curve) are therefore used as comparative baselines in later chapters, even though the main research objective is to reduce bias and improve fairness (Caruana & Niculescu-Mizil, 2006).

2.4 Data minimisation and GDPR

The increasing adoption of ML across domains such as healthcare, finance, and public administration has raised concerns about data protection and privacy. Within this context, the principle of data minimisation has gained prominence as a safeguard against the misuse of personal data. It is a core feature of the GDPR and influences both the design and operation of ML systems (Article 5(1)(c)).

2.4.1 Legal basis and purpose

The GDPR outlines data minimisation in Article 5(1)(c)[1], which states that personal data must be "adequate, relevant and limited to what is necessary in relation to the purposes for which they are processed." This principle obliges data controllers to justify each element of personal data they collect, ensuring that no unnecessary data is gathered or retained. According to Recital 39, data should be kept to a strict minimum, and controllers must periodically assess whether the personal data they process is still necessary.

Furthermore, data minimisation is closely connected to other GDPR principles, including purpose limitation (Article 5(1)(b)), storage limitation (Article 5(1)(e)), and accuracy (Article 5(1)(d)). For example, if personal data is no longer required for the stated purpose, the controller is expected to delete or anonymise it (Recital 39; Article 5(1)(e)). Similarly, inaccurate or excessive data must be rectified or removed to prevent harm to the data subject (Article 5(1)(d)).

The GDPR also requires that these principles are implemented with data protection by design and by default (Article 25). This involves including privacy considerations at every stage of processing and ensuring that only the data necessary for a specific task is collected, accessed, and stored (Recital 78).

Moreover, Article 9 of the GDPR specifically prohibits the processing of special categories of personal data, including attributes such as race, ethnicity, and health status, unless specific exceptions apply. These attributes are often essential for detecting and mitigating algorithmic bias. As a result, Article 9, combined with the principle of data minimisation, can significantly constrain the ability to apply fairness interventions in practice (Hardt et al., 2016; Tran & Fioretto, 2023).

2.4.2 Implementation challenges

Although the GDPR provides a clear legal foundation for data minimisation, implementing it in ML systems presents challenges. One issue is the lack of a formal, technical definition that aligns with legal requirements. Ganesh et al. (2025) argue that the principle is conceptually sound but difficult to operationalise, as it depends heavily on context. What is necessary in one scenario may be excessive in another.

Another consideration is identifiability. The European Data Protection Board (2020) advises that if identifiability is not necessary for the intended purpose, personal data should be anonymised or

[1] General Data Protection Regulation (GDPR), Regulation (EU) 2016/679. You can find the full text of the GDPR here: https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:32016R0679

pseudonymised (Recital 26; Article 25(1)). Controllers must therefore assess not only the necessity of each data point, but also the risk it poses to individual privacy across the data lifecycle (Recital 28).

2.4.3 Implications for ML development

In ML, data minimisation intersects with concerns around model accuracy, generalisability, and fairness. Reducing the level of detail or volume of input data can help prevent overfitting and reduce the risk of leaks, but it may also impair the model's ability to identify patterns. This trade-off is especially problematic in sensitive applications such as fraud detection or medical diagnosis, where model performance can have real-world consequences (Sonboli et al., 2024).

A practical challenge is determining which features are necessary. Feature selection becomes a legal as well as a technical question, since using irrelevant or excessive attributes could violate the necessity requirement under Article 5(1)(c). Within CRISP-DM, such considerations arise during *Data Understanding* and continue into *Data Preparation,* when datasets are refined for modelling (Schröer et al., 2021). Moreover, data minimisation may complicate fairness interventions. While reducing sensitive variables may protect privacy, it can also hinder efforts to detect or correct bias, especially in models that rely on group-specific calibration (Tran & Fioretto, 2023).

2.4.4 Accountability

To meet the GDPR's requirement for data protection by design and by default (Article 25), controllers must implement both technical and organisational safeguards. The European Data Protection Board (2020) identifies several best practices. These include limiting access to personal data, aggregating data where possible, and avoiding collection altogether if it is not essential for the task (Recital 78).

Controllers are also responsible for demonstrating compliance. This includes documenting why specific data points are collected, how they support the processing purpose, and whether alternatives were considered (Article 5(2); Recital 74). While these safeguards are essential for protecting individual privacy, they may unintentionally complicate efforts to ensure fair decision-making. Many bias mitigation strategies rely on access to sensitive attributes such as race or gender to identify and correct disparities. Under data minimisation constraints, however, this information may be unavailable, making it more challenging to detect and address bias effectively (Tran & Fioretto, 2023). The following section outlines the types of bias mitigation techniques available and considers how these constraints influence their applicability and effectiveness.

2.5 Bias mitigation techniques

Bias mitigation refers to interventions designed to reduce the unfairness in ML models caused by systematic distortions in data, model design, or implementation (Hort et al., 2024). These strategies are crucial for supporting fairness and transparency in DSS across both development and deployment stages.

According to Hort et al. (2024), bias mitigation methods can be grouped based on when the intervention takes place in the ML pipeline:

1. *Pre-processing*: mitigation in the training data, to prevent bias from reaching ML models (see Section 2.5.1);
2. *In-processing*: mitigation while training ML models (see Section 2.5.2);
3. *Post-processing*: mitigation on trained ML models (see Section 2.5.3).

2.5.1 Pre-processing techniques

Pre-processing techniques are applied during the data preparation stages of the ML pipeline and can be closely linked to the *Data Understanding* and *Data Preparation* phases of the CRISP-DM framework (Schröer et al., 2021). At a broad level, pre-processing involves cleaning, transforming, and selecting features to ensure the data is representative, accurate, and suitable for modelling. In the context of fairness, these techniques are particularly effective when biased patterns in the input data are likely to propagate through to the model's predictions (Feldman et al., 2015).

Examples include:

- Reweighting: Adjusting instance weights so that underrepresented groups have a proportionally larger influence during training (Hajian & Domingo-Ferrer, 2013; see Section 3.4.1).
- Data augmentation: Expanding datasets with synthetic or additional data to balance group representation (Chen et al., 2018).
- Disparate impact remover: Transforming input features to reduce their correlation with protected attributes (Friedler et al., 2019).

2.5.2 In-processing techniques

In-processing techniques are applied during the *Modelling* phase of CRISP-DM (Schröer et al., 2021). These techniques incorporate fairness constraints or modify the loss function to discourage discriminatory patterns (Zafar et al., 2017).

Examples include:

- Exponentiated Gradient Reduction (EGR): Iteratively trains a base classifier on reweighted data to find a fair predictor that satisfies a given fairness constraint (Agarwal et al., 2018; see Section 3.4.2).
- Regularisation: Adds penalty terms to the loss function or constrains model complexity during training to prevent overfitting and encourage more generalised predictions (Hastie et al., 2009; see Section 3.4.2).
- Latent variable models: Using hidden variables to uncover and adjust for bias in labelled data (Kamiran & Calders, 2009).
- Adversarial debiasing: Training the model to produce outputs that are not predictable from protected attributes, by using an adversary network (Yang et al., 2023).

2.5.3 Post-processing techniques

Post-processing techniques are useful when access to the training process is limited or when constraints prevent altering the model itself (Hardt et al., 2016). These techniques align most closely with the *Evaluation* phase of CRISP-DM (Schröer et al., 2021).

Examples include:

- Threshold optimiser: Adjusts classification thresholds, potentially on a group-specific basis, to align with desired fairness criteria on a trained model's probability outputs (Hardt et al., 2016; see Section 3.4.3).
- Calibration: Ensures a model's predicted probabilities accurately reflect the true likelihood of an outcome, thereby improving the reliability of its confidence scores (Zadrozny & Elkan, 2002; see Section 3.4.3).

- Reject option classification: This method targets predictions with high uncertainty and tends to assign favourable outcomes to the unprivileged group and unfavourable outcomes to the privileged group (Kamiran et al., 2012).
- Calibrated equalised odds post-processing: This method operates on calibrated classifier score outputs and finds probabilities with which to change output labels to optimise for equalised odds (Pleiss et al., 2017).
- Equalised odds post-processing: This method solves a linear program to determine the probabilities with which to alter output labels, aiming to satisfy the equalised odds criterion (Hardt et al., 2016).

2.5.4 Summary of bias mitigation techniques

Table 1. Bias mitigation methods

Summarises the bias mitigation methods discussed in this section, grouped by the stage in the ML pipeline at which they are applied.

Stage	Method	Description	Key references
Pre-processing	Reweighting	Adjusts instance weights so that underrepresented groups have a proportionally larger influence during training.	Hajian & Domingo-Ferrer (2013); Yan et al. (2022); Zhao et al. (2023)
	Data augmentation	Expands datasets with synthetic or additional data to balance group representation.	Chen et al. (2018); Hastings Blow et al. (2025)
	Disparate impact remover	Transforms input features to reduce their correlation with protected attributes, aiming to reduce indirect discrimination.	Friedler et al. (2019); Wang & Singh (2025)

Stage	Method	Description	Key references
	Feature selection/cleaning	Removes or transforms biased features during data preparation to prevent harmful patterns from propagating into the model. (Broad category relevant to fairness)	Feldman et al. (2015); Kamalov et al. (2025)
In-processing	Exponentiated Gradient Reduction	Iteratively trains a base classifier on reweighted data while updating penalties for fairness violations, producing a fairer predictor.	Agarwal et al. (2018); Farayola et al. (2025)
	Regularisation	Adds penalty terms to the loss function or constrains model complexity to encourage more generalised predictions and reduce spurious correlations.	Hastie et al. (2009); Rabonato & Beron (2025)
	Latent variable models	Uses hidden variables to detect and adjust for bias in labelled data.	Kamiran & Calders (2009); Al-Zawqari et al. (2024)

Stage	Method	Description	Key references
	Adversarial debiasing	Trains the model alongside an adversary network that predicts protected attributes, encouraging the main model to produce outputs independent of those attributes.	Yang et al. (2023); Yang et al. (2024)
Post-processing	Threshold optimiser	Adjusts classification thresholds (possibly group-specific) to satisfy fairness criteria such as equalised odds or demographic parity.	Hardt et al. (2016); Minatel et al. (2025)
	Calibration (Platt scaling / Isotonic)	Adjusts the predicted probabilities to ensure they reflect the true likelihood of outcomes, improving confidence estimates and, in some cases, fairness.	Zadrozny & Elkan (2002); Nikolić et al. (2025)
	Reject option classification	Assigns favourable outcomes to disadvantaged groups and unfavourable ones to advantaged groups when the model's predictions are uncertain.	Kamiran et al. (2012); Siddique et al. (2024)

Stage	Method	Description	Key references
	Equalised odds post-processing	Solves a linear program to determine probabilities of flipping output labels to satisfy equalised odds constraints.	Hardt et al. (2016); Awasthi et al. (2020)
	Calibrated equalised odds post-processing	Works on calibrated score outputs to flip output labels with optimised probabilities that satisfy equalised odds while preserving calibration.	Pleiss et al. (2017); Raftopoulos et al. (2025)

2.6 Classifiers

Within the realm of ML, supervised learning is particularly prominent in DSS (Silva & Bernardino, 2022). It involves training algorithms on labelled data so they can generalise and predict outcomes for new, unseen instances (Jiang et al., 2020). One of the most common supervised learning tasks is classification, which categorises inputs into predefined classes (Hastie et al., 2009). Classification has led to the development of a range of algorithms, each with different strengths in terms of accuracy, learning speed, complexity, and susceptibility to overfitting (Singh et al., 2016).

The *Modelling* phase of CRISP-DM involves selecting and training a classifier that meets the project's objectives. This section provides an overview of some of the most prominent supervised classifiers used in ML today, as identified by IBM (2024). These include:

Logistic Regression,

Decision Trees,

Random Forests,

Naive Bayes,

Support Vector Machines, and

K-Nearest Neighbours (see Sections 2.6.1 to 2.6.6).

Within this broad landscape, this thesis focuses on Logistic Regression and Random Forest, which are widely used in diverse domains and represent contrasting approaches to supervised learning. Further justification for the selection of these classifiers is provided in Chapter 3 (Methodology). The following subsections describe each classifier in detail.

2.6.1 Logistic Regression

According to Caraciolo (2011), LR is a linear model commonly used for binary and multiclass classification tasks. It estimates the probability that a given input belongs to a particular class using the logistic (sigmoid) function. The model assumes a linear relationship between the input features and the log-odds of the outcome, making it interpretable and relatively easy to implement. Logistic LR is especially effective when the relationship between features and the output is approximately linear.

2.6.2 Decision Trees

According to Rokach and Maimon (2005), decision tree classifiers use a tree-like structure to recursively split the feature space based on attribute values, creating a series of decision rules that lead to class predictions. Each internal node represents a test on a feature, each branch corresponds to an outcome of the test, and each leaf node represents a class label. Decision trees are non-parametric and capable of capturing non-linear relationships between features and outcomes. They are also highly interpretable and can handle both categorical and numerical input data.

2.6.3 Random Forests

This study uses Random Forests, a more robust ensemble method introduced by Breiman (2001). Instead of relying on a single decision tree, a Random Forest builds multiple decision trees during training and aggregates their predictions, typically using majority voting for classification. Each tree is trained on a bootstrapped sample of the data and considers only a random subset of features when splitting nodes. This introduces diversity among the trees, reduces overfitting, and improves generalisation. While Random Forests are less interpretable than individual decision trees, they offer significantly improved accuracy and stability, making them a strong baseline for evaluating the impact of bias mitigation and data minimisation.

2.6.4 Naive Bayes

According to Reddy et al. (2022), naive Bayes classifiers are based on Bayes' Theorem and assume that features are conditionally independent given the class label. Despite this strong assumption, naive Bayes models often perform well in high-dimensional spaces and are computationally efficient. Several variants exist, each tailored to specific data types:

- Multinomial naive Bayes is typically used for discrete features, such as word counts in text classification.
- Bernoulli naive Bayes handles binary-valued features.
- Gaussian naive Bayes assumes that features follow a normal (Gaussian) distribution and is well suited for continuous data.

Gaussian naive Bayes remains popular due to its simplicity and effectiveness, especially when the independence assumption approximately holds (Reddy et al., 2022).

2.6.5 Support Vector Machines

Introduced by Cortes and Vapnik (1995), Support Vector Machines (SVMs) are supervised learning models primarily used for binary classification tasks. They operate by mapping input data into a high-dimensional feature space, where the algorithm constructs an optimal separating hyperplane that maximises the margin between different classes. This margin-based approach ensures strong generalisation performance. SVMs can handle both linearly separable and non-separable data through the use of kernel functions, which enable the creation of non-linear decision boundaries. By relying only on a subset of the training points known as support vectors, SVMs achieve computational efficiency and robustness. These characteristics make SVMs a versatile and widely adopted choice in various classification problems.

2.6.6 K-Nearest Neighbours

The foundations of the K-Nearest Neighbours (KNN) algorithm were first introduced by Fix and Hodges (1951), who proposed the concept of non-parametric discrimination in a technical report. This conceptual groundwork was later expanded and formalised by Cover and Hart (1967) in their seminal paper Nearest Neighbor Pattern Classification, which provided rigorous theoretical analysis and popularised the method within the scientific community. KNN is a non-parametric classification algorithm that assigns a data point to the most common class among its k nearest neighbours in the

feature space, using a distance metric such as Euclidean distance to determine proximity. As a lazy learning method, KNN does not construct an explicit model during training; instead, classification is deferred until prediction time, when the nearest neighbours of a query instance are evaluated. While KNN can handle complex decision boundaries and is intuitive to implement, its performance depends on the choice of k and the scaling of features, and it can become computationally expensive for large datasets (Suyal & Goyal, 2022).

2.6.7 Compatibility with bias mitigation techniques

The compatibility and effectiveness of bias mitigation techniques vary depending on the classifier used (Friedler et al., 2019). While some methods integrate easily and are widely applicable, others require significant adaptation or cannot be directly applied due to the classifier's inherent characteristics. Table 2 summarises the compatibility of the bias mitigation methods discussed in this thesis across six prominent classifiers.

- *Compatible*: The mitigation method is generally well-suited and commonly applied to this classifier, often with straightforward integration or as a meta-algorithm that can wrap it as a base classifier.
- *Limited***:** The mitigation method is technically possible but might not be as effective, require significant adaptation, or is less commonly applied due to the classifier's inherent characteristics or typical use cases.
- *Not applicable***:** The mitigation method's core mechanism does not align with the classifier's typical training process or architecture, making direct application technically impossible or without any meaningful effect.

Table 2. Compatibility of bias mitigation techniques with six prominent classifiers

Bias Mitigation Technique	Stage	Logistic Regression	Decision Trees	Random Forests	Naive Bayes	Support Vector Machines	K-Nearest Neighbours
Reweighting	Pre-processing	Compatible	Compatible	Compatible	Compatible	Compatible	Compatible
Data augmentation	Pre-processing	Compatible	Compatible	Compatible	Compatible	Compatible	Compatible
Disparate impact remover	Pre-processing	Compatible	Compatible	Compatible	Compatible	Compatible	Compatible
Feature selection / cleaning	Pre-processing	Compatible	Compatible	Compatible	Compatible	Compatible	Compatible
Exponentiated Gradient Reduction	In-processing	Compatible	Compatible	Compatible	Compatible	Compatible	Compatible
Regularisation	In-processing	Compatible	Limited	Limited	Not applicable	Compatible	Not applicable
Latent variable models	In-processing	Limited	Limited	Limited	Limited	Limited	Limited
Adversarial debiasing	In-processing	Limited	Limited	Limited	Limited	Limited	Limited
Threshold optimiser	Post-processing	Compatible	Compatible	Compatible	Compatible	Compatible	Compatible
Calibration (Platt / Isotonic)	Post-processing	Compatible	Compatible	Compatible	Compatible	Compatible	Compatible
Reject option classification	Post-processing	Compatible	Compatible	Compatible	Compatible	Compatible	Compatible
Equalised odds post-processing	Post-processing	Compatible	Compatible	Compatible	Compatible	Compatible	Compatible
Calibrated equalised odds post-processing	Post-processing	Compatible	Compatible	Compatible	Compatible	Compatible	Compatible

2.7 Deployment

The *Deployment* phase is one of the most challenging parts of the ML lifecycle and has a direct impact on fairness and reliability. Moving a model from development to production often exposes issues such as integration difficulties, scalability constraints, and the absence of clear operational practices (Paleyes et al., 2022; Zimelewicz et al., 2024). Many of these challenges arise because deployment is not treated as an ongoing process, with continuous monitoring and maintenance, but rather as a one-off technical step.

MLOps frameworks aim to address this by enabling continuous delivery and systematic monitoring of models in production (Kreuzberger et al., 2023). However, they remain underutilised, and post-deployment monitoring often focuses narrowly on inputs and outputs rather than on metrics such as fairness or interpretability. This lack of holistic monitoring increases the risk that models degrade over time or that emerging biases go undetected as data distributions shift (Ferrara, 2024).

Strengthening deployment practices requires greater emphasis on operational readiness and accountability for fairness at this stage. Continuous monitoring, retraining pipelines, and clearly defined responsibilities for fairness and reliability can help ensure that models remain trustworthy beyond initial development (Paleyes et al., 2022; Ferrara, 2024).

2.8 Summary

This chapter has provided the theoretical foundation for investigating fairness in supervised machine learning under data minimisation constraints. It began by examining the importance of early problem framing, highlighting how definitions of fairness and bias are shaped by organisational priorities, stakeholder expectations, and societal norms. It then defined bias and fairness, outlined eight types of bias that can arise throughout the ML pipeline, and reviewed key fairness metrics such as demographic parity and equalised odds. Bias mitigation strategies were categorised into pre-processing, in-processing, and post-processing techniques, highlighting when and how these interventions can reduce unfairness. The chapter also introduced a range of widely used classifiers, including Logistic Regression, Decision Trees, Random Forests, Naive Bayes, Support Vector Machines, and K-Nearest Neighbours, and discussed how their underlying assumptions may influence the effectiveness of bias mitigation. It then examined the legal and technical implications of the GDPR's data minimisation principle, including how it shapes feature selection and constrains fairness interventions, before considering the deployment stage and the importance of continuous monitoring and operational readiness to maintain fairness and reliability over time. Finally, it

presented the conceptual framework that connects these elements. In this framework, bias mitigation techniques act as a mediating variable between sources of bias and the outcomes of fairness and performance. The effectiveness of these techniques may be influenced by two moderating variables: the type of classifier and the presence of data minimisation constraints. This framework provides the basis for the empirical analysis that follows.

3 Methodology

This chapter provides a detailed explanation of the methodological approach used in this study, building on the outline presented in Chapter 1 and following the CRISP-DM framework as introduced in Section 2.2. The phases most relevant to this research are as follows: *Business and Data Understanding* is addressed by defining the research objectives and describing the COMPAS dataset; *Data Preparation* is carried out through the creation of full and data-minimised versions of the dataset; *Modelling* encompasses the selection of classifiers and the application of bias mitigation techniques; and *Evaluation* is conducted using a comprehensive suite of fairness and performance metrics. As this study is experimental, no deployment phase was undertaken. The chapter is structured to follow this progression, concluding with a summary of how these components collectively address the research question.

3.1 Research design

This study adopts a structured experimental design to evaluate the effectiveness of bias mitigation techniques under data minimisation constraints in supervised machine learning. As outlined in the methodology introduction, the research follows the CRISP-DM sequence to ensure a systematic approach. The main objective is to assess whether fairness with respect to race can still be achieved when the race attribute is removed from the training data, as required under GDPR's data minimisation principles. To explore this, the COMPAS dataset is used as a case study, given its real-world relevance and inclusion of sensitive variables such as race, gender, and age. Two versions of the dataset are prepared: one containing all features and one that excludes the race attribute from the training set in line with data minimisation principles. Two classifiers are examined: Logistic Regression and Random Forest. Each model is evaluated in both its baseline form and after the application of bias mitigation methods. The outcomes of interest are fairness, assessed through group-level metrics such as demographic parity and equalised odds for racial groups, and model performance, evaluated using standard performance measures such as accuracy, F1 score, and ROC AUC. This design enables a comparative analysis of how fairness and predictive performance are shaped by model choice, mitigation strategy, and data availability.

3.2 Dataset and preprocessing

This study uses the COMPAS dataset published by ProPublica, which includes pre-trial risk assessment scores and criminal history data for defendants from Broward County, Florida (Angwin et al., 2016). The dataset contains 7,214 records and 53 features, including sensitive demographic

attributes such as race, sex, and age, as well as variables related to prior offences and charge severity. No missing values were present in the dataset.

The racial distribution is notably imbalanced, with 3,696 African-American, 2,454 Caucasian, 637 Hispanic, 377 Other, 32 Asian, and 18 Native American defendants (Figure 2). In terms of sex, the dataset contains 5,819 male and 1,395 female individuals. The binary target variable `two_year_recid`, which indicates whether a defendant reoffended within two years, has 3,963 negative cases (0) and 3,251 positive cases (1).

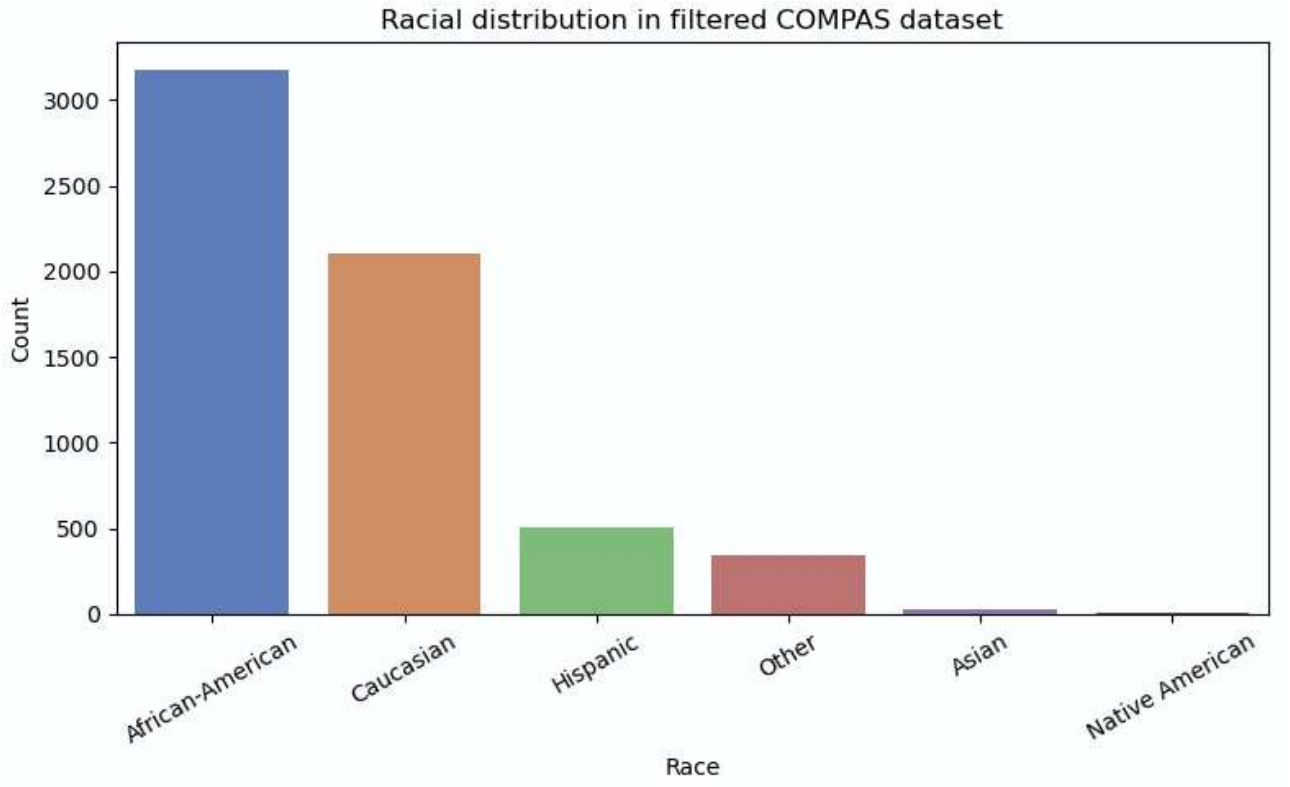

Figure 2. Racial distribution of defendants in the COMPAS dataset

To ensure data quality and consistency with prior studies, several preprocessing steps were applied. First, two date columns (`compas_screening_date` and `out_custody`) were converted to datetime format, and any rows with missing values in these fields were removed. A series of filters were then applied to refine the dataset: entries were retained only if the number of days between arrest and COMPAS screening fell within ±30 days; cases with missing recidivism data (`is_recid = -1`), non-criminal offences (`c_charge_degree = 'O'`), or missing COMPAS scores (`score_text = 'N/A'`) were excluded. These steps mirror the filtering logic used in foundational analyses of this dataset.

After filtering, five features were selected for modelling: *sex*, *age*, *race*, *priors_count*, and *c_charge_degree*. These were chosen for their relevance in both prior research and fairness auditing.

To ensure that removing race from the data-minimised version would not inadvertently leave strong proxy variables for race in the dataset, the relationships between race and the remaining non-race features were examined. To avoid redundancy from perfectly collinear dummy variables, only *sex_Male* and *c_charge_degree_M* were included, as *sex_Female* and *c_charge_degree_F* would provide identical information with opposite sign.

Figure 3 shows a heatmap of Pearson correlation coefficients between race (one-hot encoded to retain all categories) and non-race features. All correlation values were ≤ 0.22, indicating weak linear relationships between race and any single non-race feature.

Figure 4 presents mutual information scores between race and each non-race feature. The scores were all low, with *age* (0.009) and *priors_count* (0.006) showing the strongest dependencies. *c_charge_degree_M* (0.002) and *sex_Male* (0.001) had notably weaker associations. These analyses confirm that while overall dependence is low, age and priors_count are the most significant potential proxies for race. This supports the assumption that removing the race variable reduces, but does not entirely eliminate, the model's indirect access to racial information.

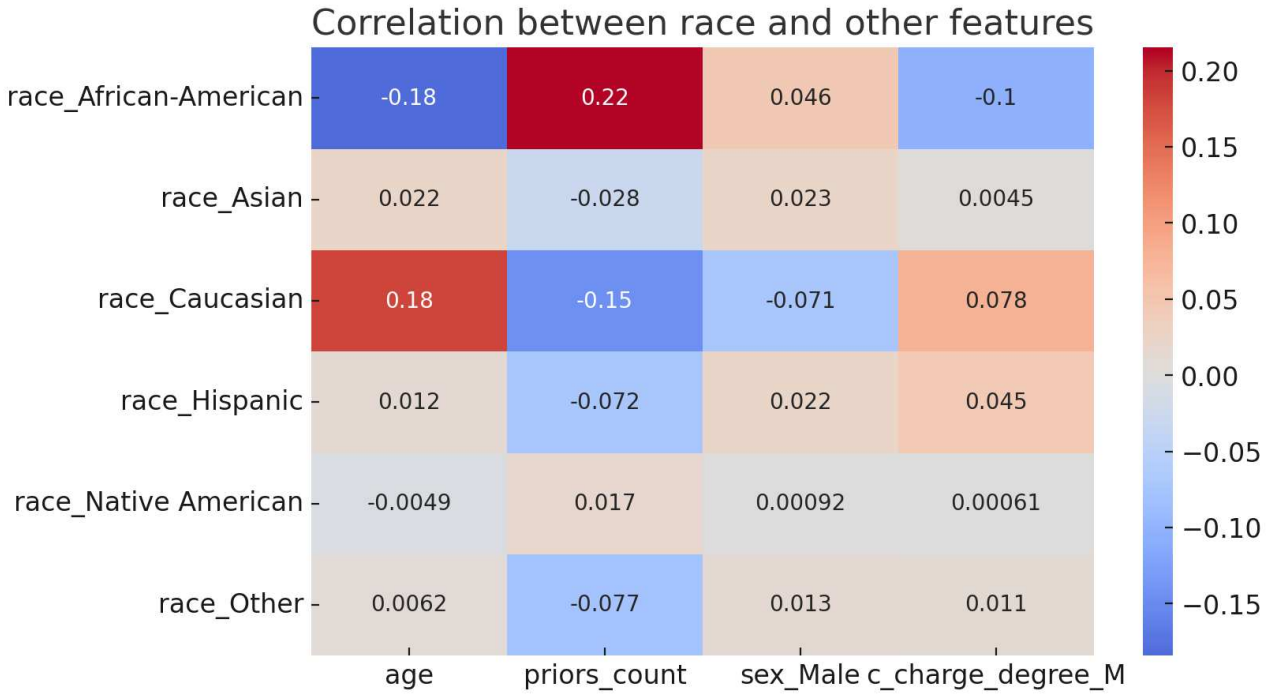

Figure 3. Pearson correlation coefficients between race and non-race features in the COMPAS dataset

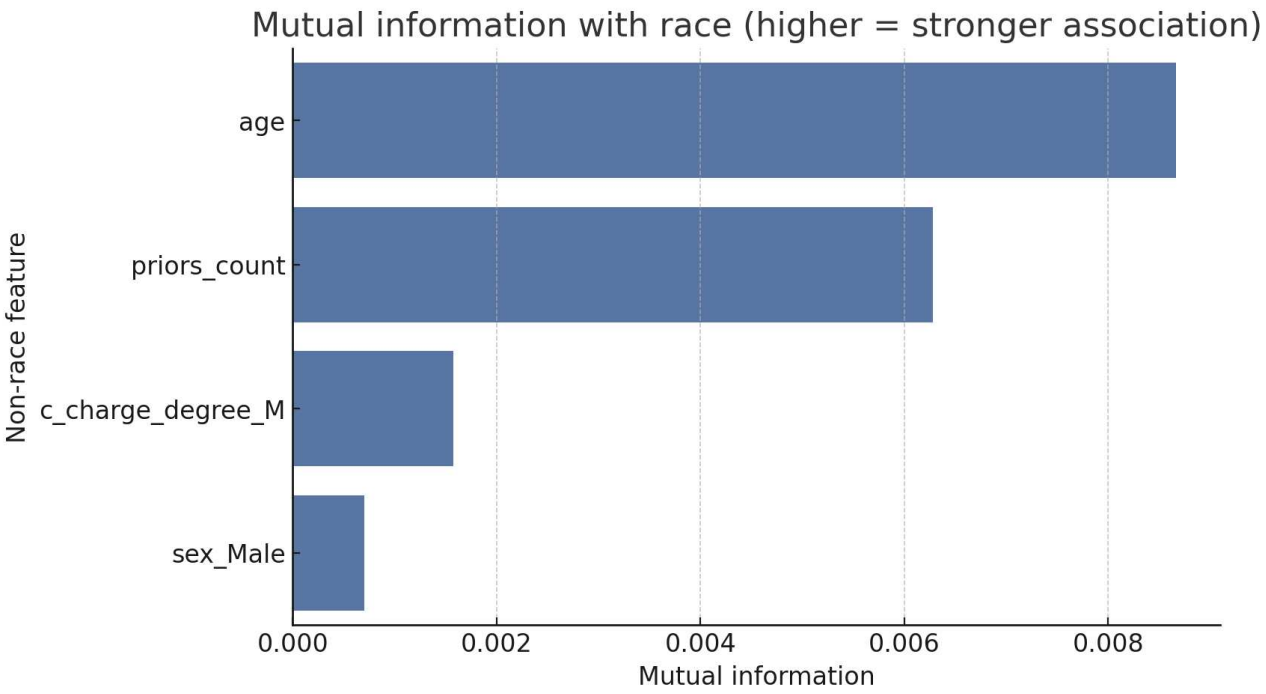

Figure 4. Mutual information scores between race and non-race features in the COMPAS dataset

The categorical variables (*sex*, *race*, and *c_charge_degree*) were one-hot encoded using pandas' *get_dummies()* function with the *drop_first=True* parameter to avoid multicollinearity. This transformation enabled compatibility with models expecting numerical input.

Two distinct versions of the dataset were prepared for the experiments:

- Full dataset: This version retained all selected features as inputs for model training.
- Data-minimised dataset: For this version, the attribute *race* was explicitly excluded from the input features provided to the models' training phase, in line with GDPR's data minimisation principles focusing on sensitive categories. Sex and age, while potentially sensitive in other contexts, were retained as general features, as the primary fairness evaluation of this study is focused on race. However, the attribute *race* was still retained separately (not as a model input) for post-training fairness evaluation across racial subgroups.

Finally, the data was split into training and test sets using stratified sampling to preserve the original class distribution of the target variable (*two_year_recid*), with 70% allocated to training and 30% to testing. For fairness evaluation, the protected attribute race was consistently retained and split alongside the input features and labels, regardless of whether it was used as model input, ensuring

consistent subgroup analysis. The full Python code implementing the dataset analysis and preprocessing steps described in this section is provided in Appendix 9.1.

3.3 Model selection

The selection of classification algorithms for this study aims to provide a diverse analytical foundation for assessing bias mitigation and data minimisation. From the various machine learning techniques (Singh et al., 2016), Logistic Regression and Random Forest were chosen due to their distinct underlying assumptions and mechanisms.

Logistic Regression was included as a representative linear model, valued for its interpretability and ease of implementation (Caraciolo, 2011). Its straightforward nature makes it a transparent baseline for evaluating the direct impact of bias mitigation and data minimisation, especially where relationships are approximately linear.

Conversely, Random Forest provides a robust and high-performing ensemble approach. Built from multiple decision trees (Rokach and Maimon, 2005), it significantly improves accuracy, stability, and generalisation by reducing overfitting compared to single models (Breiman, 2001). Despite its lower interpretability, RF's strong predictive power offers a powerful comparison, enabling a comprehensive examination of how model complexity interacts with bias mitigation and data availability.

All classifiers were implemented using the scikit-learn library (Pedregosa et al., 2011). The LR model was instantiated with an increased iteration limit (`max_iter=1000`) to ensure convergence but otherwise relied on default settings, including L2 regularisation. The RF model was also used with its default configuration, aside from setting a fixed `random_state` for reproducibility. The choice to use unmodified base models reflects a deliberate focus on assessing fairness and mitigation effectiveness under standard deployment conditions, rather than optimising classifier performance through hyperparameter tuning.

For one specific analysis, the statistical significance testing of features in the baseline LR model, an equivalent model was refitted using the Statsmodels implementation of LR. This version is optimised for statistical inference and provides outputs such as standard errors and p-values, which are not available in scikit-learn’s predictive-focused implementation.

3.4 Applied bias mitigation methods

This study employs a diverse set of bias mitigation methods, strategically selected from each stage of the machine learning pipeline: pre-processing, in-processing, and post-processing. Each method is first evaluated in isolation to provide a clear understanding of its individual impact on fairness and predictive performance. Comparisons are then made across categories to determine how methods from different stages perform relative to one another. This design provides both detailed insight into the strengths and limitations of each method and a broader perspective on which approaches are most effective when access to sensitive attributes is limited (Hort et al., 2024).

3.4.1 Pre-processing techniques

Pre-processing techniques modify the training data itself, aiming to prevent bias propagation into the model (Feldman et al., 2015),

Reweighting has been chosen as a representative pre-processing method for this study. This method adjusts instance weights so that underrepresented groups exert a proportionally larger influence during model training (Hajian & Domingo-Ferrer, 2013). Unlike oversampling or undersampling, which physically duplicate or remove data points, reweighting assigns a numerical weight to each data instance. This weight is then incorporated into the model's loss function during training, effectively changing the importance of each sample without altering the dataset's physical size or composition. Reweighting is particularly suitable for this study as it directly manipulates the data distribution to achieve fairness without altering the feature space or requiring model-specific modifications. This makes it a valuable method for assessing whether balancing group representation at the data level can effectively mitigate bias, especially in the context of sensitive attribute removal.

3.4.2 In-processing techniques

In-processing techniques intervene during the model training phase, often by incorporating fairness constraints or modifying the loss function (Zafar et al., 2017). This study implements EGR and regularisation as in-processing methods.

EGR is selected for its ability to iteratively train a base classifier on reweighted data to find a fair predictor that satisfies a given fairness constraint (Agarwal et al., 2018). EGR works by framing the fairness problem as a constrained optimisation task: it alternates between updating the classifier’s weights to minimise prediction error and adjusting a set of “dual variables” that penalise violations of the fairness constraint. This exponentiated update rule increases the penalty on fairness violations

multiplicatively, gradually steering the model towards a solution that balances accuracy and fairness. Its theoretical guarantees regarding the trade-off between fairness and accuracy make it a robust choice for assessing constrained optimisation in the presence of data minimisation. This method is particularly useful because it is model agnostic, acting as a wrapper around existing classification models (like Logistic Regression and Random Forest), and is capable of optimising for various fairness metrics. In this study, equalised odds was selected as the fairness constraint for all EGR experiments as it aligns with the group-level fairness metrics evaluated in this study (e.g., differences in TPR and FPR across racial groups) and due to its relevance to the COMPAS dataset, where disparities in error rates across racial groups have been a central focus in prior research (Angwin et al., 2016).

Regularisation is included as an in-processing technique due to its fundamental role in controlling model complexity and preventing overfitting (Hastie et al., 2009). By adding penalty terms to the loss function or constraining model complexity during training, regularization encourages more generalised predictions. In the context of this study, regularisation is applied as a form of "unaware" mitigation. While its primary purpose is not explicit bias mitigation, it can implicitly reduce reliance on spurious correlations that might lead to discriminatory patterns by promoting simpler, more generalised models.

In practice, this study uses the `LogisticRegression` class from scikit-learn, which includes L2 regularisation (also known as ridge regularisation) by default. This is implemented via the `penalty='l2' parameter`, with a default regularisation strength of `C=1.0` (inverse of regularisation strength). These defaults mean that regularisation is always present unless explicitly disabled (which was not done in this study). Thus, all LR models used in this analysis include L2 regularisation, even if it is not explicitly specified in the code (Buitinck et al., 2013).

The default L2 regularisation present in the *LogisticRegression* class was not disabled because it would conflict with established best practices in machine learning, where some form of regularisation is generally considered essential to avoid overfitting, especially in datasets with multicollinearity or high-dimensional features (Hastie et al., 2009). In this context, regularisation represents a baseline form of bias mitigation that aligns with common deployment settings in real-world applications.

This default regularisation applies only to the Logistic Regression models, not to the Random Forest models used in this study. Random Forests achieve generalisation through different mechanisms such as ensembling, bagging, and limiting tree depth, but they do not include regularisation in the same sense as linear models like Logistic Regression.

3.4.3 Post-processing techniques

Post-processing techniques adjust the predictions of an already trained model, making them suitable when access to the training process is limited or model alteration is constrained (Hardt et al., 2016). This study applies threshold optimiser and calibration as post-processing methods. These methods were fitted on the training set and applied to the test set during evaluation to ensure an unbiased final assessment.

Threshold optimiser is employed to adjust classification thresholds, potentially on a group-specific basis, to align with predefined fairness criteria on a trained model's probability outputs (Hardt et al., 2016). This method is valuable for its direct approach to manipulating prediction outcomes to achieve specific fairness objectives, offering a flexible way to mitigate bias without retraining the original model. Its ability to achieve precise fairness targets makes it crucial for understanding the final adjustments needed after a model has been built, particularly when sensitive attributes are not available during training.

Like EGR, equalised odds was selected as the fairness constraint for the threshold optimiser because it aligns with the group-level fairness metrics evaluated in this study and due to its relevance to documented disparities in COMPAS predictions (Angwin et al., 2016). The threshold optimiser's internal optimisation process can result in slight run-to-run differences. This variability is further examined in Section 3.6.

Calibration is included to ensure that a model's predicted probabilities accurately reflect the true likelihood of an outcome, thereby improving the reliability of its confidence scores (Zadrozny & Elkan, 2002). Similar to regularisation, calibration is applied as an "unaware" mitigation strategy in this study. While its primary goal is to improve the statistical soundness of probability outputs, well-calibrated models can sometimes indirectly reduce disparate impact by providing more accurate confidence estimates across all groups. Its simplicity and ability to enhance model trustworthiness make it a pertinent choice for assessing baseline post-training adjustments that might contribute to fairness without direct intervention for bias.

For calibration, two distinct methods were employed:

- Platt scaling (`method='sigmoid'`): This method fits a second LR model on the predicted scores of the base classifier to map them to calibrated probabilities. It assumes a monotonic, sigmoid-shaped distortion in the model's outputs (Platt, 1999).

- Isotonic regression (`method='isotonic'`): This method learns a piecewise constant, non-decreasing function to transform the predicted probabilities. Isotonic regression is more flexible than Platt scaling and can correct for more complex, non-linear, or even non-monotonic distortions in probability outputs. This flexibility often makes it particularly effective for calibrating ensemble models like RF, which can exhibit more irregular calibration curves (Zadrozny & Elkan, 2002).

While it would be possible to apply calibration-based methods to the full dataset, this study intentionally focuses on bias mitigation techniques that align with the availability of sensitive attributes. In the full dataset scenario, group-aware methods such as reweighting and EGR are included precisely because they can leverage protected attributes to directly reduce disparities. Conversely, the data-minimised scenario reflects legal or practical constraints under which only methods that do not require sensitive attributes during model training, such as calibration and the threshold optimiser, remain applicable. This design choice allows for a focused comparison of what fairness gains are achievable under each set of data availability constraints, rather than a comprehensive benchmarking of all methods across both datasets.

3.5 Evaluation metrics

To rigorously assess the effectiveness of various bias mitigation techniques under data minimisation constraints, this study evaluates models across two primary dimensions: predictive performance (see Section 3.5.1) and algorithmic fairness (see Section 3.5.2). This dual assessment acknowledges that effective machine learning solutions must be both accurate and equitable, particularly in high-stakes contexts where unequal treatment can lead to significant harm (Holmberg et al., 2020). As fairness lacks a single universally accepted definition and its metrics can be mathematically incompatible (Mehrabi et al., 2019; Binns, 2018), a comprehensive suite of commonly used metrics is employed.

3.5.1 Performance metrics

Standard accuracy metrics are used to evaluate the overall predictive capabilities and generalisation of the models. These metrics are essential for understanding the trade-offs between fairness interventions and overall model utility. The following metrics will be calculated:

- Accuracy: This fundamental metric measures the proportion of correctly classified instances (both recidivist and non-recidivist), providing a basic and easily understandable assessment of overall model correctness (Caruana & Niculescu-Mizil, 2006).

- ROC AUC: This metric assesses the model's ability to distinguish between the two classes (recidivist/non-recidivist) across all possible classification thresholds. It is particularly robust in the presence of class imbalance, indicating how well the model ranks positive instances higher than negative ones (Fawcett, 2006).

- F1 score: As the harmonic mean of precision and recall, the F1 score provides a single, balanced metric that accounts for both false positives and false negatives, which is crucial in imbalanced classification problems such as recidivism prediction (Manning et al., 2008).

- Classification report: This summary provides detailed precision, recall, and F1 scores for each individual class (recidivist and non-recidivist), along with overall averages. This allows for a quick diagnosis of class-specific performance issues (Pedregosa et al., 2011).

To serve as a practical benchmark for the trade-offs between fairness and performance, a 5% accuracy drop-off from a model's baseline performance will be considered the maximum acceptable tolerance for a mitigation method. This rule will be used in later chapters to determine whether a mitigation method successfully balances fairness improvements with an acceptable impact on accuracy.

3.5.2 Fairness metrics

Given that bias in machine learning can lead to unfair or non-generalisable outcomes through systematic influences from algorithmic design, data characteristics, or human decisions (Holmberg et al., 2020), fairness metrics are critical for assessing group-level disparities. Fairness is broadly understood as the absence of prejudice or favouritism toward an individual or group based on their characteristics (Mehrabi et al., 2019). In all experiments, fairness is evaluated on the respective test set used for model assessment, with the sensitive attribute, `race`, retained only for post-hoc fairness evaluation in the data-minimised scenario. This study employs the following group-level fairness metrics, disaggregated by `race` to uncover and quantify potential biases:

- Selection Rate (SR): This metric calculates the proportion of individuals within each demographic group predicted to be high-risk (recidivist). It directly addresses demographic parity, revealing if the rate of being selected for the adverse outcome (high-risk label) is disproportionate across groups, serving as a common starting point for bias discussions (Hardt et al., 2016). The SR is computed during the evaluation stage, after the model has been trained and has generated predictions on the held-out test.

- True Positive Rate (TPR) by group (equal opportunity): This metric measures, for each group, the proportion of actual recidivists who were correctly identified as such. Evaluating TPR across groups addresses the equal opportunity criterion, ensuring the model is equally effective at identifying genuine high-risk individuals across all populations, preventing certain groups from being overlooked for intervention (Hardt et al., 2016).

- False Positive Rate (FPR) by group: This metric assesses, for each group, the proportion of actual non-recidivists who were incorrectly predicted to be high-risk. In the context of recidivism prediction, a high FPR for a particular group signifies that individuals from that group are disproportionately subjected to "false alarms," potentially leading to unjust consequences despite not actually re-offending (Hardt et al., 2016).

- Equalised Odds Difference (EOD): This metric captures the maximum difference across all groups for both true positive and false positive rates. It aims for both true positive and false positive rates to be roughly equal across groups, representing a stronger fairness criterion and a key measure in assessing models like COMPAS (Hardt et al., 2016).

- Equalised Odds Difference excluding Native American and Asian (EOD-NAA): Due to small sample sizes of the Native American and Asian subgroups (see Section 3.2), the TPR and FPR for these groups can be highly volatile. This can lead to a skewed overall EOD score that may not accurately represent the model's fairness for larger, more well-represented groups. Therefore, a modified metric is also calculated that measures EOD across all groups except Native American and Asian, providing a more stable comparison of fairness.

- Group-wise ROC AUC scores: Calculating the ROC AUC for each individual group reveals whether the model's discriminatory power (its ability to correctly rank individuals by risk) is consistent or varies significantly across different groups. This helps determine if the model is operating with comparable effectiveness for all populations (Fong et al., 2022).

3.6 Reproducibility and variability

Fixed `random_state` values were used in all experiments to improve reproducibility and minimise variability across runs. This ensures deterministic data splitting and base model training. However, the threshold optimiser's internal optimisation process can still introduce slight run-to-run variability, as it does not provide a parameter to fix its own random state. Across repeated runs on the same

machine, accuracy varied by less than 0.01, while group-specific SR, TPR and FPR also varied by less than 0.01 for most racial groups. The only exception was for Asian and Native American subgroups, where small sample sizes meant that minor threshold adjustments could cause large swings in SR, TPR, and FPR (e.g., 0.00, 0.50, or 1.00), which in turn led to greater variability in EOD. ROC AUC remained identical, as it is calculated from the unchanged probability outputs of the base model (Fawcett, 2006).

It should be noted that while results were consistent across repeated runs on the same machine, small differences may still occur when running the code on different hardware or software environments. Research has shown that such run-to-run variability can arise from the inherent nature of parallel computation and its handling of non-associative floating-point arithmetic (Chou et al., 2020). Even when using identical algorithms and parameters, slight changes in the order of operations on different CPUs or GPUs can introduce minor numerical discrepancies. This hardware-level variability is a recognised challenge to achieving exact reproducibility in scientific computing (Chou et al., 2020; Zhuang et al., 2021).

3.7 Conceptual framework

The conceptual framework illustrated in Figure 5 visualises the empirical structure of this study. The solid arrows represent the direction of influence and data flow between elements, as well as the sequence of steps in the study's design. It begins with the data source, which may contain embedded bias, and proceeds through a process phase in which bias mitigation techniques are applied. These techniques function as mediating interventions intended to improve both fairness and performance outcomes. However, their effectiveness is shaped by two moderating variables: 1) data minimisation constraints, which may restrict access to sensitive attributes and limit the applicability of certain mitigation methods, and 2) the type of classifier, as different algorithms vary in how they respond to fairness interventions. The framework results in two primary outcome variables: fairness, evaluated through group-level metrics such as demographic parity and equalised odds; and performance, assessed using standard metrics such as accuracy, F1 score, and ROC AUC. This framework provides a structured overview of the study design and guides the empirical analysis that follows.

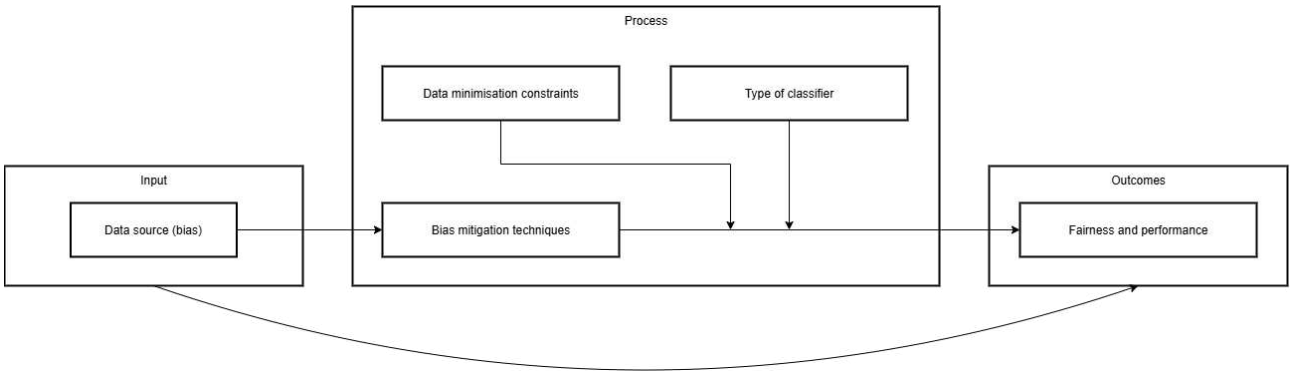

Figure 5. Conceptual framework illustrating study design, influencing factors, and outcomes

The framework outlines the study design, beginning with a potentially biased data source, applying bias mitigation techniques, and assessing fairness and performance outcomes. Effectiveness is influenced by data minimisation constraints and classifier type.

3.8 Summary

This chapter has outlined the methodological approach used to investigate the impact of bias mitigation techniques under data minimisation constraints. Guided by the CRISP-DM framework, the study follows a structured process that includes dataset preparation, model development, fairness intervention, and performance evaluation. The COMPAS dataset was selected for its real-world relevance and inclusion of sensitive features, enabling the creation of both full and data-minimised versions. Two classifiers, Logistic Regression and Random Forest, were chosen to reflect distinct modelling strategies. A combination of pre-processing, in-processing, and post-processing bias mitigation techniques was implemented. To assess model outcomes, a comprehensive set of performance and fairness metrics was used, including accuracy, F1 score, ROC AUC, and group-level fairness indicators such as Selection Rate and equalised odds. These methodological choices provide the foundation for a robust analysis of whether algorithmic fairness can be maintained when sensitive attributes are excluded from model training.

4 Empirical results: Full dataset

This chapter presents the empirical results obtained from training and evaluating ML models on the full version of the COMPAS dataset, which includes the sensitive attributes `race`. The purpose of this phase is to establish a baseline for model performance and fairness before applying data minimisation constraints. Two classifiers are used in this stage: Logistic Regression and Random Forest. These models are trained using the complete set of input features.

The chapter begins by reporting predictive performance using metrics such as accuracy, F1 score, and ROC AUC. It then evaluates algorithmic fairness with group-level metrics, including SR, TPR, FPR, EOD, EOD-NAA, and group-wise ROC AUC scores. These results provide a foundation for analysing the impact of bias mitigation techniques applied later in the chapter. They also serve as a benchmark for comparison with the data-minimised experiments presented in the following chapter.

4.1 Baseline model outcomes

4.1.1 Logistic Regression

The baseline LR model demonstrated solid predictive performance, achieving an accuracy of 0.684, ROC AUC of 0.730, and an F1 score of 0.623 (see Table 3). The detailed classification report is available in Table 4. However, significant fairness disparities were evident in the metrics. As shown in Table 6, the EOD was 0.696, with an SR disparity of 0.531 and an FPR disparity of 0.344. The EOD-NAA, which excludes Native American and Asian subgroups due to their small sample sizes, was significantly lower at 0.487. A group-wise breakdown (see Table 5) shows that African-American individuals experienced the highest FPR (0.344), while other groups had much lower values. The group-wise ROC AUC scores were relatively consistent, ranging from 0.702 to 0.77, though the score for Asian individuals was a notable outlier at 0.6, likely reflecting data imbalance. The ROC curve for this model is presented in Figure 6.

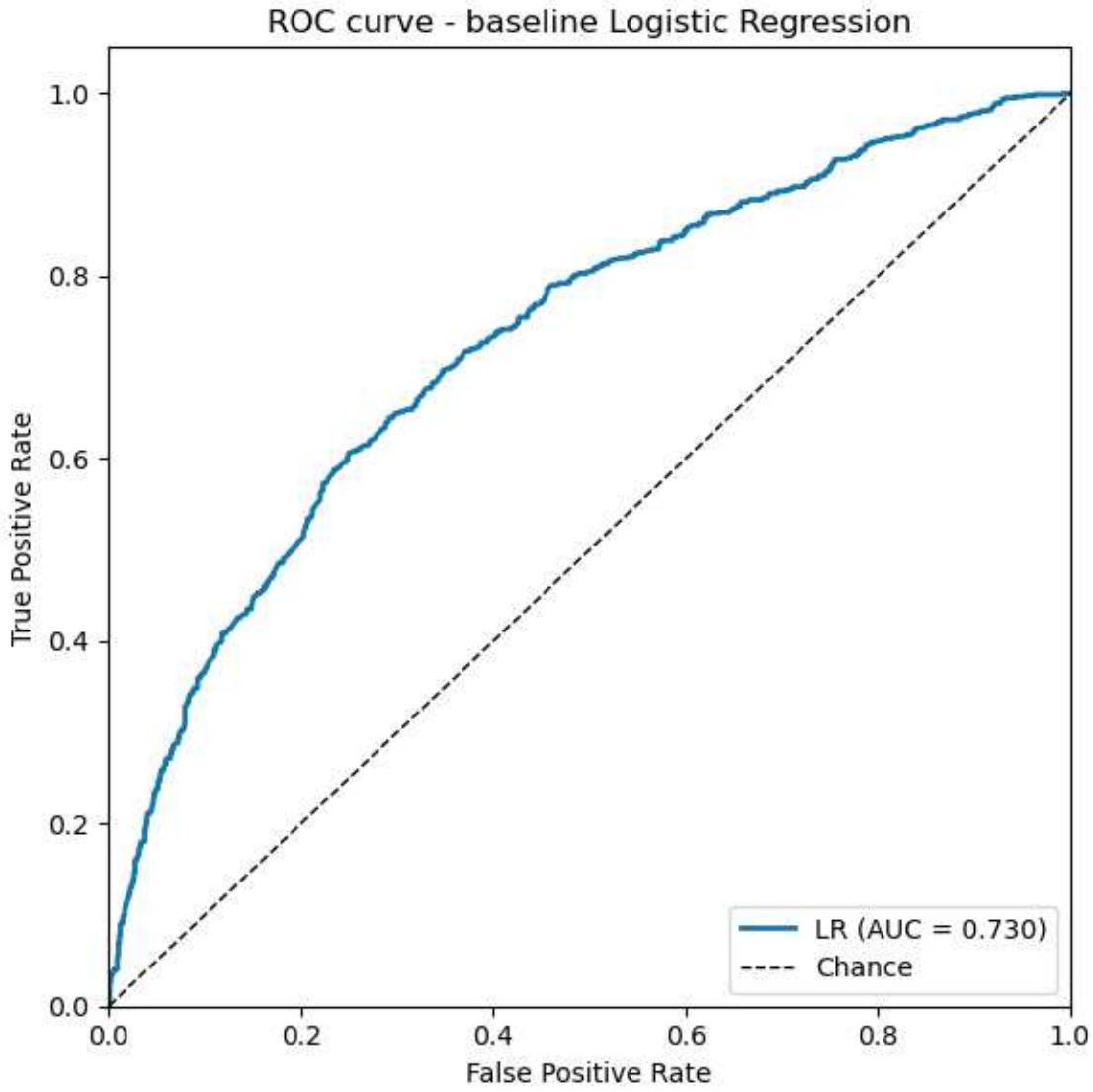

Figure 6. ROC curve for baseline LR model (full dataset)

Table 3. Overall performance metrics for the baseline LR (full dataset)

Metric	Baseline
Accuracy	0.684
ROC AUC	0.730
F1 score	0.623

Table 4. Classification report for baseline LR (full dataset)

Class	Precision	Recall	F1 score	Support
0 (Non-recidivist)	0.69	0.78	0.73	1009
1 (Recidivist)	0.68	0.57	0.62	843
Macro avg	0.68	0.67	0.68	1852

Weighted avg	0.68	0.68	0.68	1852

Table 5. Group-wise fairness metrics for baseline LR (full dataset)

Race	SR	TPR	FPR	ROC AUC
African-American	0.531	0.696	0.344	0.725
Caucasian	0.258	0.442	0.148	0.702
Hispanic	0.168	0.283	0.104	0.721
Other	0.109	0.209	0.045	0.77
Asian	0	0	0	0.6
Native American	0.25	0.5	0	0.75

Table 6. Fairness disparities for baseline LR (full dataset)

Metric	Baseline
SR disparity	0.531
TPR disparity	0.696
FPR disparity	0.344
EOD	0.696
EOD-NAA	0.487

Figure 7 visualises the predicted score distributions by race, illustrating the fairness disparities identified in the metrics above. *Predicted scores* refer to the probabilities assigned by the LR model that an individual will reoffend within two years. The curves are *kernel density estimates* (KDE), which smooth the distribution of predicted scores for each racial group to make patterns easier to compare.

The plot shows that African-American individuals tend to receive higher predicted recidivism probabilities, shifting their distribution towards the right side of the scale. The spike for the Asian group is likely a consequence of the small sample size, rather than a systematic difference in the model's behaviour. The Native American subgroup is absent from the plot because it contained fewer than five samples in the test set, which would have produced a statistically unreliable and potentially misleading KDE curve. This separation of the prediction scores is a key indicator of how the LR model is treating different groups distinctly, visualising the disparities reported earlier, such as the large EOD (0.696) and SR disparity (0.531).

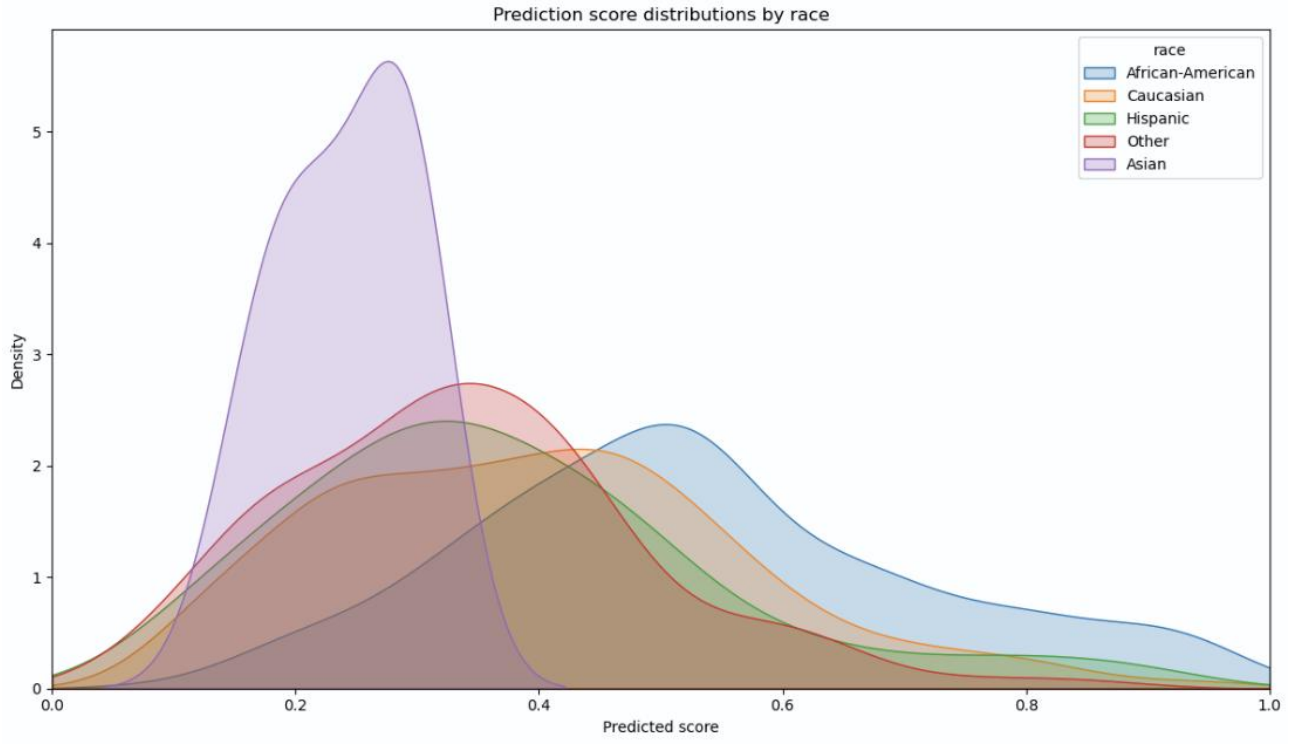

Figure 7. Predicted recidivism score distributions by race for baseline LR (full dataset)

To assess the statistical significance of the chosen features, I fitted another LR model using the Statsmodels library instead of scikit-learn. Statsmodels is designed for statistical inference and provides outputs such as standard errors, p-values, and confidence intervals, which are not available in scikit-learn's implementation, as it is primarily focused on prediction.

A Wald test was used to evaluate race as a group of predictors, with "Other" set as the reference category. The resulting p-value (0.120) indicates that *race*, considered as a whole, did not meet the conventional threshold for statistical significance ($p < 0.05$). In contrast, *age* ($p < .001$), *priors_count* ($p < .001$), *sex_Male* ($p < .001$), and *c_charge degree_M* ($p = .003$) were all found to be statistically significant predictors of two-year recidivism. The coefficient estimates, p-values, and cross-model comparisons are shown in Table 7.

To confirm that these findings are representative of the model used in the bias mitigation experiments, the same training split and features were fit using scikit-learn's LR model. The coefficients from this model were almost identical to those from the Statsmodels version, indicating that the p-value results from the latter are relevant and can be considered applicable.

Table 7. Statistical significance and coefficient comparison for the LR model (full dataset)

Feature	Coef (Statsmodels)	p-value	Coef (scikit-learn)	Difference

age	-0.0446	<0.001	-0.0446	0.0000
priors_count	0.1650	<0.001	0.1651	0.0001
sex_Male	0.3678	<0.001	0.3626	-0.0051
c_charge_degree_M	–-.2083	0.003	-0.2081	0.0002
race (Wald test)		0.120		

The complete Python code for training, evaluating, and analysing the baseline LR model on the full dataset is provided in Appendix 9.2.

4.1.2 Random Forest

The baseline RF model produced an accuracy of 0.623, ROC AUC of 0.668, and F1 score of 0.583 (see Table 8). The full classification report is available in Table 9. Despite adequate overall performance, fairness outcomes remained problematic. As detailed in Table 11, EOD was 0.679, but the EOD-NAA was significantly lower at 0.344, highlighting the impact of small subgroups. SR disparity was 0.322, and FPR disparity was 0.313. Group-wise metrics (see Table 10) show that African-American individuals had the highest FPR (0.433). There was wide variation in group-wise TPR and AUC scores, with the Asian subgroup showing unusually high values (TPR = 1, AUC = 0.9), indicating potential overfitting, while the Native American subgroup had an AUC of 0.5, suggesting a lack of meaningful discrimination. The ROC curve for this model is presented in Figure 8.

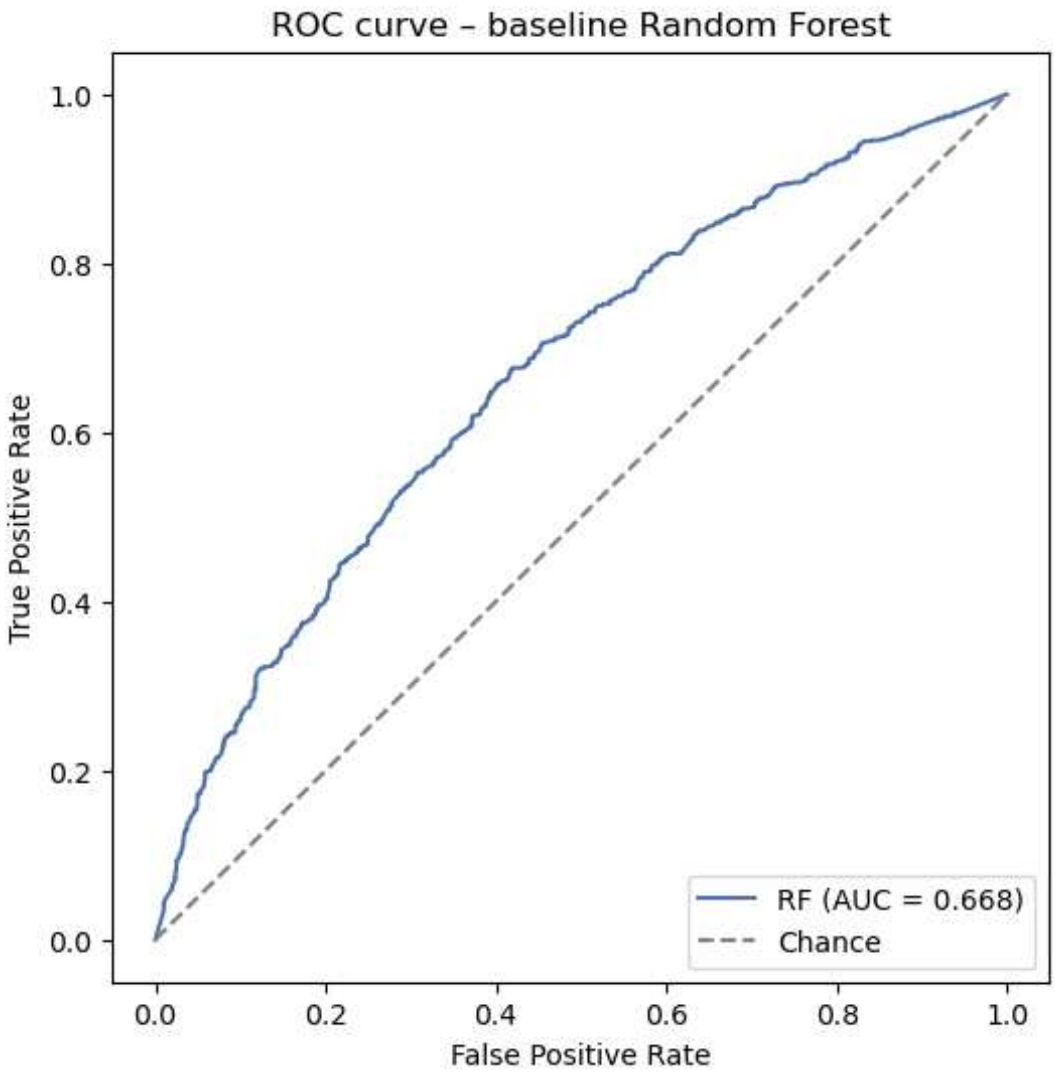

Figure 8. ROC curve for baseline RF model (full dataset)

Table 8. Overall performance metrics for baseline RF (full dataset)

Metric	Baseline
Accuracy	0.623
ROC AUC	0.668
F1 score	0.583

Table 9. Classification report for baseline RF (full dataset)

Class	Precision	Recall	F1 score	Support
0 (Non-recidivist)	0.65	0.66	0.66	1009
1 (Recidivist)	0.59	0.58	0.58	843
Macro avg	0.62	0.62	0.62	1852
Weighted avg	0.62	0.62	0.62	1852

Table 10. Group-wise fairness metrics and ROC AUC for baseline RF (full dataset)

Race	SR	TPR	FPR	ROC AUC
African-American	0.556	0.665	0.433	0.657
Caucasian	0.356	0.468	0.289	0.642
Hispanic	0.235	0.321	0.188	0.613
Other	0.309	0.442	0.224	0.736
Asian	0.429	1	0.2	0.9
Native American	0.5	0.5	0.5	0.5

Table 11. Fairness disparities for baseline RF (full dataset)

Metric	Baseline
SR disparity	0.322
TPR disparity	0.679
FPR disparity	0.313
EOD	0.679
EOD-NAA	0.344

Figure 9 visualises the predicted score distributions by race, illustrating the fairness disparities identified in the metrics above. The curves appear to extend beyond the 0-1 range on the x-axis, which is a visual artifact of the kernel density estimation smoothing process. This occurs because the KDE technique places a smooth kernel over each data point. When these data points are clustered near the boundaries of 0 or 1, the smoothing tails of these kernels bleed outside the 0-1 range. It is important to note that this does not indicate that the model's predicted scores themselves fall outside this range. The Native American subgroup is absent from the plot because it contained fewer than five samples in the test set, which would have produced a statistically unreliable and potentially misleading KDE curve.

The plot shows a clear shift in the distributions. The distribution for African-American individuals is shifted to the right, indicating that the model assigns them higher recidivism probabilities compared to other groups. This is consistent with the fairness metrics, where African-American individuals have the highest selection rate (0.556) and false positive rate (0.433).

While the overall pattern of disparity is similar to the LR model, the distributions are less distinct and more overlapping. The RF model appears to have a more complex and multimodal distribution of scores. This separation of scores still points to how the RF model is treating different groups

distinctly, visualising the disparities reported earlier, such as the large EOD (0.679) and the wide TPR disparity of 0.679.

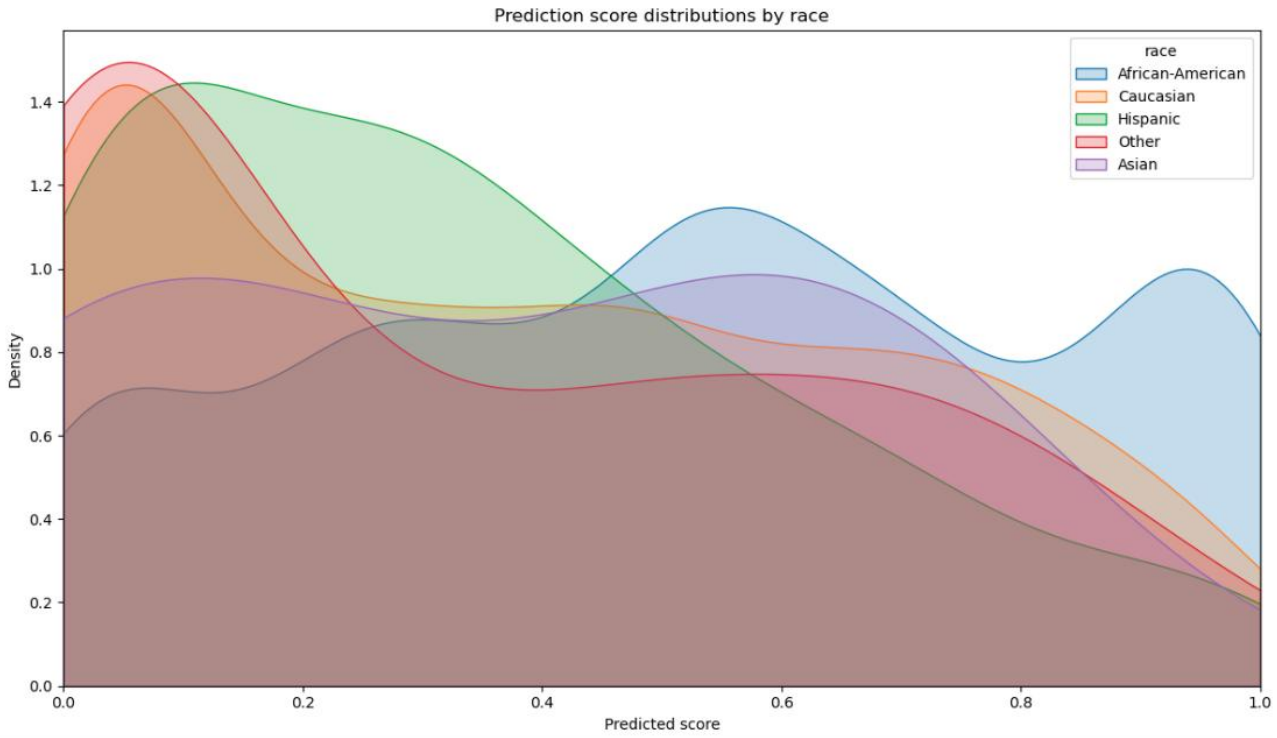

Figure 9. Predicted recidivism score distributions by race for baseline RF (full dataset)

The complete Python code for training, evaluating, and analysing the baseline RF model on the full dataset is provided in Appendix 9.3.

4.2 Effects of bias mitigation

4.2.1 Logistic Regression

Reweighting resulted in moderate fairness improvements, with EOD reduced from 0.696 to 0.581 and SR disparity from 0.531 to 0.385 (see Table 15). The EOD-NAA showed a more substantial improvement, dropping from 0.487 to a much lower value of 0.105. Predictive performance remained stable (see Table 12), with a slight decrease in accuracy to 0.668, ROC AUC to 0.718, and F1 score to 0.594. The detailed classification report and group-wise metrics are provided in Table 13 and Table 14, respectively. These results suggest that reweighting can improve group-level fairness with minimal impact on overall model performance. The complete Python code for training, evaluating, and analysing the LR model with reweighting on the full dataset is provided in Appendix 9.4.

Table 12. Overall performance metrics for LR with reweighting (full dataset)

Metric	Baseline	Reweighting	Δ (Delta)
Accuracy	0.684	0.668	−0.016
ROC AUC	0.730	0.718	−0.012
F1 score	0.623	0.594	−0.029

Table 13. Classification report for LR with reweighting (full dataset)

Class	Precision	Recall	F1 score	Support
0 (Non-recidivist)	0.67	0.78	0.72	1009
1 (Recidivist)	0.67	0.53	0.59	843
Macro avg	0.67	0.66	0.66	1852
Weighted avg	0.67	0.67	0.66	1852

Table 14. Group-wise fairness metrics and ROC AUC for LR with reweighting (full dataset)

Race	SR	TPR	FPR	ROC AUC
African-American	0.361	0.512	0.19	0.725
Caucasian	0.385	0.579	0.269	0.702
Hispanic	0.329	0.528	0.219	0.722
Other	0.327	0.581	0.164	0.77
Asian	0	0	0	0.6
Native American	0.25	0.5	0	0.75

Table 15. Fairness disparities for LR with reweighting (full dataset)

Metric	Baseline	Reweighting	Δ (Delta)
SR disparity	0.531	0.385	−0.146
TPR disparity	0.696	0.581	−0.115
FPR disparity	0.344	0.269	−0.075
EOD	0.696	0.581	−0.115
EOD-NAA	0.487	0.105	−0.382

EGR also provided notable fairness gains, as shown in Table 19. EOD was reduced from 0.696 to 0.547 and SR disparity from 0.531 to 0.387. Similarly, the EOD-NAA showed a significant drop to 0.121. Predictive performance metrics remained stable (see Table 16): accuracy decreased slightly to 0.663, ROC AUC to 0.721, and F1 score to 0.585. The classification report is available in Table 17. Group-wise ROC AUC scores were stable, ranging from 0.699 to 1.0 (see Table 18), indicating that

ranking performance was largely preserved. The complete Python code for training, evaluating, and analysing the LR model with EGR on the full dataset is provided in Appendix 9.5.

Table 16. Overall performance metrics for LR with EGR (full dataset)

Metric	Baseline	EGR	Δ (Delta)
Accuracy	0.684	0.663	−0.021
ROC AUC	0.730	0.721	−0.009
F1 score	0.623	0.585	−0.038

Table 17. Classification report for LR with EGR (full dataset)

Class	Precision	Recall	F1 score	Support
0 (Non-recidivist)	0.66	0.78	0.72	1009
1 (Recidivist)	0.66	0.52	0.58	843
Macro avg	0.66	0.65	0.65	1852
Weighted avg	0.66	0.66	0.66	1852

Table 18. Group-wise fairness metrics and ROC AUC for LR with EGR (full dataset)

Race	SR	TPR	FPR	ROC AUC
African-American	0.387	0.533	0.221	0.724
Caucasian	0.341	0.511	0.24	0.699
Hispanic	0.336	0.547	0.219	0.727
Other	0.245	0.442	0.119	0.764
Asian	0	0	0	0.8
Native American	0.25	0.5	0	1

Table 19. Fairness disparities for LR with EGR (full dataset)

Metric	Baseline	Reweighting	Δ (Delta)
SR disparity	0.531	0.387	−0.144
TPR disparity	0.696	0.547	−0.149
FPR disparity	0.344	0.24	−0.104
EOD	0.696	0.547	−0.149
EOD-NAA	0.487	0.121	−0.366

Threshold optimiser produced the most significant fairness improvements for LR. EOD was reduced from 0.696 to 0.500, and SR disparity from 0.531 to 0.285 (Table 23). Critically, the EOD-NAA showed the most substantial improvement among all methods, dropping to 0.077. This came at a

larger performance cost (Table 20), with accuracy declining to 0.655 and F1 score dropping to 0.516 due to lower recidivist recall. ROC AUC remained at 0.730 because threshold optimisation adjusts the decision boundary without altering the underlying probability estimates, meaning the model's ranking ability is unaffected. The classification report is in Table 21, and group-wise metrics in Table 22. The complete Python code for training, evaluating, and analysing the LR model with threshold optimiser on the full dataset is provided in Appendix 9.6.

Table 20. Overall performance metrics for LR with threshold optimiser (full dataset)

Metric	Baseline	Threshold optimiser	Δ (Delta)
Accuracy	0.684	0.655	−0.029
ROC AUC	0.730	0.730	0
F1 score	0.623	0.516	−0.107

Table 21. Classification report for LR with threshold optimiser (full dataset)

Class	Precision	Recall	F1 score	Support
0 (Non-recidivist)	0.63	0.87	0.73	1009
1 (Recidivist)	0.71	0.4	0.52	843
Macro avg	0.67	0.63	0.62	1852
Weighted avg	0.67	0.65	0.63	1852

Table 22. Group-wise fairness metrics and ROC AUC for LR with threshold optimiser (full dataset)

Race	SR	TPR	FPR	ROC AUC
African-American	0.285	0.41	0.143	0.725
Caucasian	0.234	0.391	0.141	0.702
Hispanic	0.255	0.415	0.167	0.721
Other	0.2	0.372	0.09	0.77
Asian	0	0	0	0.6
Native American	0.25	0.5	0	0.75

Table 23. Fairness disparities for LR with threshold optimiser (full dataset)

Metric	Baseline	Threshold optimiser	Δ (Delta)
SR disparity	0.531	0.285	−0.246
TPR disparity	0.696	0.5	−0.196
FPR disparity	0.344	0.167	−0.177
EOD	0.696	0.5	−0.196
EOD-NAA	0.487	0.077	−0.41

4.2.2 Random Forest

Reweighting slightly improved fairness for RF. EOD was reduced from 0.679 to 0.585, and SR disparity from 0.322 to 0.191 (see Table 27). The EOD-NAA also showed a strong improvement, dropping to 0.191. Performance remained nearly unchanged, with accuracy at 0.621, ROC AUC at 0.666, and F1 score at 0.576 (see Table 24). The classification report is provided in Table 25. Group-wise AUC variability and a high TPR for the Asian subgroup (1.0) persisted (see Table 26), indicating continued issues with group-level stability. The complete Python code for training, evaluating, and analysing the RF model with reweighting on the full dataset is provided in Appendix 9.7.

Table 24. Overall performance metrics for RF with reweighting (full dataset)

Metric	Baseline	Reweighting	Δ (Delta)
Accuracy	0.623	0.621	−0.002
ROC AUC	0.668	0.666	−0.002
F1 score	0.583	0.576	−0.007

Table 25. Classification report for RF with reweighting (full dataset)

Class	Precision	Recall	F1 score	Support
0 (Non-recidivist)	0.65	0.67	0.66	1009
1 (Recidivist)	0.59	0.57	0.58	843
Macro avg	0.62	0.62	0.62	1852
Weighted avg	0.62	0.62	0.62	1852

Table 26. Group-wise fairness metrics and ROC AUC for RF with reweighting (full dataset)

Race	SR	TPR	FPR	ROC AUC
African-American	0.495	0.606	0.368	0.658
Caucasian	0.399	0.524	0.325	0.644
Hispanic	0.309	0.415	0.25	0.603
Other	0.345	0.488	0.254	0.743
Asian	0.429	1	0.2	1
Native American	0.5	0.5	0.5	0.75

Table 27. Fairness disparities for RF with reweighting (full dataset)

Metric	Baseline	Reweighting	Δ (Delta)

SR disparity	0.322	0.191	−0.131
TPR disparity	0.679	0.585	−0.094
FPR disparity	0.313	0.300	−0.013
EOD	0.679	0.585	−0.094
EOD-NAA	0.344	0.191	−0.153

EGR modestly improved fairness for RF, with EOD decreasing from 0.679 to 0.604 (see Table 31). The EOD-NAA also showed an improvement, dropping to 0.212. Overall performance remained stable, with accuracy at 0.626 and ROC AUC at 0.665 (see Table 28). The classification report is in Table 29. However, large disparities across groups remained (see Table 30), including a disproportionately high AUC for Asian individuals (0.9) and a flat AUC for Native Americans (0.5), suggesting the mitigation did not fully address underlying imbalances. The complete Python code for training, evaluating, and analysing the RF model with EGR on the full dataset is provided in Appendix 9.8.

Table 28. Overall performance metrics for RF with EGR (full dataset)

Metric	Baseline	EGR	Δ (Delta)
Accuracy	0.623	0.626	+0.003
ROC AUC	0.668	0.665	−0.003
F1 score	0.583	0.578	−0.005

Table 29. Classification report for RF with EGR (full dataset)

Class	Precision	Recall	F1 score	Support
0 (Non-recidivist)	0.65	0.68	0.66	1009
1 (Recidivist)	0.59	0.56	0.58	843
Macro avg	0.62	0.62	0.62	1852
Weighted avg	0.62	0.63	0.62	1852

Table 30. Group-wise fairness metrics and ROC AUC for RF with EGR (full dataset)

Race	SR	TPR	FPR	ROC AUC
African-American	0.494	0.608	0.364	0.66
Caucasian	0.391	0.519	0.315	0.64
Hispanic	0.289	0.396	0.229	0.608
Other	0.309	0.442	0.224	0.728
Asian	0.429	1	0.2	0.9

Native American	0.25	0.5	0	0.75

Table 31. Fairness disparities for RF with EGR (full dataset)

Metric	Baseline	EGR	Δ (Delta)
SR disparity	0.322	0.244	−0.078
TPR disparity	0.679	0.604	−0.075
FPR disparity	0.313	0.364	+0.051
EOD	0.679	0.604	−0.075
EOD-NAA	0.344	0.212	−0.132

Threshold optimiser produced the strongest fairness improvements for RF among the tested methods. EOD fell from 0.679 to 0.581, and SR disparity from 0.322 to 0.271 (see Table 35). The EOD-NAA showed a substantial reduction to 0.156. Performance metrics were stable (see Table 32), with accuracy at 0.619 and ROC AUC at 0.668, which is expected since threshold optimisation does not alter the underlying probability rankings. The classification report and group-wise metrics are detailed in Table 33 and Table 34, respectively. Group-level inconsistencies persisted, however, with the AUC for Asian individuals remaining high (0.9) and the Native American subgroup low at 0.5. This suggests the mitigation did not fully resolve the underlying subgroup instability. The complete Python code for training, evaluating, and analysing the RF model with threshold optimiser on the full dataset is provided in Appendix 9.9.

Table 32. Overall performance metrics for RF with threshold optimiser (full dataset)

Metric	Baseline	Threshold optimiser	Δ (Delta)
Accuracy	0.623	0.619	−0.004
ROC AUC	0.668	0.668	0
F1 score	0.583	0.56	−0.023

Table 33. Classification report for RF with threshold optimiser (full dataset)

Class	Precision	Recall	F1 score	Support
0 (Non-recidivist)	0.64	0.69	0.66	1009
1 (Recidivist)	0.59	0.53	0.56	843
Macro avg	0.61	0.61	0.61	1852
Weighted avg	0.62	0.62	0.62	1852

Table 34. Group-wise fairness metrics and ROC AUC for RF with threshold optimiser (full dataset)

Race	SR	TPR	FPR	ROC AUC
African-American	0.461	0.575	0.333	0.657
Caucasian	0.375	0.485	0.309	0.642
Hispanic	0.322	0.434	0.26	0.613
Other	0.3	0.419	0.224	0.736
Asian	0.571	1	0.4	0.9
Native American	0.5	0.5	0.5	0.5

Table 35. Fairness disparities for RF with threshold optimiser (full dataset)

Metric	Baseline	Threshold optimiser	Δ (Delta)
SR disparity	0.322	0.271	−0.051
TPR disparity	0.679	0.581	−0.098
FPR disparity	0.313	0.276	−0.037
EOD	0.679	0.581	−0.098
EOD-NAA	0.273	0.156	−0.117

4.3 Comparative analysis of models

This section compares the performance and fairness of LR and RF classifiers, both at baseline and after applying bias mitigation techniques. The focus is on interpreting how each model behaves in terms of predictive accuracy and group-level fairness, particularly with respect to race.

Table 36. Baseline and post-mitigation results for LR and RF (full dataset)

Model	Mitigation Method	EOD	EOD-NAA	Accuracy	ROC AUC
LR	Baseline	0.696	0.487	0.684	0.730
LR	Reweighting	0.581	0.105	0.668	0.718
LR	EGR	0.547	0.121	0.663	0.721
LR	Threshold optimiser	0.500	0.077	0.655	0.730
RF	Baseline	0.679	0.344	0.623	0.668
RF	Reweighting	0.585	0.191	0.621	0.666
RF	EGR	0.604	0.212	0.626	0.665
RF	Threshold optimiser	0.581	0.156	0.619	0.668

4.3.1 Baseline models

The LR model consistently outperformed the RF model across all baseline performance metrics (see Table 36), achieving a higher accuracy (0.684 vs. 0.623), ROC AUC (0.730 vs. 0.668), and F1 score (0.623 vs. 0.583). Both models, however, exhibited substantial group-level disparities. The LR model had a high EOD of 0.696, which was largely influenced by the small subgroups. This is highlighted by its much lower EOD-NAA of 0.487. The RF model had a slightly lower EOD of 0.679, with an EOD-NAA of 0.344, but its subgroup behaviour was less stable. Notably, its group-wise AUC of 0.9 for Asian individuals and 0.5 for Native Americans pointed to overfitting and poor generalisability for small-sample-size subgroups.

4.3.2 Effects of bias mitigation

LR responded more favourably to bias mitigation. The threshold optimiser produced the most significant fairness improvements, reducing EOD from 0.696 to 0.500 and, most notably, EOD-NAA to a very low 0.077 (see Table 36). EGR also yielded strong fairness gains, reducing EOD to 0.547 and EOD-NAA to 0.121. Reweighting was also effective, lowering EOD to 0.581 and EOD-NAA to 0.105. All three methods achieved these improvements with only modest reductions in accuracy and F1 score.

In contrast, RF showed more moderate fairness gains. Its best EOD, achieved with both reweighting and the threshold optimiser, was 0.581 and 0.585, respectively. The EOD-NAA for these methods dropped to 0.156 and 0.191, which are good improvements but not as low as the LR model's results. None of the interventions could reduce SR disparity below 0.191. The subgroup AUC anomalies, particularly the inflated value of 0.9 for Asian individuals and the flat value of 0.5 for Native Americans, persisted across all mitigation methods (see Table 34), indicating that these techniques did not fully resolve the model's underlying instability with these subgroups.

4.3.3 Trade-offs and stability

Both models experienced trade-offs between fairness and predictive quality, but LR achieved a more favourable balance. Its fairness improvements were larger and more consistent, and its group-wise AUC scores remained relatively stable after mitigation. The RF model exhibited persistent volatility in subgroup metrics, limiting the effectiveness of all three bias mitigation methods. The threshold optimiser produced the strongest fairness improvements for RF, reducing EOD from 0.679 to 0.581 while maintaining baseline ROC AUC. However, subgroup-level inconsistencies persisted, limiting

the reliability of the fairness gains. These results suggest that LR's simpler structure is more compatible with fairness-aware learning, particularly in use cases where consistent treatment across groups is essential.

4.4 Summary

This section evaluated the performance and fairness of LR and RF models trained on the full dataset including race. LR consistently outperformed RF across performance metrics, achieving higher accuracy, ROC AUC, and F1 score at baseline. Both models, however, exhibited substantial group-level disparities, which led to the application of the following bias mitigation techniques: reweighting, EGR, and the threshold optimiser.

Among these methods, the threshold optimiser proved most effective at reducing EOD for both models, bringing LR's EOD down to 0.5 and RF's to 0.581. The EGR method was also highly effective, particularly for LR, reducing EOD to 0.547. Reweighting delivered the most modest fairness gains but with minimal performance cost. Examining EOD-NAA showed that all mitigation methods were highly effective at reducing disparities for the larger racial groups, with LR's EOD-NAA dropping to as low as 0.077.

The results indicate that LR demonstrated a more favourable response to all mitigation methods on this dataset. Its more consistent performance and stable subgroup behaviour suggest a better compatibility with fairness-aware techniques. RF, on the other hand, showed less consistent improvements. While its EOD was reduced, subgroup-level anomalies persisted, limiting the reliability of fairness gains. These findings will serve as a baseline for comparison with the data-minimised models discussed in the next chapter.

5 Empirical results: Data-minimised dataset

This chapter presents the empirical results from models trained on a data-minimised version of the COMPAS dataset, where the race feature was excluded as an input during training but retained for post hoc fairness evaluation. The primary goal is to assess how model performance and group-level disparities change when sensitive attributes are removed from the training data. As with the full-data analysis, Logistic Regression and Random Forest were tested. The bias mitigation methods explored in this stage include calibration (Platt scaling and isotonic regression) and the threshold optimiser. The results are presented in terms of performance and fairness metrics, which will enable a direct comparison with the full-data models discussed in the previous chapter.

5.1 Baseline model outcomes

5.1.1 Logistic Regression

The baseline LR model achieved solid predictive performance on the data-minimised dataset, with an accuracy of 0.678, an ROC AUC of 0.732, and an F1 score of 0.612, as shown in Table 37. The model's ROC curve is visualised in Figure 10. Despite the exclusion of race from training, fairness metrics revealed persistent disparities. The EOD reached 0.659, with significant gaps in selection and false positive rates (Table 40). However, the EOD-NAA was much lower at 0.251, indicating that the disparity was less pronounced when excluding the small subgroups. A detailed classification report is provided in Table 38. African-American individuals showed a high false positive rate of 0.319, while subgroups like “Other” had a much lower FPR of 0.075 (Table 39). The group-wise ROC AUC scores were relatively stable, ranging from 0.7 to 0.77.

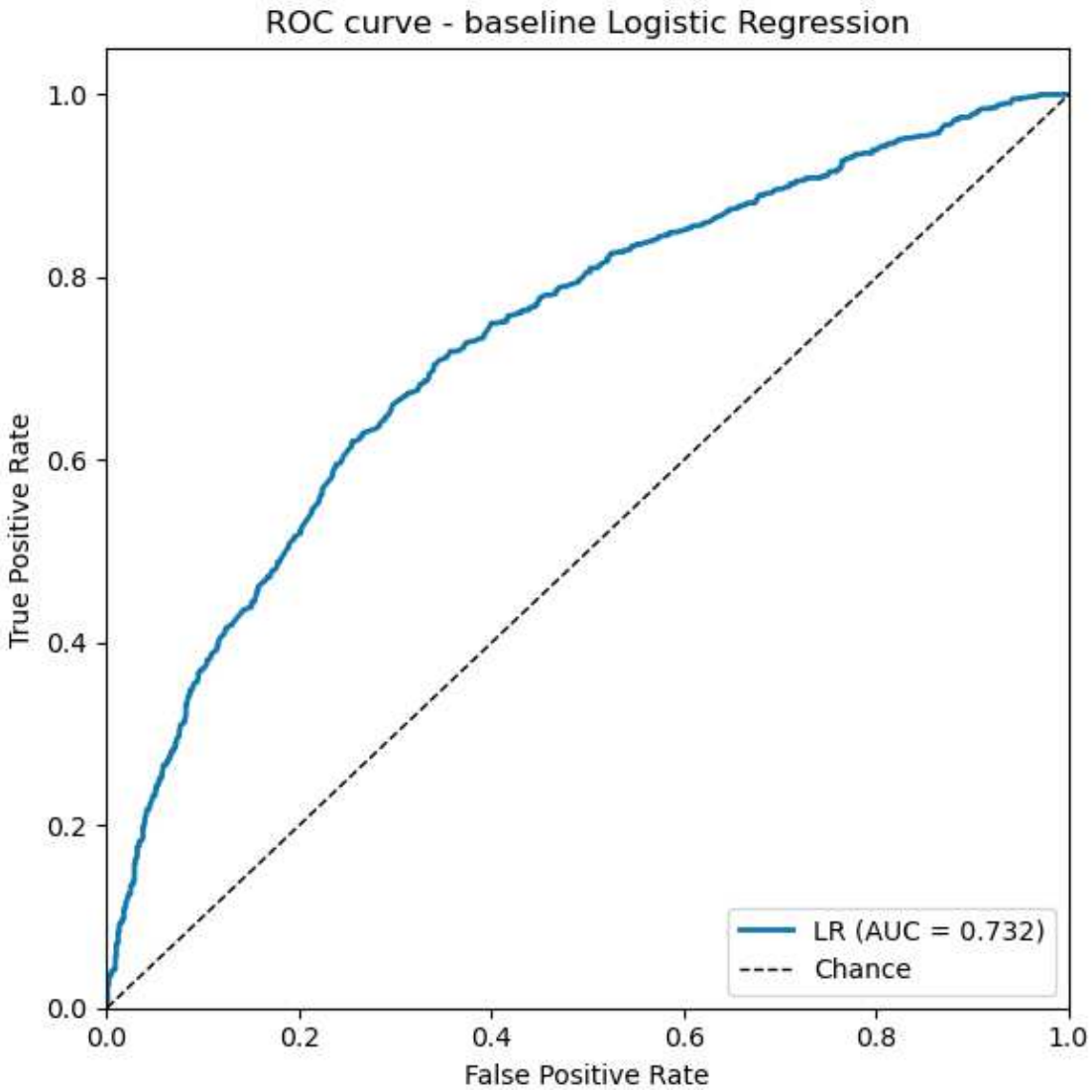

Figure 10. ROC curve for baseline LR model (data-minimised dataset)

Table 37. Overall performance metrics for baseline LR (data-minimised dataset)

Metric	Baseline
Accuracy	0.678
ROC AUC	0.732
F1 score	0.612

Table 38. Classification report for baseline LR (data-minimised dataset)

Class	Precision	Recall	F1 score	Support
0 (Non-recidivist)	0.68	0.78	0.73	1009
1 (Recidivist)	0.68	0.56	0.61	843
Macro avg	0.68	0.67	0.67	1852

Weighted avg	0.68	0.68	0.67	1852

Table 39. Group-wise fairness metrics and ROC AUC for baseline LR (data-minimised dataset)

Race	SR	TPR	FPR	ROC AUC
African-American	0.5	0.659	0.319	0.726
Caucasian	0.244	0.408	0.146	0.703
Hispanic	0.262	0.415	0.177	0.721
Other	0.5	0.659	0.319	0.77
Asian	0	0	0	0.7
Native American	0.5	0.5	0.5	0.75

Table 40. Fairness disparities for baseline LR (data-minimised dataset)

Metric	Baseline
SR disparity	0.5
TPR disparity	0.659
FPR disparity	0.5
EOD	0.659
EOD-NAA	0.251

Figure 11 visualises the predicted score distributions by race, illustrating the fairness disparities identified in the metrics above. Despite the removal of race from the training data, significant fairness disparities persist in the model's outcomes. The different shapes of the predicted score distributions is a visual representation of the high FPR disparity of 0.5 and the EOD of 0.659 reported in Table 40.

The distributions for African-American and “Other” individuals are still notably shifted to the right, indicating that the model continues to assign them higher recidivism probabilities. The Native American subgroup is absent from the plot because it contained fewer than five samples in the test set, which would have produced a statistically unreliable and potentially misleading KDE curve. While some groups, such as Caucasian and Hispanic individuals, show distributions that are more aligned than in the baseline model, the curves for African-American and “Other” individuals remain distinct. This confirms that the model is still treating groups differently.

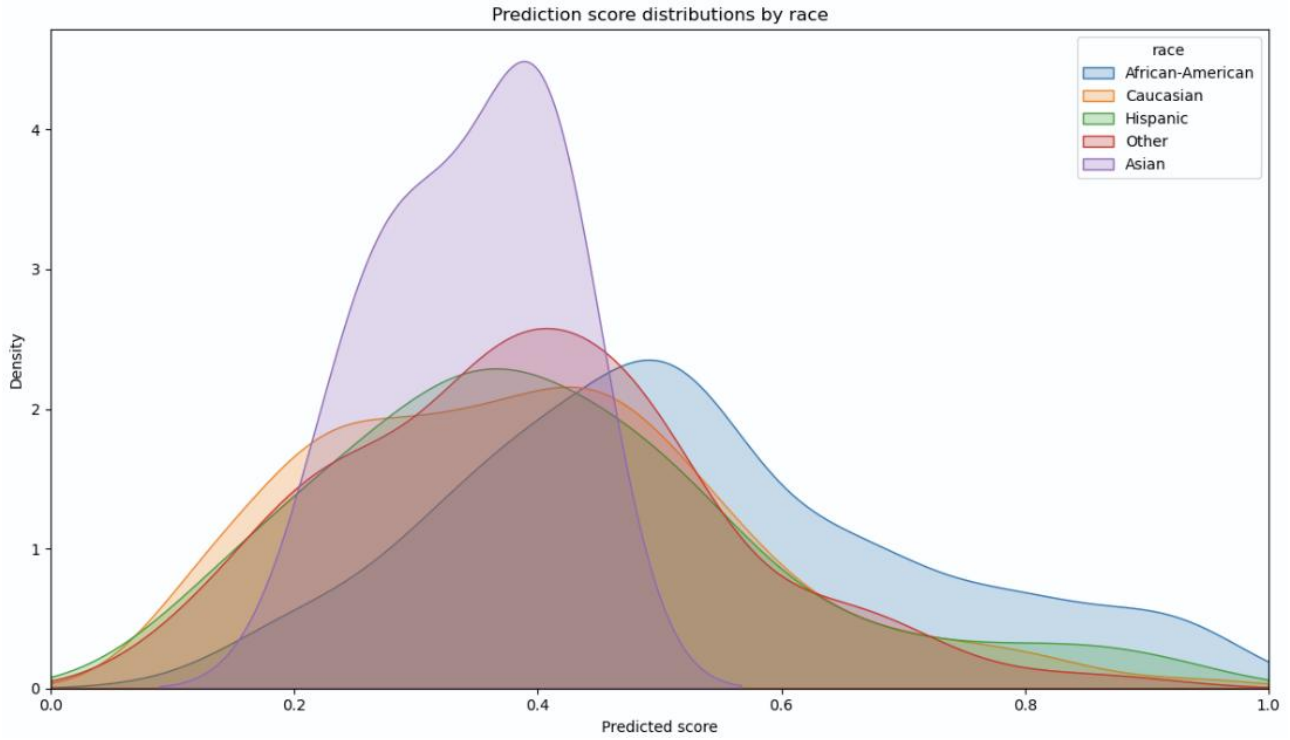

Figure 11. Predicted recidivism score distributions by race for baseline LR (data-minimised dataset)

The complete Python code for training, evaluating, and analysing the baseline LR model on the data-minimised dataset is provided in Appendix 9.10.

5.1.2 Random Forest

The baseline RF model on the data-minimised dataset yielded an accuracy of 0.652, an ROC AUC of 0.686, and an F1 score of 0.608 (Table 41). Its ROC curve is shown in Figure 12. The TPR disparity was lower than in the LR model (0.137), but the EOD still reached 0.500 due to a large FPR gap (Table 44), though the EOD-NAA significantly lower at 0.205. The classification report is in Table 42. The group-wise fairness metrics in Table 43 revealed continued challenges with subgroup stability: the AUC for Asian individuals remained inflated at 0.9, while Native American scores were flat at 0.75.

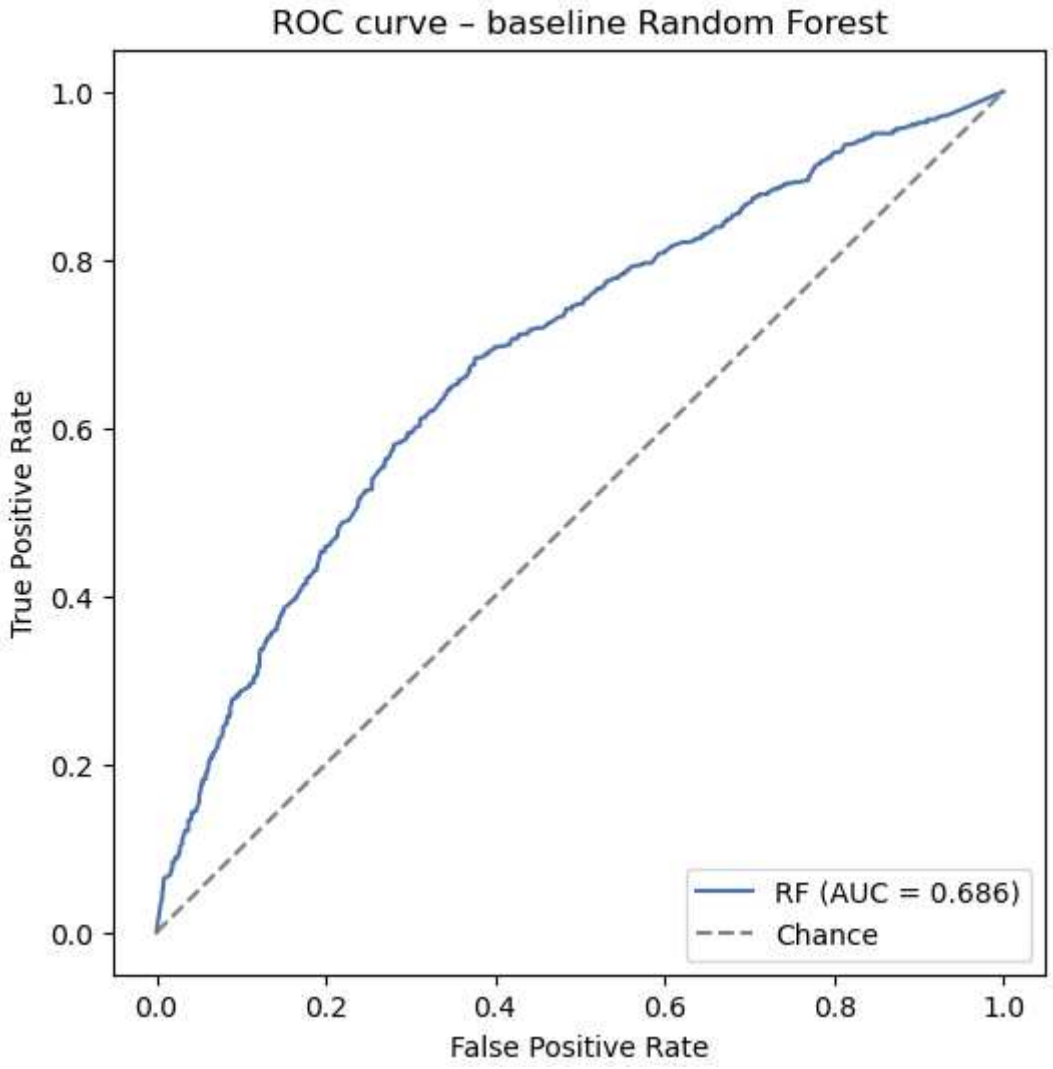

Figure 12. ROC curve for baseline RF model (data-minimised dataset)

Table 41. Overall performance metrics for baseline RF (data-minimised dataset)

Metric	Baseline
Accuracy	0.652
ROC AUC	0.686
F1 score	0.608

Table 42. Classification report for baseline RF (data-minimised dataset)

Class	Precision	Recall	F1 score	Support
0 (Non-recidivist)	0.67	0.7	0.69	1009
1 (Recidivist)	0.62	0.59	0.61	843
Macro avg	0.65	0.65	0.65	1852
Weighted avg	0.65	0.65	0.65	1852

Table 43. Group-wise fairnesss metrics and ROC AUC for baseline RF (data-minimised dataset)

Race	SR	TPR	FPR	ROC AUC
African-American	0.519	0.637	0.384	0.663
Caucasian	0.338	0.506	0.238	0.673
Hispanic	0.329	0.509	0.229	0.666
Other	0.355	0.628	0.179	0.778
Asian	0.143	0.5	0	0.9
Native American	0.5	0.5	0.5	0.75

Table 44. Fairness disparities for baseline RF (data-minimised dataset)

Metric	Baseline
SR disparity	0.376
TPR disparity	0.137
FPR disparity	0.5
EOD	0.5
EOD-NAA	0.205

Figure 13 visualises the predicted score distributions by race, illustrating the fairness disparities identified in the metrics above. The distributions for all racial groups are more similar and overlapping than in the baseline model, which suggests that removing the race feature had a mitigating effect on the model's tendency to produce disparate outcomes. This is reflected in the noticeably lower TPR disparity of the data-minimised model. The Native American subgroup is absent from the plot because it contained fewer than five samples in the test set, which would have produced a statistically unreliable and potentially misleading KDE curve.

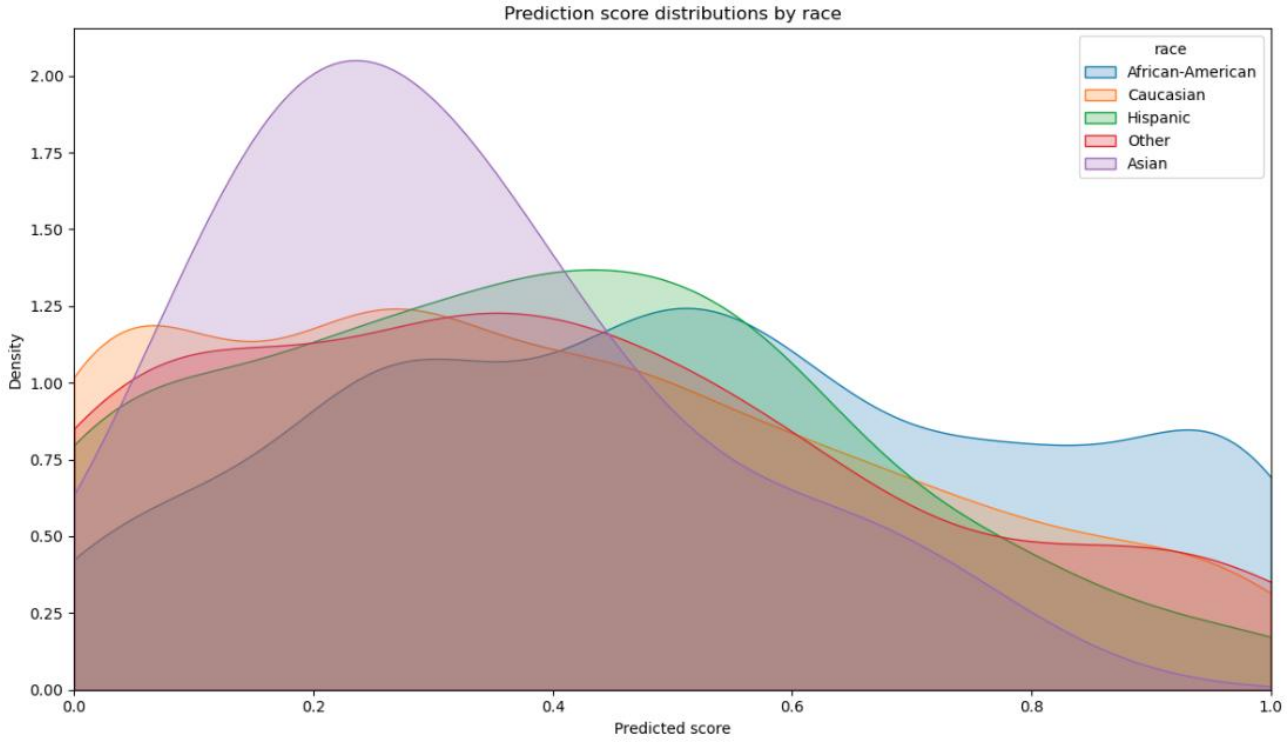

Figure 13. Predicted recidivism score distributions by race for baseline RF (data-minimised dataset)

The complete Python code for training, evaluating, and analysing the baseline RF model on the data-minimised dataset is provided in Appendix 9.11.

5.2 Effects of bias mitigation

5.2.1 Logistic Regression

To assess the effect of calibration, two methods were applied:

Platt scaling (`method='sigmoid'`) did not yield any observable changes in overall performance metrics or in the group-wise fairness metrics when compared to the uncalibrated baseline. Given that the results were identical to the baseline, separate presentation of these metrics is omitted for brevity and to avoid redundancy. This outcome is consistent with the LR model's inherent tendency to produce well-calibrated probabilities, where Platt scaling finds no significant systematic bias to correct (Hastie et al., 2009).

Isotonic regression (`method='isotonic'`) yielded the following results:

Performance was slightly improved, with accuracy increasing to 0.682 and the F1 score to 0.617 (Table 45). However, fairness metrics deteriorated. The EOD rose sharply from 0.659 to 1.000, which

was driven by extreme TPR variation, especially for Native American individuals (TPR = 1.0, as seen in Table 47). Although isotonic regression is not a classic bias mitigation method, it is often applied with the expectation that better probability calibration may lead to improved fairness. In this case, however, the calibration step introduced substantial group-level imbalances. A full breakdown of the classification report and fairness metrics is presented in Tables 46, 47, and 48. The complete Python code for training, evaluating, and analysing the LR model with calibration (Platt scaling and isotonic regression) on the data-minimised dataset is provided in Appendix 9.12.

Table 45. Overall performance metrics for LR with isotonic regression (data-minimised dataset)

Metric	Baseline	Isotonic regression	Δ (Delta)
Accuracy	0.678	0.682	+0.004
ROC AUC	0.732	0.732	0
F1 score	0.612	0.617	+0.005

Table 46. Classification report for LR with isotonic regression (data-minimised dataset)

Class	Precision	Recall	F1 score	Support
0 (Non-recidivist)	0.68	0.78	0.73	1009
1 (Recidivist)	0.68	0.56	0.62	843
Macro avg	0.68	0.67	0.67	1852
Weighted avg	0.68	0.68	0.68	1852

Table 47. Group-wise fairness metrics and ROC AUC for LR with isotonic regression (data-minimised dataset)

Race	SR	TPR	FPR	ROC AUC
African-American	0.498	0.659	0.315	0.726
Caucasian	0.245	0.416	0.143	0.701
Hispanic	0.255	0.415	0.167	0.719
Other	0.209	0.395	0.09	0.775
Asian	0	0	0	0.65
Native American	0.75	1	0.5	0.75

Table 48. Fairness disparities for LR with isotonic regression (data-minimised dataset)

Metric	Baseline	Isotonic regression	Δ (Delta)
SR disparity	0.5	0.75	+0.25
TPR disparity	0.659	1	+0.341
FPR disparity	0.5	0.5	0
EOD	0.659	1	+0.341
EOD-NAA	0.251	0.264	+0.013

Threshold optimiser reduced EOD from 0.659 to 0.416 and SR disparity from 0.500 to 0.147 (Table 52). This was achieved with a slight drop in accuracy to 0.651 and F1 score to 0.511 (Table 49), largely due to reduced recall for the recidivist class (Table 50). It also dramatically reduced the EOD-NAA to a very low 0.076. The ROC AUC remained stable at 0.732 as threshold optimisation does not alter the underlying probability rankings. The full breakdown of metrics is presented in Tables 50, 51, and 52. The complete Python code for training, evaluating, and analysing the LR model with threshold optimiser on the data-minimised dataset is provided in Appendix 9.13.

Table 49. Overall performance metrics for LR with threshold optimiser (data-minimised dataset)

Metric	Baseline	Threshold optimiser	Δ (Delta)
Accuracy	0.678	0.651	−0.027
ROC AUC	0.732	0.732	0
F1 score	0.612	0.511	−0.101

Table 50. Classification report for LR with threshold optimiser (data-minimised dataset)

Class	Precision	Recall	F1 score	Support
0 (Non-recidivist)	0.63	0.86	0.73	1009
1 (Recidivist)	0.70	0.40	0.51	843
Macro avg	0.67	0.63	0.62	1852
Weighted avg	0.66	0.65	0.63	1852

Table 51. Group-wise fairness metrics and ROC AUC for LR with threshold optimiser (data-minimised dataset)

Race	SR	TPR	FPR	ROC AUC
African-American	0.290	0.416	0.147	0.726
Caucasian	0.236	0.395	0.141	0.703
Hispanic	0.201	0.340	0.125	0.721
Other	0.209	0.372	0.105	0.770
Asian	0.143	0	0.2	0.700
Native American	0.250	0	0.5	0.750

Table 52. Fairness disparities for LR with threshold optimiser (data-minimised dataset)

Metric	Baseline	Threshold optimiser	Δ (Delta)
SR disparity	0.500	0.147	−0.353
TPR disparity	0.659	0.416	−0.243
FPR disparity	0.500	0.396	−0.104

EOD	0.659	0.416	−0.243
EOD-NAA	0.251	0.076	−0.175

5.2.2 Random Forest

To assess the effect of calibration, two methods were applied:

Platt scaling (`method='sigmoid'`) yielded the following results:

It had a minimal effect on the RF model's predictive metrics, as detailed in Table 53. Accuracy remained at 0.650, the F1 score increased slightly to 0.581, and the ROC AUC rose to 0.691. The EOD stayed constant at 0.500 (Table 56). However, disparities in FPR persisted, particularly for Native American individuals (0.5), and the inflated AUC for Asian individuals (0.9) was not corrected (Table 55). The full breakdown of metrics is presented in Tables 54, 55, and 56.

Table 53. Overall performance metrics for RF with Platt scaling (data-minimised dataset)

Metric	Baseline	Platt scaling	Δ (Delta)
Accuracy	0.652	0.65	−0.002
ROC AUC	0.686	0.691	+0.005
F1 score	0.608	0.581	−0.027

Table 54. Classification report for RF with Platt scaling (data-minimised dataset)

Class	Precision	Recall	F1 score	Support
0 (Non-recidivist)	0.66	0.75	0.70	1009
1 (Recidivist)	0.64	0.53	0.58	843
Macro avg	0.65	0.64	0.64	1852
Weighted avg	0.65	0.65	0.65	1852

Table 55. Group-wise fairness metrics and ROC AUC for RF with Platt scaling (data-minimised dataset)

Race	SR	TPR	FPR	ROC AUC
African-American	0.458	0.576	0.324	0.669
Caucasian	0.298	0.455	0.205	0.674
Hispanic	0.295	0.453	0.208	0.665
Other	0.282	0.535	0.119	0.776
Asian	0.143	0.500	0	0.900
Native American	0.500	0.500	0.500	0.750

Table 56. Fairness disparities for RF with Platt scaling (data-minimised dataset)

Metric	Baseline	Platt scaling	Δ (Delta)
SR disparity	0.376	0.357	−0.019
TPR disparity	0.137	0.124	−0.013
FPR disparity	0.500	0.500	0
EOD	0.500	0.500	0
EOD-NAA	0.205	0.205	0

Isotonic regression (`method='isotonic'`) yielded the following results:

It marginally reduced the F1 score to 0.587 and improved the ROC AUC to 0.692 (Table 57). However, it had no measurable impact on fairness: EOD remained at 0.5, and SR disparity at 0.357 (Table 60). Group-level anomalies, such as the high AUC for Asian individuals and flat scores for Native Americans, persisted (Table 59). The full breakdown of metrics is presented in Tables 58, 59, and 60. The complete Python code for training, evaluating, and analysing the RF model with calibration (Platt scaling and isotonic regression) on the data-minimised dataset is provided in Appendix 9.14.

Table 57. Overall performance metrics for RF with isotonic regression (data-minimised dataset)

Metric	Baseline	Isotonic regression	Δ (Delta)
Accuracy	0.652	0.651	−0.001
ROC AUC	0.686	0.692	+0.006
F1 score	0.608	0.587	−0.021

Table 58. Classification report for RF with isotonic regression (data-minimised dataset)

Class	Precision	Recall	F1 score	Support
0 (Non-recidivist)	0.66	0.74	0.70	1009
1 (Recidivist)	0.64	0.54	0.59	843
Macro avg	0.65	0.64	0.64	1852
Weighted avg	0.65	0.65	0.65	1852

Table 59. Group-wise fairness metrics and ROC AUC for RF with isotonic regression (data-minimised dataset)

Race	SR	TPR	FPR	ROC AUC
African-American	0.472	0.59	0.337	0.669
Caucasian	0.306	0.464	0.212	0.676
Hispanic	0.289	0.472	0.188	0.663

Other	0.291	0.535	0.134	0.775
Asian	0.143	0.5	0	1
Native American	0.5	0.5	0.5	0.75

Table 60. Fairness disparities for RF with isotonic regression (data-minimised dataset)

Metric	Baseline	Isotonic regression	Δ (Delta)
SR disparity	0.376	0.357	−0.019
TPR disparity	0.137	0.127	−0.010
FPR disparity	0.500	0.500	0
EOD	0.500	0.500	0
EOD-NAA	0.205	0.203	-0.002

Threshold optimiser reduced EOD from 0.5 to 0.317, while EOD-NAA decreased moderately from 0.205 to a low of 0.076. This was achieved with a slight drop in accuracy to 0.636 and F1 score to 0.570 (Table 61). While predictive performance decreased marginally, this method provided the largest fairness gains for RF in the data-minimised setting. The full breakdown of metrics is presented in Tables 62, 63, and 64. The complete Python code for training, evaluating, and analysing the RF model with threshold optimiser on the data-minimised dataset is provided in Appendix 9.15.

Table 61. Overall performance metrics for RF with threshold optimiser (data-minimised dataset)

Metric	Baseline	Threshold optimiser	Δ (Delta)
Accuracy	0.652	0.636	−0.016
ROC AUC	0.686	0.686	0
F1 score	0.608	0.57	−0.038

Table 62. Classification report for RF with threshold optimiser (data-minimised dataset)

Class	Precision	Recall	F1 score	Support
0 (Non-recidivist)	0.65	0.72	0.68	1009
1 (Recidivist)	0.62	0.53	0.57	843
Macro avg	0.63	0.63	0.63	1852
Weighted avg	0.63	0.64	0.63	1852

Table 63. Group-wise fairness metrics and ROC AUC for RF with threshold optimiser (data-minimised dataset)

Race	SR	TPR	FPR	ROC AUC
African-American	0.43	0.529	0.317	0.663

Caucasian	0.361	0.524	0.263	0.673
Hispanic	0.322	0.491	0.229	0.666
Other	0.373	0.651	0.194	0.778
Asian	0.143	0.5	0	0.9
Native American	0.25	0.5	0	0.75

Table 64. Fairness disparities for RF with threshold optimiser (data-minimised dataset)

Metric	Baseline	Threshold optimiser	Δ (Delta)
SR disparity	0.376	0.287	−0.089
TPR disparity	0.137	0.161	+0.024
FPR disparity	0.500	0.317	−0.183
EOD	0.500	0.317	−0.183
EOD-NAA	0.205	0.16	−0.045

5.3 Comparative analysis of models

This section compares the performance and fairness of Logistic Regression and Random Forest classifiers trained on the data-minimised version of the COMPAS dataset, where race was excluded from the training data but retained for post hoc evaluation. The analysis includes baseline results as well as the effects of fairness interventions, namely Platt scaling, isotonic regression, and the threshold optimiser. The focus is on how each model behaves in terms of predictive accuracy and group-level disparities. A summary of all baseline and post-mitigation results for both models is presented in Table 65.

Table 65. Baseline and post-mitigation results for LR and RF (data-minimised dataset)

Model	Mitigation method	EOD	EOD-NAA	Accuracy	ROC AUC
LR	Baseline	0.659	0.251	0.678	0.732
LR	Platt scaling	0.659	0.251	0.678	0.732
LR	Isotonic regression	1	0.264	0.682	0.732
LR	Threshold optimiser	0.416	0.076	0.651	0.732
RF	Baseline	0.5	0.205	0.652	0.686
RF	Platt scaling	0.5	0.205	0.65	0.691
RF	Isotonic regression	0.5	0.203	0.651	0.692
RF	Threshold optimiser	0.317	0.16	0.636	0.686

5.3.1 Baseline models

Both LR and RF achieved reasonable baseline accuracy despite the exclusion of `race` from the input features. LR showed slightly better performance overall, with an accuracy of 0.678 and ROC AUC of 0.732, compared to 0.652 and 0.686 for RF. However, fairness disparities persisted in both models. LR exhibited a high EOD of 0.659, while RF had a lower EOD of 0.500. When considering EOD-NAA, the models were closer: LR at 0.251 and RF at 0.205. While the RF model had a smaller TPR disparity, its subgroup performance remained unstable, especially with inflated AUC scores for Asian individuals and flat scores for Native Americans.

5.3.2 Effects of bias mitigation

LR displayed mixed results when fairness interventions were applied. Platt scaling produced no measurable change in metrics, which aligns with LR's tendency to produce calibrated outputs. Isotonic regression led to a slight improvement in accuracy (to 0.682), but fairness worsened substantially with EOD increasing to 1, due to extreme TPR imbalance for the subgroups with a small sample size. Additionally, it slightly worsened EOD-NAA, which increased to 0.264. This suggests that while calibration can improve overall probability accuracy, it can also unintentionally introduce new disparities in subgroups. The threshold optimiser, on the other hand, was highly effective at reducing bias. It dramatically improved EOD to 0.416 and EOD-NAA to a very low 0.076. The accuracy drop for this method was 4%, which is within the 5% tolerance defined in the methodology, making it a successful trade-off.

RF showed smaller but steadier changes in response to mitigation. Platt scaling and isotonic regression had minimal impact on fairness metrics, leaving EOD and EOD-NAA largely unchanged. The threshold optimiser proved to be the most effective intervention for RF in this setting, reducing EOD to 0.317 and EOD-NAA to 0.16. This was achieved with a modest accuracy drop of 2.5%, which is well within the acceptable 5% tolerance. However, the subgroup AUC anomalies, particularly the inflated value for Asian individuals (0.9) and the flat scores for Native Americans (0.5), persisted across all mitigation methods, indicating that these techniques did not fully resolve the model's underlying instability with these subgroups.

5.3.3 Trade-offs and stability

Both models faced the usual trade-offs between fairness improvements and performance. LR showed stronger responsiveness to fairness correction but was also more sensitive to instability, particularly

under isotonic calibration. RF maintained more stable behaviour across mitigation techniques, though with more limited fairness gains. The threshold optimiser struck the best balance for both models by significantly improving fairness metrics while keeping model performance within the 5% tolerance. These results suggest that even with data minimisation, post hoc bias mitigation remains a necessary step to achieve a more equitable outcome.

5.4 Summary

This section evaluated the performance and fairness of LR and RF models trained on a data-minimised version of the COMPAS dataset, where `race` was excluded during training but retained for evaluation. Despite this exclusion, both models continued to exhibit substantial group-level disparities, demonstrating that removing sensitive attributes does not eliminate bias. At baseline, LR showed higher disparity (EOD = 0.659) than RF (EOD = 0.5), but these values reflect only aggregate measures, and subgroup behaviour remained inconsistent in both models.

Among the tested mitigation methods, the threshold optimiser was the most effective for both models, reducing EOD to 0.416 for LR and 0.317 for RF. It also dramatically improved EOD-NAA to a very low 0.076 (for LR) and 0.16 (for RF). The performance cost for this method was within the acceptable 5% tolerance for both models. Calibration-based techniques produced limited or harmful results. Isotonic regression severely worsened fairness in LR, increasing EOD to 1 despite a slight accuracy improvement. Platt scaling had a negligible effect across both models, particularly for LR, which was already well-calibrated.

Overall, these findings illustrate that data minimisation alone is insufficient to ensure fairness. Group-level disparities persist even when sensitive attributes are excluded, underscoring the need for post hoc evaluation and targeted mitigation. Model behaviour varied by intervention, further reinforcing the importance of context-specific fairness assessments when designing equitable DSS.

6 Cross-dataset comparison

This chapter provides a comparative analysis of models trained on two versions of the COMPAS dataset: a full version that includes the sensitive attribute *race*, and a data-minimised version where *race* was excluded from the inputs. The primary goal is to evaluate the impact of data minimisation on both predictive performance and group-level fairness. A key aspect of this comparison is how the range of applicable bias mitigation techniques differs when sensitive attributes are not available during training, and what level of fairness and accuracy can be achieved under these constraints.

For the full dataset, mitigation methods included fairness-aware techniques like reweighting, EGR, and the threshold optimiser. These methods directly use the *race* attribute to reduce disparities. In the data-minimised condition, mitigation was limited to post hoc calibration methods (Platt scaling and isotonic regression) and the threshold optimiser, all of which operate without access to `race`. Table 66 summarises the results across both experimental conditions for the LR and RF classifiers.

Table 66. Baseline and post-mitigation results for LR and RF on both datasets

Model	Dataset	Mitigation Method	EOD	EOD-NAA	Accuracy	ROC AUC
LR	Full	Baseline	0.696	0.487	0.684	0.730
LR	Full	Reweighting	0.581	0.105	0.668	0.718
LR	Full	EGR	0.547	0.121	0.663	0.721
LR	Full	Threshold optimiser	0.5	0.077	0.655	0.730
LR	Full	Reweighting+EGR+Threshold optimiser	0.674	0.203	0.625	0.714
LR	Minim.	Baseline	0.659	0.251	0.678	0.732
LR	Minim.	Platt scaling	0.659	0.251	0.678	0.732
LR	Minim.	Isotonic regression	1	0.264	0.682	0.732
LR	Minim.	Threshold optimiser	0.416	0.076	0.651	0.732
RF	Full	Baseline	0.679	0.344	0.623	0.668
RF	Full	Reweighting	0.585	0.191	0.621	0.666
RF	Full	EGR	0.604	0.212	0.626	0.665
RF	Full	Threshold optimiser	0.581	0.156	0.619	0.668
RF	Full	Reweighting+EGR+Threshold optimiser	0.623	0.204	0.621	0.675
RF	Minim.	Baseline	0.5	0.205	0.652	0.686
RF	Minim.	Platt scaling	0.5	0.205	0.65	0.691
RF	Minim.	Isotonic regression	0.5	0.203	0.651	0.692
RF	Minim.	Threshold optimiser	0.317	0.16	0.636	0.686

6.1 Comparative performance outcomes

Predictive performance was notably stable across both datasets and mitigation methods, with the accuracy and ROC AUC scores showing minimal fluctuation. For the LR model, accuracy ranged from 0.625 to 0.684 on the full dataset and from 0.651 to 0.682 on the data-minimised dataset. To investigate if a synergistic effect could be achieved, all three fairness-aware methods were combined. However, this approach resulted in the lowest accuracy of 0.625 and did not improve fairness. Similarly, ROC AUC remained consistently around 0.730 for all LR interventions, suggesting that excluding race did not significantly impair the model's overall predictive power. For RF, accuracy was slightly lower but also stable, ranging from 0.619 to 0.626 on the full dataset and from 0.636 to 0.652 on the data-minimised dataset. The threshold optimiser caused an accuracy drop of approximately 4.0% for LR and 2.5% for RF on the data-minimised dataset, both of which were within the acceptable 5% tolerance.

Across both conditions, LR consistently outperformed RF in predictive performance, achieving higher accuracy and ROC AUC at baseline and after nearly every mitigation technique. This suggests that the LR model is generally more robust for this application. However, a closer look at the subgroup-level metrics revealed persistent instability. Despite the stable aggregate scores, True and False Positive Rates varied substantially between racial groups, particularly in the data-minimised setting. This highlights the limitations of using only aggregate metrics and reinforces the need for disaggregated fairness evaluation.

6.2 Comparative fairness outcomes

Fairness outcomes showed a much clearer distinction between the two datasets. In the full dataset, fairness-aware methods that leveraged the sensitive attribute were highly effective. The threshold optimiser on the LR model on the full dataset achieved a low EOD-NAA of 0.077 (down from a baseline of 0.487), demonstrating the power of a mitigation method that can use sensitive attributes to directly correct bias. The same method, when applied to the RF model on the data-minimised dataset, produced the lowest EOD of 0.317 (down from a baseline of 0.5), and achieved an even lower EOD-NAA of 0.076 with the LR model on the data-minimised dataset. This highlights the exceptional effectiveness of the threshold optimiser, regardless of whether sensitive attributes are available. EGR also performed well, with an EOD of 0.547 and an EOD-NAA of 0.121, also without a significant performance loss.

In the data-minimised dataset, mitigation was limited to post hoc methods. The threshold optimiser was the most effective intervention, dramatically reducing LR's EOD to 0.416 and achieving the lowest overall EOD-NAA of 0.076. This was achieved with a tolerable performance drop, making it a successful trade-off. Calibration methods were unreliable; while Platt scaling had no measurable effect, isotonic regression was actively harmful to fairness for LR, causing EOD to rise to a maximum of 1 and EOD-NAA to slightly worsen.

For the RF model, fairness improvements were less consistent. The threshold optimiser in the data-minimised setting produced a low EOD of 0.317 (and EOD-NAA of 0.16), which was a significant numerical improvement over its best EOD on the full dataset (0.581 with the threshold optimiser). However, this result for RF is undermined by persistent subgroup AUC anomalies, with the score for Asian individuals remaining inflated at 0.9 and for Native Americans staying flat at 0.5 across all interventions. These anomalies suggest that low EOD values for RF do not necessarily reflect genuine fairness and are likely a result of post hoc adjustments that failed to resolve the underlying instability. For both LR and RF, combining all mitigation methods on the full dataset actually resulted in a worse EOD and EOD-NAA than the best individual methods, suggesting that the combination of interventions is not always synergistic. The complete Python code for training and evaluating the LR and RF models with combined reweighting, EGR, and threshold optimiser on the full dataset is provided in Appendices 9.16 and 9.17.

6.3 Summary

This chapter compared the fairness and performance of models trained on a full versus a data-minimised version of the COMPAS dataset. The findings show that while predictive performance remained relatively stable in both conditions, data minimisation alone is insufficient to ensure fairness, as significant group-level disparities persisted. The analysis also revealed a crucial finding: the baseline models for the data-minimised setting already had lower EOD and EOD-NAA scores than their full-data counterparts. This suggests that excluding `race` from the training data had a positive initial effect, preventing the models from making biased associations from the outset. The best numerical fairness outcomes were ultimately achieved with the threshold optimiser on the data-minimised dataset: the RF model achieved the lowest EOD (0.317), while the LR model achieved the lowest overall EOD-NAA (0.076).

The effectiveness of the available post hoc methods in the data-minimised setting was mixed. The threshold optimiser emerged as the most viable tool, providing moderate fairness gains for LR (EOD reduced to 0.416 and EOD-NAA to a very low 0.076) without a significant performance cost.

Calibration methods, on the other hand, proved unreliable and in some cases were detrimental to fairness. Crucially, LR was consistently more responsive to fairness mitigation and exhibited more stable subgroup behaviour than RF, which remained volatile across interventions.

These results illustrate a complex trade-off: while data minimisation can provide a better starting point for fairness and, when paired with an effective method like the threshold optimiser, produce the best numerical scores, it severely constrains the range and reliability of bias mitigation strategies.

7 Discussion

This chapter interprets and contextualises the empirical results presented in the previous chapters. It highlights the most important findings, discusses their practical and theoretical implications, identifies key limitations of the study, and suggests directions for future research. The focus is on the consequences of data minimisation for fairness in ML and the effectiveness of mitigation strategies under different constraints.

7.1 Key findings

This study reveals a complex relationship between data minimisation, bias mitigation, and algorithmic fairness. A crucial finding is that the data-minimised baseline models already had lower EOD and EOD-NAA scores than their full-data counterparts. This suggests that excluding the sensitive attribute `race` from the training data had a positive initial effect, preventing the models from making certain biased associations from the outset.

The availability of sensitive attributes during model training still plays a central role in the full-data setting. When race was included as a feature, fairness-aware methods such as EGR and reweighting achieved substantial reductions in disparity. For example, applying EGR to the LR model on the full dataset reduced EOD from 0.696 to 0.547, with an EOD-NAA of 0.121, while maintaining strong predictive performance. These findings align with claims that fairness-enhancing techniques often rely on access to sensitive features to operate effectively (Hardt et al., 2016; Hort et al., 2024).

In the data-minimised setting, the choice of mitigation methods was restricted to post hoc techniques. The threshold optimiser proved to be the most effective intervention under these constraints, and its results demonstrate the best fairness outcomes of the entire study. For the LR model, it reduced EOD to 0.416 and EOD-NAA to an exceptionally low 0.076. For the RF model, it achieved the lowest overall EOD of 0.317. However, these numerical improvements for RF were undermined by persistent instability in subgroup behaviour, with inflated or flat ROC AUC scores for certain racial groups. This concern has also been raised by Mehrabi et al. (2019), who noted that algorithmic fairness cannot be guaranteed by aggregated metrics alone.

Calibration methods, such as Platt scaling and isotonic regression, had limited or negative effects. Platt scaling did not noticeably change fairness metrics, especially for LR, which was already well-calibrated. Isotonic regression slightly improved accuracy but severely disrupted fairness, increasing EOD to 1 in some cases due to distorted TPR across subgroups. This supports concerns about

technical bias and the risk that corrective strategies can unintentionally exacerbate disparities (Holmberg et al., 2020).

Overall, LR consistently outperformed RF in both predictive performance and responsiveness to fairness interventions. It delivered higher accuracy, better ROC AUC scores, and more stable subgroup behaviour across both datasets. This confirms earlier observations in the literature that interpretable models can offer greater reliability in regulated settings (Caraciolo, 2011).

The most important findings concern the impact of data minimisation on fairness outcomes. While removing `race` from model inputs had the positive effect of creating a fairer baseline, it did not eliminate discrimination. It also restricted the choice of mitigation methods and limited the overall reliability of the available toolkit. These results underline a fundamental trade-off: adhering to a data minimisation principle can lead to a better starting point and can achieve very good fairness results, but it restricts the toolkit to a few post-hoc methods, increasing reliance on a single intervention. For decision-making systems where fairness is critical, this trade-off must be carefully considered when designing data collection, model training, and mitigation strategies.

7.2 Implications

This study highlights a practical tension between legal compliance and algorithmic fairness in ML development. While adherence to data minimisation, a core principle of the GDPR (Article 5(1)(c); Ganesh et al., 2025), can lead to fairer baseline outcomes, it significantly restricts the available toolkit of interventions. This creates a reliance on a limited number of post-hoc methods, some of which, as seen with isotonic regression in this study, can be ineffective or even harmful. The trade-off, therefore, lies between strict compliance and the flexibility and robustness of the fairness-mitigation process.

From a business perspective, this creates a regulatory dilemma. Organisations are required to limit the use of personal data, including sensitive attributes such as race or gender, in accordance with data protection principles (Recital 39; Article 25). However, these attributes are often necessary for identifying and mitigating bias in predictive models (Tran & Fioretto, 2023; Mehrabi et al., 2019). When sensitive features are unavailable, mitigation options become limited, and performance depends heavily on a small set of post-hoc interventions. While post-hoc methods, such as the threshold optimiser, are often considered less effective than fairness-aware techniques that operate during model training (Hardt et al., 2016; Hort et al., 2024), this research provides a counter-narrative. In the data-minimised setting, the threshold optimiser achieved the best numerical fairness outcomes

of the entire study, reducing LR's EOD-NAA to 0.076. This suggests that, in certain contexts, post-hoc interventions can be as effective as, or even outperform, fairness-aware methods that rely on sensitive attributes.

At the same time, the full-dataset results demonstrate that access to sensitive attributes can enable more direct and potentially more robust fairness interventions, such as EGR and reweighting. This creates mixed evidence for the GDPR's "necessity" criterion under Article 9(2)(g). While sensitive data can clearly improve certain fairness metrics, strong fairness outcomes can also be achieved without it. These findings suggest that the necessity of processing sensitive attributes for fairness should be assessed on a case-by-case basis, considering both the fairness gains and the reliability of the available methods.

For policymakers, the results underscore a gap in current regulatory frameworks, which emphasise privacy and data minimisation but provide limited guidance on achieving fairness under these constraints (Ganesh et al., 2025). Clearer provisions or best practices could help organisations balance these objectives, for example by allowing limited and safeguarded access to sensitive attributes strictly for fairness evaluation (Recitals 26 and 78).

For researchers, the findings underline the methodological challenges of working without sensitive attributes. Most bias mitigation techniques rely on such data (Hort et al., 2024), and their absence leaves only a narrow range of post-hoc methods, increasing the risk of suboptimal fairness outcomes. Future work should focus on developing fairness interventions that do not require sensitive features but still produce meaningful group-level improvements (Mehrabi et al., 2019). More generally, data protection and fairness should not be treated as isolated objectives. Meeting privacy standards is not sufficient to ensure fair model behaviour, and achieving fairness can require access to precisely the data that privacy law seeks to limit. Addressing this contradiction will require closer collaboration between legal experts, data scientists, and decision-makers, and the adoption of development practices that integrate fairness and compliance considerations from the outset (Ganesh et al., 2025; Tran & Fioretto, 2023).

7.3 Limitations

While this study provides useful insights into the relationship between data minimisation and algorithmic fairness, several limitations should be considered.

First, the analysis relied on a single dataset: COMPAS. Although it is widely used in fairness research and well-suited for examining group-level disparities (Angwin et al., 2016), its focus on the U.S.

criminal justice system may limit the generalisability of findings to other domains such as finance, healthcare, or employment. The behaviour of models and fairness interventions can vary depending on the data context and decision-making environment (Holmberg et al., 2020; Mehrabi et al., 2019).

Second, the dataset contains significant imbalances across racial groups. Some subgroups, such as Asian and Native American individuals, are underrepresented, which may lead to unreliable or unstable performance and fairness metrics for these groups. This imbalance limits the ability to draw firm conclusions about fairness across all demographic categories and increases the risk of overfitting or noise in subgroup-level analysis (Suresh & Guttag, 2021; Baer, 2019).

Third, the study focused on two classifiers: Logistic Regression and Random Forest. These models were chosen for their interpretability and practical relevance (Caraciolo, 2011; Breiman, 2001) but do not capture the full diversity of machine learning methods used in real-world systems. More complex or opaque models, such as neural networks or large-scale ensembles, may respond differently to both bias and mitigation techniques (Singh et al., 2016).

Fourth, only a limited selection of bias mitigation methods was applied. For the full dataset, the study used reweighting, Exponentiated Gradient Reduction, and the threshold optimiser. For the data-minimised setting, it used thresholding and calibration methods. While these methods were chosen to reflect a range of mitigation strategies (Hort et al., 2024), other approaches such as constraint-based optimisation, adversarial learning, or fairness-aware data generation were not included and may yield different results.

Fifth, the fairness evaluation focused on group-level metrics such as Equalised Odds Difference, Selection Rate disparity, and ROC AUC. These are appropriate for assessing disparities across demographic groups (Hardt et al., 2016) but do not account for individual fairness, causal reasoning, or long-term impacts of algorithmic decisions (Binns, 2018; Mehrabi et al., 2019). In practical deployments, organisations may be required to consider a wider set of fairness perspectives depending on legal, social, or operational priorities.

Finally, the implementation of data minimisation was modelled by removing `race` from training features while retaining it for post hoc fairness evaluation. This reflects a practical constraint that organisations may face under the GDPR (Article 5(1)(c); Ganesh et al., 2025). However, the interpretation of necessity under data protection law is context-dependent (European Data Protection Board, 2020). This study does not resolve legal questions about what constitutes sufficient

justification for sensitive data usage and does not simulate all variations of privacy-preserving data practices.

These limitations indicate that the study's findings should be interpreted in context. They also point to avenues for future research that can address broader datasets, alternative model types, expanded fairness metrics, and legal interpretations of data minimisation.

7.4 Recommendations for future research

This study illustrates the challenges of achieving algorithmic fairness when machine learning models are developed under data minimisation constraints. One important direction for future research is the design of bias mitigation methods that do not rely on access to sensitive attributes. Many current techniques require group-specific demographic information for calibration or constraint-based optimisation (Agarwal et al., 2018; Zafar et al., 2017), which makes them difficult to apply in privacy-sensitive settings. Alternative strategies that use proxy signals, aggregate statistics, or inferred group membership (Holmberg et al., 2020) warrant further investigation. These approaches must be critically assessed for their technical effectiveness and ethical implications, particularly with regard to indirect discrimination.

Future studies could also expand the scope of models examined. This research focused on Logistic Regression and Random Forest due to their interpretability and real-world applicability (Caraciolo, 2011; Rokach & Maimon, 2005). However, as more organisations adopt complex models such as deep neural networks, it is essential to understand how these architectures respond to fairness constraints in the absence of sensitive features. Prior research suggests that model complexity can influence both bias propagation and mitigation effectiveness (Caruana & Niculescu-Mizil, 2006; Suresh & Guttag, 2021).

In addition, interdisciplinary research is needed to explore how legal principles like data minimisation are interpreted and implemented in practical contexts. The GDPR requires that personal data be limited to what is necessary (Regulation (EU) 2016/679), but the definition of necessity is often context-dependent (European Data Protection Board, 2020). Studies could examine how organisations operationalise this principle during model development, including the trade-offs they make between compliance, accuracy, and fairness. Case studies or interviews with practitioners could help clarify whether technical best practices align with legal expectations and ethical norms (Barocas & Selbst, 2016; Binns, 2018).

Another area for exploration is how fairness evaluations are conducted when access to sensitive data is restricted. Recent work has highlighted the limitations of fairness audits that rely solely on non-sensitive features (Sonboli et al., 2024; Tran & Fioretto, 2023). Investigating the use of aggregated demographic summaries, synthetic data, or anonymised group statistics as substitutes could offer practical solutions, though these too must be evaluated for effectiveness and compliance.

Finally, future research should examine how fairness and privacy concerns are communicated to stakeholders, including regulators, business leaders, and affected individuals. As public scrutiny of algorithmic systems increases (Singer, 2018; Wiessner, 2023), transparent reporting practices and standardised fairness documentation become essential for building trust. While some tools and frameworks have been proposed (Bellamy et al., 2018), their real-world adoption remains limited. Research into how these tools are implemented and understood by non-technical audiences could support more accountable and responsible AI governance.

8 Conclusion

This thesis explored the relationship between data minimisation and algorithmic fairness in ML, using the COMPAS dataset to assess how the availability of sensitive attributes, particularly race, affects the ability to reduce bias. The central aim was to evaluate whether fairness goals can be achieved when data protection regulations, such as the General Data Protection Regulation (GDPR), limit access to sensitive data. By comparing model performance and fairness outcomes across both full and data-minimised versions of the dataset, the study contributes to the ongoing discourse on the trade-offs between privacy and fairness in algorithmic decision-making.

To directly answer the research question:

"How effective are different bias mitigation methods across multiple supervised machine learning classifiers when access to sensitive data is restricted due to data minimisation?"

The results show that while the removal of race from training inputs restricted the range of applicable bias mitigation techniques, it also produced fairer baseline models. In the data-minimised setting, both LR and RF achieved lower EOD and EOD-NAA scores at baseline compared to their full-data counterparts. This suggests that excluding sensitive features can prevent certain biased associations from being learned in the first place.

When race was included, fairness-aware methods such as EGR and reweighting significantly reduced disparities in EOD while maintaining strong predictive performance. In the data-minimised setting, mitigation options were limited to post-hoc techniques. While post-hoc methods are generally considered less effective than fairness-aware techniques that operate during model training (Hardt et al., 2016; Hort et al., 2024), this research provides a counter-narrative. The threshold optimiser consistently delivered the best fairness outcomes of the entire study, achieving an EOD-NAA as low as 0.076 for LR and an EOD of 0.317 for RF, both within acceptable accuracy tolerance limits. However, calibration methods such as isotonic regression and Platt scaling were less reliable, with isotonic regression in particular introducing substantial disparities in TPR across subgroups.

These findings align with prior work indicating that many fairness-enhancing techniques rely on sensitive attributes to operate effectively (Hardt et al., 2016; Agarwal et al., 2018; Mehrabi et al., 2019). In their absence, organisations are left with a narrower and less predictable toolkit, a concern also highlighted by Holmberg et al. (2020) and Tran and Fioretto (2023). Beyond model behaviour, this study reinforces a regulatory dilemma: the GDPR's data minimisation principle (Article 5(1)(c); Recital 39) requires limiting the use of personal data, including special category data such as race,

yet such attributes are often essential for measuring and correcting bias. Article 9(2)(g) provides an exception where processing is necessary for substantial public interest with appropriate safeguards, and the empirical results presented here could inform debates on whether fairness evaluation qualifies under this provision.

From a practical perspective, the findings suggest that simpler, interpretable models such as LR may be more suitable for fairness-critical and regulated applications. LR consistently outperformed RF across predictive metrics, demonstrated more stable subgroup behaviour, and responded more effectively to mitigation methods. This supports arguments in the literature advocating for interpretable models in high-stakes decision-making settings (Caraciolo, 2011).

Despite these contributions, this research has limitations. It used a single dataset from the U.S. criminal justice system (Angwin et al., 2016), examined only two classifiers, and applied a limited set of bias mitigation methods. The fairness evaluation was restricted to group-level metrics such as EOD and SR disparity, without addressing individual fairness, causal reasoning, or long-term impacts (Binns, 2018; Mehrabi et al., 2019). As such, the results should be interpreted in context and not overgeneralised.

To summarise, this thesis provides empirical evidence that fairness in ML can, in some cases, improve under data minimisation constraints, as seen in the fairer baselines of both LR and RF. However, these constraints also limit the range of effective mitigation strategies, increasing reliance on a small set of post-hoc tools, which may vary in stability. Addressing this challenge will require technical innovation, clearer legal guidance on the use of sensitive data for fairness evaluation, and closer collaboration between data scientists, regulators, and policymakers. Only through such coordinated efforts can ML systems be developed that are both privacy-compliant and fair.

9 References

Agarwal, A., Beygelzimer, A., Dudík, M., Langford, J., & Wallach, H. (2018). A Reductions Approach to Fair Classification. *Proceedings of Machine Learning Research*, *80*, 60–69. https://proceedings.mlr.press/v80/agarwal18a.html

Al-Zawqari, A. M. M., Peumans, D., & Vandersteen, G. (2024). Latent Space Bias Mitigation for Predicting At-Risk Students. *Computers and Education. Artificial Intelligence*, *7*, Article 100300. https://doi.org/10.1016/j.caeai.2024.100300

Angwin, J., Larson, J., Kirchner, L., & Mattu, S. (2016, May 23). *Machine bias*. ProPublica. https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing

Arnott, D. (2006). Cognitive biases and decision support systems development: A design science approach. *Information Systems Journal*, *16*(1), 55–78. https://doi.org/10.1111/j.1365-2575.2006.00208.x

Awasthi, P., Kleindessner, M., & Morgenstern, J. (2020). Equalized odds postprocessing under imperfect group information. In *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics* (pp. 1770–1780). Proceedings of Machine Learning Research. https://proceedings.mlr.press/v108/awasthi20a.html

Baer, T. (2019). *Understand, Manage, and Prevent Algorithmic Bias*. https://doi.org/10.1007/978-1-4842-4885-0

Baeza-Yates, R. (2018). Bias on the web. *Communications of the ACM*, *61*(6), 54–61. https://doi.org/10.1145/3209581

Barocas, S., Hardt, M., & Narayanan, A. (2019). *Fairness in machine learning: Limitations and opportunities*. Princeton University Press.

Barocas, S., & Selbst, A. D. (2016). Big Data's disparate impact. *SSRN Electronic Journal*. https://doi.org/10.2139/ssrn.2477899

Bellamy, R. K. E., Dey, K., Hind, M., Hoffman, S. C., Houde, S., Kannan, K., Lohia, P., Martino, J., Mehta, S., Mojsilovic, A., Nagar, S., Ramamurthy, K. N., Richards, J., Saha, D., Sattigeri, P., Singh, M., Varshney, K. R., & Zhang, Y. (2018). *AI fairness 360: An extensible toolkit for detecting, understanding, and mitigating unwanted algorithmic bias*. arXiv. http://arxiv.org/abs/1810.01943

Bertrand, M., & Mullainathan, S. (2004). Are Emily and Greg More Employable Than Lakisha and Jamal? A Field Experiment on Labor Market Discrimination. American Economic Review, 94(4), 991-1013.

Binns, R. (2018). Fairness in machine learning: Lessons from political philosophy. In S. A. Friedler & C. Wilson (Eds.), *Proceedings of the 1st Conference on Fairness, Accountability and Transparency* (Vol. 81, pp. 149–159). PMLR. https://proceedings.mlr.press/v81/binns18a.html

Breiman, L. (2001). Random Forests. *Machine Learning*, *45*(1), 5–32. https://doi.org/10.1023/A:1010933404324

Buitinck, L., Louppe, G., Blondel, M., Pedregosa, F., Mueller, A., Grisel, O., Niculae, V., Prettenhofer, P., Gramfort, A., Grobler, J., Layton, R., VanderPlas, J., Joly, A., Holt, B., & Varoquaux, G. (2013). API design for machine learning software: experiences from the scikit-learn project. In *ECML PKDD Workshop: Languages for Data Mining and Machine Learning* (pp. 108–122).

Caraciolo, M. (2011). *Machine Learning with Python - Logistic Regression*. Artificial Intelligence in Motion. Retrieved July 18, 2025, from http://aimotion.blogspot.lt/2011/11/machine-learning-with-python-logistic.html

Caruana, R., & Niculescu-Mizil, A. (2006). An empirical comparison of supervised learning algorithms. In *Proceedings of the 23rd International Conference on Machine Learning* (pp. 161–168). Association for Computing Machinery. https://doi.org/10.1145/1143844.1143865

Chapman, P., Clinton, J., Kerber, R., Khabaza, T., Reinartz, T., Shearer, C., & Wirth, R. (2000). *CRISP-DM 1.0 Step-by-step data mining guide*. The CRISP-DM Consortium.

Chou, Y. H., Ng, C., Cattell, S., Intan, J., Sinclair, M. D., Devietti, J., Rogers, T. G., & Aamodt, T. M. (2020). Deterministic Atomic Buffering. In *2020 53rd Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)* (pp. 981–995). IEEE. https://doi.org/10.1109/MICRO50266.2020.00083

Chouldechova, A. (2017). Fair prediction with disparate impact: A study of bias in recidivism prediction instruments. *Big Data*, *5*(2), 153–163. https://doi.org/10.1089/big.2016.0047

Chui, M., Hall, B., Mayhew, H., Singla, A., & Sukharevsky, A. (2022, December 6). *The state of AI in 2022-and a half decade in Review*. McKinsey & Company. https://www.mckinsey.com/capabilities/quantumblack/our-insights/the-state-of-ai-in-2022-and-a-half-decade-in-review

Collins, E. (2018). Punishing risk. *Georgetown Law Journal, 107, 57-108.*

Corbett-Davies, S., Gaebler, J. D., Nilforoshan, H., Shroff, R., & Goel, S. (2023). *The measure and mismeasure of fairness*. arXiv. https://arxiv.org/abs/1808.00023

Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine Learning*, *20*(3), 273–297. https://doi.org/10.1007/BF00994018

Cover, T. M., & Hart, P. E. (1967). Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, *13*(1), 21–27.

Davenport, T. H., & Ronanki, R. (2018). Artificial intelligence for the real world. Harvard Business Review, 96(1), 108–116.

EEOC Sues iTutorGroup for Age Discrimination. (2022, May 5). *EEOC*. Retrieved from https://www.eeoc.gov/newsroom/eeoc-sues-itutorgroup-age-discrimination.

European Commission. (2019). *Ethics Guidelines for Trustworthy AI*. Publications Office of the European Union. https://ec.europa.eu/futurium/en/ai-alliance-consultation/guidelines/1

European Data Protection Board. (2020). *Guidelines 4/2019 on Article 25 Data Protection by Design and by Default (Version 2.0)*. Retrieved from https://www.edpb.europa.eu

Farayola, M. M., Tal, I., Saber, T., Connolly, R., & Bendechache, M. (2025). A fairness-focused approach to recidivism prediction: implications for accuracy, trust, and equity. *AI & SOCIETY : Journal of Knowledge, Culture and Communication*, 1–19. https://doi.org/10.1007/s00146-025-02452-1

Fawcett, T. (2006). Introduction to ROC analysis. *Pattern Recognition Letters*, *27*, 861–874. https://doi.org/10.1016/j.patrec.2005.10.010

Feldman, M., Friedler, S. A., Moeller, J., Scheidegger, C., & Venkatasubramanian, S. (2015). Certifying and removing disparate impact. *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 259–268. https://doi.org/10.1145/2783258.2783311

Ferrara, E. (2024). Fairness and Bias in Artificial Intelligence: A Brief Survey of Sources, Impacts, and Mitigation Strategies. *Sci*, *6*(1), 3. https://doi.org/10.3390/sci6010003

Fix, E., & Hodges, J. L., Jr. (1951). *Discriminatory analysis. Nonparametric discrimination: Consistency properties* (Technical Report No. 1210-0-296). School of Aviation Medicine, U.S. Air Force, Randolph Field, TX.

Fong, H., Kumar, V., Mehrotra, A., & Vishnoi, N. K. (2022). *Fairness for AUC via Feature Augmentation* [Preprint]. arXiv. https://arxiv.org/abs/2111.12823

Friedler, S. A., Scheidegger, C., Venkatasubramanian, S., Choudhary, S., Hamilton, E. P., & Roth, D. (2019). A comparative study of fairness-enhancing interventions in machine learning. FAT* '19: Proceedings of the 2019 Conference on Fairness, Accountability, and Transparency. https://doi.org/10.1145/3287560.3287589

Friedman, B., & Nissenbaum, H. (1996). Bias in Computer Systems. *ACM Transactions on Information Systems*, *14*(3), 330–347. https://doi.org/10.1145/230538.230561

Goddard, K., Roudsari, A., & Wyatt, J. C. (2012). Automation bias: a systematic review of frequency, effect mediators, and mitigators. *Journal of the American Medical Informatics Association : JAMIA*, *19*(1), 121–127. https://doi.org/10.1136/amiajnl-2011-000089

Gupta, D., & Krishnan, T. S. (2020). Algorithmic bias: Why bother? https://cmr.berkeley.edu/2020/11/algorithmic-bias/

Hardt, M., Price, E., & Srebro, N. (2016). *Equality of opportunity in supervised learning*. arXiv. https://arxiv.org/abs/1610.02413

Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction* (2nd ed.). Springer.

Hastings Blow, C., Qian, L., Gibson, C., Obiomon, P., & Dong, X. (2025). Data augmentation via diffusion model to enhance AI fairness. *Frontiers in Artificial Intelligence*, *8*, 1530397. https://doi.org/10.3389/frai.2025.1530397

Holmberg, M., Skogholt, P., & Tjøstheim, T. A. (2020). *Bias in machine learning: A survey on sources, impacts and reduction*. arXiv. https://arxiv.org/abs/2004.00686

Holstein, K., Wortman Vaughan, J., Daumé III, H., & Wallach, H. (2023). *Human-in-the-loop Fairness: Integrating Stakeholder Feedback to Incorporate Fairness Perspectives in Responsible AI*. arXiv. https://arxiv.org/html/2312.08064v3

Hort, M., Chen, Z., Zhang, J. M., Harman, M., & Sarro, F. (2024). Bias Mitigation for Machine Learning Classifiers: A Comprehensive Survey. *ACM Journal on Responsible Computing*, *1*(2), 1-52.

IBM. (2024, May 6). *Classification in machine learning: A comprehensive guide*. IBM. https://www.ibm.com/think/topics/classification-machine-learning

Jiang, T., Gradus, J. L., & Rosellini, A. J. (2020). Supervised Machine Learning: A Brief Primer. *Behavior therapy*, *51*(5), 675–687. https://doi.org/10.1016/j.beth.2020.05.002

Kamalov, F., Sulieman, H., Alzaatreh, A., Emarly, M., Chamlal, H., & Safaraliev, M. (2025). Mathematical Methods in Feature Selection: A Review. *Mathematics*, *13*(6), 996. https://doi.org/10.3390/math13060996

Kamiran, F., Karim, A., & Zhang, X. (2012). Decision theory for discrimination-aware classification. In *Proceedings of the 12th IEEE International Conference on Data Mining* (pp. 924–929). IEEE.

Kleinberg, J., Mullainathan, S., & Raghavan, M. (2017). Inherent trade-offs in the fair determination of risk scores. *Proceedings of Innovations in Theoretical Computer Science*, 43:1–43:23. https://doi.org/10.4230/LIPIcs.ITCS.2017.43

Kreuzberger, D., Kühl, N., & Hirschl, S. (2023). Machine Learning Operations (MLOps): Overview, Definition, and Architecture. *IEEE Access*, *11*, 31866–31879. https://doi.org/10.1109/ACCESS.2023.3262138

Lan, J., Hu, M. Y., Patuwo, E., & Zhang, G. P. (2010). An investigation of neural network classifiers with unequal misclassification costs and group sizes. *Decision Support Systems*, *48*(4), 582–591. https://doi.org/10.1016/j.dss.2009.11.008

Manning, C. D., Raghavan, P., & Schütze, H. (2008). *Introduction to Information Retrieval*. Cambridge University Press. https://nlp.stanford.edu/IR-book/

Martin, J. (2007, January 1). *Gender-related material in the new Core Curriculum*. Stanford Graduate School of Business. https://www.gsb.stanford.edu/experience/news-history/gender-related-material-new-core-curriculum

Mehrabi, N., Morstatter, F., Saxena, N., Lerman, K., & Galstyan, A. (2019). A survey on bias and fairness in machine learning.

Minatel, D., Parmezan, A. R. S., Roque dos Santos, N., Curi, M., & Lopes, A. (2025). A DIF-driven threshold tuning method for improving group fairness. In *Proceedings of the 40th ACM/SIGAPP Symposium on Applied Computing* (pp. 890–898). Association for Computing Machinery. https://doi.org/10.1145/3672608.3707875

Mullainathan, S., & Obermeyer, Z. (2017, May). Does machine learning automate moral hazard and error? *American Economic Review*, *107*(5), 476–480. https://doi.org/10.1257/aer.p20171084

Nikolić, M., Nikolić, D., Stefanović, M., Koprivica, S., & Stefanović, D. (2025). Mitigating algorithmic bias through probability calibration: A case study on lead generation data. *Mathematics*, *13*(13), 2183. https://doi.org/10.3390/math13132183

Noble, S. U. (2018). *Algorithms of oppression: How search engines reinforce racism*. New York University Press.

Olteanu, A., Castillo, C., Diaz, F., & Kıcıman, E. (2019). Social Data: Biases, methodological pitfalls, and ethical boundaries. *Frontiers in Big Data*, *2*. https://doi.org/10.3389/fdata.2019.00013

Paleyes, A., Urma, R.-G., & Lawrence, N. D. (2022). Challenges in Deploying Machine Learning: A Survey of Case Studies. *ACM Computing Surveys*, *55*(6), Article 114. https://doi.org/10.1145/3533378

Partnership on AI. (2024, July 25). *AI Needs Inclusive Stakeholder Engagement Now More Than Ever*. https://partnershiponai.org/ai-needs-inclusive-stakeholder-engagement-now-more-than-ever/

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Joly, A., Passos, D., Cournapeau, F., Perrot, M., Brucher, R., Péchaud, M., Robert, A., Virginius, J., & Duchesnay, É. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, *12*, 2825–2830.

Platt, J. C. (1999). Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In A. J. Smola, P. Bartlett, B. Schölkopf, & D. Schuurmans (Eds.), *Advances in Large Margin Classifiers* (pp. 61–74). MIT Press.

Pleiss, G., Raghavan, M., Wu, F., Kleinberg, J. M., & Weinberger, K. Q. (2017). On fairness and calibration. In *Proceedings of the Annual Conference on Neural Information Processing Systems 2017* (pp. 5680–5689).

Rabonato, R.T., Berton, L. A systematic review of fairness in machine learning. *AI Ethics* 5, 1943–1954 (2025). https://doi.org/10.1007/s43681-024-00577-5

Raftopoulos, G., Davrazos, G., & Kotsiantis, S. (2025). Evaluating fairness strategies in educational data mining: A comparative study of bias mitigation techniques. *Electronics*, *14*(9), 1856. https://doi.org/10.3390/electronics14091856

Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation). (2016, May 4). *Official Journal of the European Union, L 119*. https://eur-lex.europa.eu/eli/reg/2016/679/oj

Rokach, L., & Maimon, O. (2005). Top-down induction of decision trees classifiers – a survey. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 35(4), 476–487. https://doi.org/10.1109/TSMCC.2004.843247

Schröer, C., Kruse, F., & Marx Gómez, J. (2021). A systematic literature review on applying CRISP-DM process model. *Procedia Computer Science*, *181*, 526–534. https://doi.org/10.1016/j.procs.2021.01.199

Siddique, S., Haque, M. A., George, R., Gupta, K. D., Gupta, D., & Faruk, M. J. H. (2024). Survey on machine learning biases and mitigation techniques. *Digital*, *4*(1), 1–68. https://doi.org/10.3390/digital4010001

Silva, H., & Bernardino, J. (2022). Machine Learning Algorithms: An Experimental Evaluation for Decision Support Systems. *Algorithms*, *15*(4), 130. https://doi.org/10.3390/a15040130

Singer, N. (2018, July 26). Amazon's facial recognition wrongly identifies 28 lawmakers, a.c.l.u. says. *The New York Times*. Retrieved from

https://www.nytimes.com/2018/07/26/technology/amazon-aclu-facial-recognition-congress.html.

Singh, A., Thakur, N., & Sharma, A. (2016). A review of supervised machine learning algorithms. *2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom)*, 1310–1315.

Sonboli, N., Li, S., Elahi, M., & Biega, A. (2024). *The trade-off between data minimization and fairness in collaborative filtering*. arXiv. https://arxiv.org/abs/2410.07182

Suresh, H., Gong, J. J., & Guttag, J. V. (2018). Learning tasks for multitask learning. *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. https://doi.org/10.1145/3219819.3219930

Suresh, H., & Guttag, J. (2021). A Framework for Understanding Sources of Harm throughout the Machine Learning Life Cycle. *Equity and Access in Algorithms, Mechanisms, and Optimization*. ACM.

Suyal and Goyal (2022) - Journal Article Suyal, M., & Goyal, P. (2022). A review on analysis of K-Nearest Neighbor classification machine learning algorithms based on supervised learning. *International Journal of Engineering Trends and Technology*, *70*(7), 43–48.

Tran, C., & Fioretto, F. (2023). *Data minimization at inference time*. arXiv. https://arxiv.org/abs/2305.17593

Turban, E., Aronson, J. E., & Liang, T. P. (2005). *Decision Support Systems and Intelligent Systems* (7th ed.). Pearson Prentice Hall.

van Giffen, B., Herhausen, D., & Fahse, T. (2022). Overcoming the pitfalls and perils of algorithms: A classification of machine learning biases and mitigation methods. *Journal of Business Research*, *144*, 93–106. https://doi.org/10.1016/j.jbusres.2022.01.076

Wang, Y., & Singh, L. (2025). Impact on bias mitigation algorithms to variations in inferred sensitive attribute uncertainty. *Frontiers in Artificial Intelligence*, *8*, 1520330. https://doi.org/10.3389/frai.2025.1520330

Wiessner, D. (2023, August 10). Tutoring firm settles US agency's first bias lawsuit involving ai software. *Reuters*. Retrieved from https://www.reuters.com/legal/tutoring-firm-settles-us-agencys-first-bias-lawsuit-involving-ai-software-2023-08-10/.

Yan, B., Seto, S., & Apostoloff, N. (2022). *FORML: Learning to reweight data for fairness*. arXiv. https://arxiv.org/abs/2202.01719

Yang, J., Clifton, L., Dung, N. T., & et al. (2024). Mitigating machine learning bias between high income and low–middle income countries for enhanced model fairness and generalizability. *Scientific Reports*, *14*, 13318. https://doi.org/10.1038/s41598-024-64210-5

Yang, J., Soltan, A. A. S., Eyre, D. W., & et al. (2023). An adversarial training framework for mitigating algorithmic biases in clinical machine learning. *npj Digital Medicine*, *6*, 55. https://doi.org/10.1038/s41746-023-00805-y

Zadrozny, B., & Elkan, C. (2002, July). Transforming classifier scores into accurate multiclass probability estimates. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 694–699). Association for Computing Machinery. https://doi.org/10.1145/775047.775143

Zafar, M. B., Valera, I., Gomez Rodriguez, M., & Gummadi, K. P. (2017). Fairness Constraints: Mechanisms for Fair Classification. *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS)*, 30, 962–970.

Zarya, V. (2018, May 21). The share of female CEOs in the Fortune 500 dropped by 25% in 2018. *Fortune*. https://fortune.com/2018/05/21/women-fortune-500-2018

Zhao, X., Fabbrizzi, S., Lobo, P. R., Ghodsi, S., Broelemann, K., Staab, S., & Kasneci, G. (2023). *Adversarial reweighting guided by Wasserstein distance for bias mitigation*. arXiv. https://arxiv.org/abs/2311.12684

Zhuang, D., Zhang, X., Song, S., & Hooker, S. (2022). Randomness in neural network training: Characterizing the impact of tooling. In D. Marculescu, Y. Chi, & C. Wu (Eds.), *Proceedings of Machine Learning and Systems* (Vol. 4, pp. 316–336). MLSys. https://proceedings.mlsys.org/paper_files/paper/2022/file/427e0e886ebf87538afdf0badb805b7f-Paper.pdf

Zimelewicz, E., Kalinowski, M., Méndez Fernández, D., Giray, G., Alves, A., Lavesson, N., Azevedo, K., Villamizar, H., Escovedo, T., Lopes, H., Biffl, S., Musil, J., Felderer, M., Wagner, S., Baldassarre, M., & Gorschek, T. (2024). ML-Enabled Systems Model Deployment and Monitoring: Status Quo and Problems. In U. Frank, F. Macias, M. Felderer, & S. Wagner (Eds.), *Software Engineering Aspects of Machine Learning: 5th International Workshop, SEAML 2023, Toronto, ON, Canada, May 15, 2023, Revised Selected Papers* (pp. 94–112). Springer Nature Switzerland. https://doi.org/10.1007/978-3-031-56281-5_7

Appendices

9.1 Analysis of COMPAS dataset features

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.feature_selection import mutual_info_classif
import numpy as np

RANDOM_STATE = 0

# Load dataset
df = pd.read_csv("compas-scores-two-years.csv")

# Convert dates and drop rows with missing values
df["compas_screening_date"] = pd.to_datetime(df["compas_screening_date"],
errors='coerce')
df["out_custody"] = pd.to_datetime(df["out_custody"], errors='coerce')
df = df.dropna(subset=["compas_screening_date", "out_custody"])

# Apply standard COMPAS data filters
df = df[
    (df["days_b_screening_arrest"] <= 30) &
    (df["days_b_screening_arrest"] >= -30) &
    (df["is_recid"] != -1) &
    (df["c_charge_degree"] != "O") &
    (df["score_text"] != "N/A")
].reset_index(drop=True)

# Select features of interest
features = ["sex", "age", "race", "priors_count", "c_charge_degree"]
X = df[features]

# One-hot encode features (keep all race categories)
encoded = pd.get_dummies(X, columns=["sex", "c_charge_degree"],
drop_first=True)
race_dummies = pd.get_dummies(X["race"], prefix="race", drop_first=False)
encoded = pd.concat([encoded.drop(columns=["race"]), race_dummies],
axis=1)

# Identify race and non-race columns
race_cols = [c for c in encoded.columns if c.startswith("race_")]
non_race_cols = [c for c in encoded.columns if not c.startswith("race_")]

# Pearson correlation between race and other features
corr = encoded[race_cols + non_race_cols].corr().loc[race_cols,
non_race_cols]
plt.figure(figsize=(8, 5))
sns.heatmap(corr, annot=True, cmap="coolwarm", center=0)
plt.title("Correlation between race and other features")
plt.show()

# Mutual information between race and non-race features
```

```
mi_scores = {}
for col in non_race_cols:
    is_discrete = encoded[col].dropna().isin([0, 1]).all()
    vals = []
    for rc in race_cols:
        mi = mutual_info_classif(
            encoded[[col]], encoded[rc].astype(int),
            discrete_features=[is_discrete],
            random_state=RANDOM_STATE
        )[0]
        vals.append(mi)
    mi_scores[col] = float(np.mean(vals))

# Bar plot of mutual information scores
mi_df = pd.DataFrame({
    "Feature": list(mi_scores.keys()),
    "Mutual information": list(mi_scores.values())
}).sort_values("Mutual information", ascending=False)

plt.figure(figsize=(8, 5))
sns.barplot(data=mi_df, x="Mutual information", y="Feature",
color="#4c72b0")
plt.title("Mutual information with race")
plt.xlabel("Mutual information")
plt.ylabel("Feature")
plt.show()

# Distribution of race in dataset
plt.figure(figsize=(8, 5))
sns.countplot(data=df, x="race", order=df["race"].value_counts().index,
palette="muted")
plt.title("Racial distribution in filtered COMPAS dataset")
plt.xlabel("Race")
plt.ylabel("Count")
plt.xticks(rotation=30)
plt.tight_layout()
plt.show()
```

9.2 Analysis of baseline LR model (full dataset)

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt

import statsmodels.formula.api as smf
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import (
    accuracy_score, roc_auc_score, f1_score, classification_report,
roc_curve
)
from fairlearn.metrics import MetricFrame, selection_rate,
true_positive_rate, false_positive_rate

RANDOM_STATE = 0

# Load and filter dataset
df = pd.read_csv("compas-scores-two-years.csv")
df["compas_screening_date"] = pd.to_datetime(df["compas_screening_date"],
errors="coerce")
df["out_custody"] = pd.to_datetime(df["out_custody"], errors="coerce")
df = df.dropna(subset=["compas_screening_date", "out_custody"])
df = df[
    (df["days_b_screening_arrest"] <= 30) &
    (df["days_b_screening_arrest"] >= -30) &
    (df["is_recid"] != -1) &
    (df["c_charge_degree"] != "O") &
    (df["score_text"] != "N/A")
].reset_index(drop=True)

# Features and target
features = ["sex", "age", "race", "priors_count", "c_charge_degree"]
X = df[features]
y = df["two_year_recid"]
sens = df["race"]

# One-hot encode and split
X_encoded = pd.get_dummies(X, columns=["sex", "race", "c_charge_degree"],
drop_first=True)
X_train, X_test, y_train, y_test, sens_train, sens_test =
train_test_split(
    X_encoded, y, sens, test_size=0.30, stratify=y,
random_state=RANDOM_STATE
)

# Train logistic regression model
lr = LogisticRegression(max_iter=1000, random_state=RANDOM_STATE)
lr.fit(X_train, y_train)

# Predictions
y_pred = lr.predict(X_test)
y_prob = lr.predict_proba(X_test)[:, 1]
```

```
# Performance metrics
print("Accuracy:", accuracy_score(y_test, y_pred))
print("ROC AUC:", roc_auc_score(y_test, y_prob))
print("F1 Score:", f1_score(y_test, y_pred))
print("\nClassification Report:\n", classification_report(y_test,
y_pred))

# Fairness metrics by race
metrics = {
    "selection_rate": selection_rate,
    "true_positive_rate": true_positive_rate,
    "false_positive_rate": false_positive_rate,
}
fairness_metrics = MetricFrame(
    metrics=metrics, y_true=y_test, y_pred=y_pred,
sensitive_features=sens_test
)
print("Fairness metrics by race:")
print(fairness_metrics.by_group)

# Fairness disparities
print("\nFairness disparities (max-min):")
print(fairness_metrics.difference())

# Equalised Odds Difference
eod = max(
    fairness_metrics.by_group["true_positive_rate"].max() -
fairness_metrics.by_group["true_positive_rate"].min(),
    fairness_metrics.by_group["false_positive_rate"].max() -
fairness_metrics.by_group["false_positive_rate"].min(),
)
print(f"\nEqualised Odds Difference: {eod:.4f}")

# Group-wise ROC AUC
group_auc_scores = {}
for group in sens_test.unique():
    idx = sens_test == group
    group_auc_scores[group] = roc_auc_score(y_test[idx], y_prob[idx])

print("\nGroup-wise ROC AUC:")
for group, auc_val in group_auc_scores.items():
    print(f"{group}: {auc_val:.4f}")

# Prediction score distributions by race
df_plot = pd.DataFrame({"race": sens_test.values, "score":
y_prob}).dropna()
race_order = ["African-American", "Caucasian", "Hispanic", "Other",
"Asian", "Native American"]
race_order = [r for r in race_order if r in df_plot["race"].unique()]

plt.figure(figsize=(12, 7))
for r in race_order:
    s = df_plot.loc[df_plot["race"] == r, "score"]
    if len(s) < 5:
        continue
```

```
    sns.kdeplot(s, label=r, fill=True, alpha=0.25, bw_adjust=1.0,
clip=(0, 1), common_norm=False)

plt.title("Prediction score distributions by race")
plt.xlabel("Predicted score")
plt.ylabel("Density")
plt.xlim(0, 1)
plt.legend(title="race")
plt.tight_layout()
plt.show()

# ROC curve
fpr_lr, tpr_lr, _ = roc_curve(y_test, y_prob)
roc_auc_lr = roc_auc_score(y_test, y_prob)

plt.figure(figsize=(6, 6))
plt.plot(fpr_lr, tpr_lr, lw=2, label=f"LR (AUC = {roc_auc_lr:.3f})")
plt.plot([0, 1], [0, 1], "k--", lw=1, label="Chance")
plt.xlim(0, 1)
plt.ylim(0, 1.05)
plt.xlabel("False Positive Rate")
plt.ylabel("True Positive Rate")
plt.title("ROC curve — baseline Logistic Regression")
plt.legend(loc="lower right")
plt.tight_layout()
plt.show()

# Train statsmodels logistic regression on same split
train_idx = X_train.index
train_df = X.loc[train_idx].copy()
train_df["two_year_recid"] = y.loc[train_idx].values

# Reference categories
race_ref = "Other" if "Other" in train_df["race"].unique() else
train_df["race"].mode().iat[0]
sex_ref = "Female" if "Female" in train_df["sex"].unique() else
train_df["sex"].mode().iat[0]
chg_ref = "F" if "F" in train_df["c_charge_degree"].unique() else
train_df["c_charge_degree"].mode().iat[0]

# Model formula
formula = (
    f"two_year_recid ~ age + priors_count "
    f"+ C(sex, Treatment(reference='{sex_ref}')) "
    f"+ C(race, Treatment(reference='{race_ref}')) "
    f"+ C(c_charge_degree, Treatment(reference='{chg_ref}'))"
)
logit_res = smf.logit(formula=formula, data=train_df).fit(disp=False)

# Wald test for race
wald_terms = logit_res.wald_test_terms(skip_single=False)
race_row = [i for i in wald_terms.table.index if "race" in i.lower()][0]
race_wald = wald_terms.table.loc[race_row]

# Key coefficients and p-values
keep_terms = [
```

```
    f"C(sex, Treatment(reference='{sex_ref}'))[T.Male]",
    f"C(c_charge_degree, Treatment(reference='{chg_ref}'))[T.M]",
    "age", "priors_count"
]
coef = logit_res.params.reindex(keep_terms)
pval = logit_res.pvalues.reindex(keep_terms)
se   = logit_res.bse.reindex(keep_terms)
ci   = logit_res.conf_int().loc[keep_terms]
odds = np.exp(coef)
key_tbl = pd.DataFrame({
    "coef": coef, "std_err": se, "p_value": pval,
    "ci_low": ci[0], "ci_high": ci[1], "odds_ratio": odds
}).rename_axis("term").reset_index()

# Compare sklearn vs statsmodels coefficients
sk_series = pd.Series(lr.coef_[0], index=X_train.columns,
name="sklearn_L2")
sm_map = {"age": logit_res.params["age"], "priors_count":
logit_res.params["priors_count"]}
sm_map["sex_Male"] = logit_res.params.get(f"C(sex,
Treatment(reference='{sex_ref}'))[T.Male]", np.nan)
sm_map["c_charge_degree_M"] = logit_res.params.get(f"C(c_charge_degree,
Treatment(reference='{chg_ref}'))[T.M]", np.nan)
for r in X["race"].unique():
    if r == race_ref:
        continue
    sm_map[f"race_{r}"] = logit_res.params.get(f"C(race,
Treatment(reference='{race_ref}'))[T.{r}]", np.nan)

sm_series = pd.Series(sm_map,
name="statsmodels_no_reg").reindex(sk_series.index)
cmp = pd.concat([sk_series, sm_series], axis=1)
valid = cmp.dropna()
corr = valid.corr().iloc[0, 1]
rmse = np.sqrt(np.mean((valid["sklearn_L2"] -
valid["statsmodels_no_reg"])**2))

# Outputs
print("\nWald test for race (joint significance):")
print(race_wald)
print("\nKey coefficients with p-values (statsmodels, no
regularisation):")
print(key_tbl.to_string(index=False, float_format=lambda x: f"{x:.4f}"))
print("\nCoefficient alignment: sklearn vs statsmodels:")
print(cmp.to_string(float_format=lambda x: f"{x:.6f}"))
print(f"\nCoefficient vector similarity: Pearson r = {corr:.4f}, RMSE =
{rmse:.6f}")
```

9.3 Analysis of baseline RF model (full dataset)

```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import (
    accuracy_score, roc_auc_score, f1_score, classification_report,
roc_curve
)
from fairlearn.metrics import MetricFrame, selection_rate,
true_positive_rate, false_positive_rate
import matplotlib.pyplot as plt
import seaborn as sns

RANDOM_STATE = 0

# Load dataset
df = pd.read_csv("compas-scores-two-years.csv")

# Convert dates and drop missing values
df["compas_screening_date"] = pd.to_datetime(df["compas_screening_date"],
errors="coerce")
df["out_custody"] = pd.to_datetime(df["out_custody"], errors="coerce")
df = df.dropna(subset=["compas_screening_date", "out_custody"])

# Apply filters
df = df[
    (df["days_b_screening_arrest"] <= 30) &
    (df["days_b_screening_arrest"] >= -30) &
    (df["is_recid"] != -1) &
    (df["c_charge_degree"] != "O") &
    (df["score_text"] != "N/A")
].reset_index(drop=True)

# Select features and target
features = ["sex", "age", "race", "priors_count", "c_charge_degree"]
X = df[features]
y = df["two_year_recid"]

# Save sensitive attribute for fairness metrics
sensitive_attr = df["race"]

# One-hot encode categorical features
X_encoded = pd.get_dummies(X, columns=["sex", "race", "c_charge_degree"],
drop_first=True)

# Train/test split
X_train, X_test, y_train, y_test, sens_train, sens_test =
train_test_split(
    X_encoded, y, sensitive_attr, test_size=0.3, stratify=y,
random_state=RANDOM_STATE
)

# Train Random Forest model
```

```
rf_clf = RandomForestClassifier(random_state=RANDOM_STATE)
rf_clf.fit(X_train, y_train)

# Predictions and probabilities
y_pred = rf_clf.predict(X_test)
y_proba = rf_clf.predict_proba(X_test)[:, 1]

# Classification metrics
print("Accuracy:", accuracy_score(y_test, y_pred))
print("ROC AUC:", roc_auc_score(y_test, y_proba))
print("F1 Score:", f1_score(y_test, y_pred))
print("\nClassification Report:\n", classification_report(y_test,
y_pred))

# Fairness metrics by race
metrics = {
    "selection_rate": selection_rate,
    "true_positive_rate": true_positive_rate,
    "false_positive_rate": false_positive_rate,
}
fairness_metrics = MetricFrame(
    metrics=metrics,
    y_true=y_test,
    y_pred=y_pred,
    sensitive_features=sens_test
)
print("Fairness Metrics by Race:")
display(fairness_metrics.by_group)
print("\nFairness Disparities (Max-Min):")
display(fairness_metrics.difference())

# Equalised Odds Difference
equalised_odds_diff = max(
    fairness_metrics.by_group["true_positive_rate"].max() -
fairness_metrics.by_group["true_positive_rate"].min(),
    fairness_metrics.by_group["false_positive_rate"].max() -
fairness_metrics.by_group["false_positive_rate"].min()
)
print(f"\nEqualised Odds Difference: {equalised_odds_diff:.4f}")

# Group-wise ROC AUC
group_auc_scores = {}
for group in sens_test.unique():
    group_idx = sens_test == group
    group_auc = roc_auc_score(y_test[group_idx], y_proba[group_idx])
    group_auc_scores[group] = group_auc
print("\nGroup-wise ROC AUC Scores:")
for group, auc in group_auc_scores.items():
    print(f"{group}: {auc:.4f}")

# Prediction score distributions by race
df_plot = pd.DataFrame({"race": sens_test.values, "score":
y_proba}).dropna()
race_order = ["African-American", "Caucasian", "Hispanic", "Other",
"Asian", "Native American"]
race_order = [r for r in race_order if r in df_plot["race"].unique()]
```

```
plt.figure(figsize=(12, 7))
for r in race_order:
    s = df_plot.loc[df_plot["race"] == r, "score"]
    if len(s) < 5:
        continue
    sns.kdeplot(s, label=r, fill=True, alpha=0.25, bw_adjust=1.0,
clip=(0, 1), common_norm=False)
plt.title("Prediction score distributions by race")
plt.xlabel("Predicted score")
plt.ylabel("Density")
plt.xlim(0, 1)
plt.legend(title="race")
plt.tight_layout()
plt.show()

# ROC curve
fpr_rf, tpr_rf, _ = roc_curve(y_test, y_proba)
auc_rf = roc_auc_score(y_test, y_proba)
plt.figure(figsize=(6, 6))
plt.plot(fpr_rf, tpr_rf, lw=2, label=f"RF (AUC = {auc_rf:.3f})")
plt.plot([0, 1], [0, 1], "k--", lw=1, label="Chance")
plt.xlabel("False Positive Rate")
plt.ylabel("True Positive Rate")
plt.title("ROC curve - baseline Random Forest")
plt.legend(loc="lower right")
plt.tight_layout()
plt.show()
```

9.4 Analysis of LR model with reweighting (full dataset)

```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, roc_auc_score, f1_score,
classification_report
from fairlearn.metrics import MetricFrame, selection_rate,
true_positive_rate, false_positive_rate

RANDOM_STATE = 0

# Load and filter dataset
df = pd.read_csv("compas-scores-two-years.csv")
df["compas_screening_date"] = pd.to_datetime(df["compas_screening_date"],
errors="coerce")
df["out_custody"] = pd.to_datetime(df["out_custody"], errors="coerce")
df = df.dropna(subset=["compas_screening_date", "out_custody"])
df = df[
    (df["days_b_screening_arrest"] <= 30) &
    (df["days_b_screening_arrest"] >= -30) &
    (df["is_recid"] != -1) &
    (df["c_charge_degree"] != "O") &
    (df["score_text"] != "N/A")
].reset_index(drop=True)

# Features, target, and sensitive attribute
features = ["sex", "age", "race", "priors_count", "c_charge_degree"]
X = df[features]
y = df["two_year_recid"]
sens = df["race"]

# One-hot encode and split
X_encoded = pd.get_dummies(X, columns=["sex", "race", "c_charge_degree"],
drop_first=True)
X_train, X_test, y_train, y_test, sens_train, sens_test =
train_test_split(
    X_encoded, y, sens, test_size=0.30, stratify=y,
random_state=RANDOM_STATE
)

# Reweighting: compute sample weights to decorrelate race and label
(P(r)*P(y)/P(r,y))
train_df = pd.DataFrame({"race": sens_train, "label": y_train})
p_race = train_df["race"].value_counts(normalize=True)
p_label = train_df["label"].value_counts(normalize=True)
p_joint = (
    train_df.groupby(["race", "label"])
    .size()
    .div(len(train_df))
    .rename("p_joint")
    .reset_index()
)
train_df = train_df.merge(p_joint, on=["race", "label"], how="left")
```

```
train_df["p_race"] = train_df["race"].map(p_race)
train_df["p_label"] = train_df["label"].map(p_label)
sample_weights = (train_df["p_race"] * train_df["p_label"] /
train_df["p_joint"]).values

# Train logistic regression with sample weights
lr = LogisticRegression(max_iter=1000, random_state=RANDOM_STATE)
lr.fit(X_train, y_train, sample_weight=sample_weights)

# Predictions
y_pred = lr.predict(X_test)
y_prob = lr.predict_proba(X_test)[:, 1]

# Performance metrics
print("Accuracy:", accuracy_score(y_test, y_pred))
print("ROC AUC:", roc_auc_score(y_test, y_prob))
print("F1 Score:", f1_score(y_test, y_pred))
print("\nClassification Report:\n", classification_report(y_test,
y_pred))

# Fairness metrics by race
metrics = {
    "selection_rate": selection_rate,
    "true_positive_rate": true_positive_rate,
    "false_positive_rate": false_positive_rate,
}
fairness_metrics = MetricFrame(
    metrics=metrics,
    y_true=y_test,
    y_pred=y_pred,
    sensitive_features=sens_test
)
print("Fairness metrics by race:")
print(fairness_metrics.by_group)

# Disparities (max-min) and EOD
print("\nFairness disparities (max-min):")
print(fairness_metrics.difference())
eod = max(
    fairness_metrics.by_group["true_positive_rate"].max() -
fairness_metrics.by_group["true_positive_rate"].min(),
    fairness_metrics.by_group["false_positive_rate"].max() -
fairness_metrics.by_group["false_positive_rate"].min(),
)
print(f"\nEqualised Odds Difference: {eod:.4f}")

# Group-wise ROC AUC
group_auc_scores = {}
for group in sens_test.unique():
    idx = sens_test == group
    group_auc_scores[group] = roc_auc_score(y_test[idx], y_prob[idx])

print("\nGroup-wise ROC AUC Scores:")
for group, auc_val in group_auc_scores.items():
    print(f"{group}: {auc_val:.4f}")
```

9.5 Analysis of LR model with EGR (full dataset)

```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, roc_auc_score, f1_score,
classification_report
from fairlearn.metrics import MetricFrame, selection_rate,
true_positive_rate, false_positive_rate
from fairlearn.reductions import ExponentiatedGradient, EqualizedOdds

RANDOM_STATE = 0

# Load and preprocess COMPAS dataset
df = pd.read_csv("compas-scores-two-years.csv")
df["compas_screening_date"] = pd.to_datetime(df["compas_screening_date"],
errors='coerce')
df["out_custody"] = pd.to_datetime(df["out_custody"], errors='coerce')
df = df.dropna(subset=["compas_screening_date", "out_custody"])
df = df[
    (df["days_b_screening_arrest"] <= 30) &
    (df["days_b_screening_arrest"] >= -30) &
    (df["is_recid"] != -1) &
    (df["c_charge_degree"] != "O") &
    (df["score_text"] != "N/A")
].reset_index(drop=True)

# Features, target, and sensitive attribute
features = ["sex", "age", "race", "priors_count", "c_charge_degree"]
X = df[features]
y = df["two_year_recid"]
sensitive_attr = df["race"]

# One-hot encode categorical features
X_encoded = pd.get_dummies(X, columns=["sex", "race", "c_charge_degree"],
drop_first=True)

# Split data
X_train, X_test, y_train, y_test, sens_train, sens_test =
train_test_split(
    X_encoded, y, sensitive_attr, test_size=0.3, stratify=y,
random_state=RANDOM_STATE
)

# Define LR model with Equalised Odds constraint
base_estimator = LogisticRegression(max_iter=1000,
random_state=RANDOM_STATE)
constraint = EqualizedOdds()
expgrad = ExponentiatedGradient(estimator=base_estimator,
constraints=constraint)
expgrad.fit(X_train, y_train, sensitive_features=sens_train)

# Predict probabilities and apply fixed threshold
probs = np.zeros(len(X_test), dtype=float)
```

```
for h, w in zip(expgrad.predictors_, expgrad.weights_):
    probs += w * h.predict_proba(X_test)[:, 1]
y_prob = probs
y_pred = (y_prob >= 0.5).astype(int)

# Performance metrics
print("Accuracy:", accuracy_score(y_test, y_pred))
print("ROC AUC:", roc_auc_score(y_test, y_prob))
print("F1 Score:", f1_score(y_test, y_pred))
print("\nClassification Report:\n", classification_report(y_test,
y_pred))

# Fairness metrics by race
metrics = {
    "selection_rate": selection_rate,
    "true_positive_rate": true_positive_rate,
    "false_positive_rate": false_positive_rate,
}
fairness_metrics = MetricFrame(metrics=metrics, y_true=y_test,
y_pred=y_pred, sensitive_features=sens_test)
print("Fairness Metrics by Race:")
display(fairness_metrics.by_group)
print("\nFairness Disparities (Max-Min):")
display(fairness_metrics.difference())

# Equalised Odds Difference
equalised_odds_diff = max(
    fairness_metrics.by_group['true_positive_rate'].max() -
fairness_metrics.by_group['true_positive_rate'].min(),
    fairness_metrics.by_group['false_positive_rate'].max() -
fairness_metrics.by_group['false_positive_rate'].min()
)
print(f"\nEqualised Odds Difference: {equalised_odds_diff:.4f}")

# Group-wise ROC AUC
group_auc_scores = {}
for group in sens_test.unique():
    idx = sens_test == group
    group_auc_scores[group] = roc_auc_score(y_test[idx], y_prob[idx])
print("\nGroup-wise ROC AUC Scores:")
for group, auc in group_auc_scores.items():
    print(f"{group}: {auc:.4f}")
```

9.6 Analysis of LR model with threshold optimiser (full dataset)

```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, roc_auc_score, f1_score,
classification_report
from fairlearn.metrics import MetricFrame, selection_rate,
true_positive_rate, false_positive_rate
from fairlearn.postprocessing import ThresholdOptimizer

RANDOM_STATE = 0

# Load and preprocess dataset
df = pd.read_csv("compas-scores-two-years.csv")
df["compas_screening_date"] = pd.to_datetime(df["compas_screening_date"],
errors="coerce")
df["out_custody"] = pd.to_datetime(df["out_custody"], errors="coerce")
df = df.dropna(subset=["compas_screening_date", "out_custody"])
df = df[
    (df["days_b_screening_arrest"] <= 30) &
    (df["days_b_screening_arrest"] >= -30) &
    (df["is_recid"] != -1) &
    (df["c_charge_degree"] != "O") &
    (df["score_text"] != "N/A")
].reset_index(drop=True)

# Features, target, and sensitive attribute
features = ["sex", "age", "race", "priors_count", "c_charge_degree"]
X = df[features]
y = df["two_year_recid"]
sens = df["race"]

# One-hot encode and split
X_encoded = pd.get_dummies(X, columns=["sex", "race", "c_charge_degree"],
drop_first=True)
X_train, X_test, y_train, y_test, sens_train, sens_test =
train_test_split(
    X_encoded, y, sens, test_size=0.30, stratify=y,
random_state=RANDOM_STATE
)

# Train base LR
base_model = LogisticRegression(max_iter=1000, random_state=RANDOM_STATE)
base_model.fit(X_train, y_train)

# Threshold optimiser with equalised odds
threshold_optim = ThresholdOptimizer(
    estimator=base_model,
    constraints="equalized_odds",
    predict_method="predict_proba",
    prefit=True
)
threshold_optim.fit(X_train, y_train, sensitive_features=sens_train)
```

```
# Post-processed predictions (group-specific thresholds)
y_pred = threshold_optim.predict(X_test, sensitive_features=sens_test)

# Use base model probabilities for ROC AUC (ranking is unchanged by
thresholding)
y_prob = base_model.predict_proba(X_test)[:, 1]

# Performance metrics
print("Accuracy:", accuracy_score(y_test, y_pred))
print("ROC AUC:", roc_auc_score(y_test, y_prob))
print("F1 Score:", f1_score(y_test, y_pred))
print("\nClassification Report:\n", classification_report(y_test,
y_pred))

# Fairness metrics by race
metrics = {
    "selection_rate": selection_rate,
    "true_positive_rate": true_positive_rate,
    "false_positive_rate": false_positive_rate,
}
fairness_metrics = MetricFrame(
    metrics=metrics,
    y_true=y_test,
    y_pred=y_pred,
    sensitive_features=sens_test
)
print("Fairness metrics by race:")
print(fairness_metrics.by_group)

# Disparities (max-min) and EOD
print("\nFairness disparities (max-min):")
print(fairness_metrics.difference())
eod = max(
    fairness_metrics.by_group["true_positive_rate"].max() -
fairness_metrics.by_group["true_positive_rate"].min(),
    fairness_metrics.by_group["false_positive_rate"].max() -
fairness_metrics.by_group["false_positive_rate"].min(),
)
print(f"\nEqualised Odds Difference: {eod:.4f}")

# Group-wise ROC AUC
group_auc_scores = {}
for group in sens_test.unique():
    idx = sens_test == group
    group_auc_scores[group] = roc_auc_score(y_test[idx], y_prob[idx])

print("\nGroup-wise ROC AUC Scores:")
for group, auc_val in group_auc_scores.items():
    print(f"{group}: {auc_val:.4f}")
```

9.7 Analysis of RF model with reweighting (full dataset)

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, roc_auc_score, f1_score,
classification_report
from fairlearn.metrics import MetricFrame, selection_rate,
true_positive_rate, false_positive_rate

RANDOM_STATE = 0

# Load and preprocess dataset
df = pd.read_csv("compas-scores-two-years.csv")
df["compas_screening_date"] = pd.to_datetime(df["compas_screening_date"],
errors="coerce")
df["out_custody"] = pd.to_datetime(df["out_custody"], errors="coerce")
df = df.dropna(subset=["compas_screening_date", "out_custody"])
df = df[
    (df["days_b_screening_arrest"] <= 30) &
    (df["days_b_screening_arrest"] >= -30) &
    (df["is_recid"] != -1) &
    (df["c_charge_degree"] != "O") &
    (df["score_text"] != "N/A")
].reset_index(drop=True)

# Features, target, and sensitive attribute
features = ["sex", "age", "race", "priors_count", "c_charge_degree"]
X = df[features]
y = df["two_year_recid"]
sens = df["race"]

# One-hot encode and split
X_encoded = pd.get_dummies(X, columns=["sex", "race", "c_charge_degree"],
drop_first=True)
X_train, X_test, y_train, y_test, sens_train, sens_test =
train_test_split(
    X_encoded, y, sens, test_size=0.30, stratify=y,
random_state=RANDOM_STATE
)

# Reweighting: w = P(race) * P(label) / P(race, label)
train_df = pd.DataFrame({"race": sens_train, "label": y_train})
p_race = train_df["race"].value_counts(normalize=True)
p_label = train_df["label"].value_counts(normalize=True)
p_joint = (
    train_df.groupby(["race", "label"])
    .size()
    .div(len(train_df))
    .rename("p_joint")
    .reset_index()
)
train_df = train_df.merge(p_joint, on=["race", "label"], how="left")
train_df["p_race"] = train_df["race"].map(p_race)
train_df["p_label"] = train_df["label"].map(p_label)
```

```
sample_weights = (train_df["p_race"] * train_df["p_label"] /
train_df["p_joint"]).values

# Train RF with sample weights
rf = RandomForestClassifier(random_state=RANDOM_STATE)
rf.fit(X_train, y_train, sample_weight=sample_weights)

# Predictions and probabilities
y_pred = rf.predict(X_test)
y_proba = rf.predict_proba(X_test)[:, 1]

# Performance metrics
print("Accuracy:", accuracy_score(y_test, y_pred))
print("ROC AUC:", roc_auc_score(y_test, y_proba))
print("F1 Score:", f1_score(y_test, y_pred))
print("\nClassification Report:\n", classification_report(y_test,
y_pred))

# Fairness metrics by race
metrics = {
    "selection_rate": selection_rate,
    "true_positive_rate": true_positive_rate,
    "false_positive_rate": false_positive_rate,
}
fairness_metrics = MetricFrame(
    metrics=metrics,
    y_true=y_test,
    y_pred=y_pred,
    sensitive_features=sens_test
)
print("Fairness metrics by race:")
print(fairness_metrics.by_group)

# Disparities (max-min) and EOD
print("\nFairness disparities (max-min):")
print(fairness_metrics.difference())
eod = max(
    fairness_metrics.by_group["true_positive_rate"].max() -
fairness_metrics.by_group["true_positive_rate"].min(),
    fairness_metrics.by_group["false_positive_rate"].max() -
fairness_metrics.by_group["false_positive_rate"].min(),
)
print(f"\nEqualised Odds Difference: {eod:.4f}")

# Group-wise ROC AUC
group_auc_scores = {}
for group in sens_test.unique():
    idx = sens_test == group
    group_auc_scores[group] = roc_auc_score(y_test[idx], y_proba[idx])

print("\nGroup-wise ROC AUC Scores:")
for group, auc_val in group_auc_scores.items():
    print(f"{group}: {auc_val:.4f}")
```

9.8 Analysis of RF model with EGR (full dataset)

```
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, roc_auc_score, f1_score,
classification_report
from fairlearn.metrics import MetricFrame, selection_rate,
true_positive_rate, false_positive_rate
from fairlearn.reductions import ExponentiatedGradient, EqualizedOdds

RANDOM_STATE = 0

# Load and preprocess dataset
df = pd.read_csv("compas-scores-two-years.csv")
df["compas_screening_date"] = pd.to_datetime(df["compas_screening_date"],
errors="coerce")
df["out_custody"] = pd.to_datetime(df["out_custody"], errors="coerce")
df = df.dropna(subset=["compas_screening_date", "out_custody"])
df = df[
    (df["days_b_screening_arrest"] <= 30) &
    (df["days_b_screening_arrest"] >= -30) &
    (df["is_recid"] != -1) &
    (df["c_charge_degree"] != "O") &
    (df["score_text"] != "N/A")
].reset_index(drop=True)

# Features, target, and sensitive attribute
features = ["sex", "age", "race", "priors_count", "c_charge_degree"]
X = df[features]
y = df["two_year_recid"]
sens = df["race"]

# One-hot encode and split
X_enc = pd.get_dummies(X, columns=["sex", "race", "c_charge_degree"],
drop_first=True)
X_train, X_test, y_train, y_test, sens_train, sens_test =
train_test_split(
    X_enc, y, sens, test_size=0.30, stratify=y, random_state=RANDOM_STATE
)

# Base RF and EGR with Equalised Odds constraint
rf_base = RandomForestClassifier(
    n_estimators=500,
    random_state=RANDOM_STATE,
    n_jobs=1,         # keep runs deterministic
    bootstrap=True
)
egr = ExponentiatedGradient(estimator=rf_base,
constraints=EqualizedOdds())
egr.fit(X_train, y_train, sensitive_features=sens_train)

# Ensemble probabilities and fixed threshold
weights = np.array(egr.weights_, dtype=float)
```

```
weights = weights / weights.sum() if weights.sum() != 0 else weights
probs_list = [h.predict_proba(X_test)[:, 1] for h in egr.predictors_]
y_prob = np.sum([w * p for w, p in zip(weights, probs_list)], axis=0)
y_pred = (y_prob >= 0.5).astype(int)

# Performance metrics
print("Accuracy:", accuracy_score(y_test, y_pred))
print("ROC AUC:", roc_auc_score(y_test, y_prob))
print("F1 Score:", f1_score(y_test, y_pred))
print("\nClassification Report:\n", classification_report(y_test,
y_pred))

# Fairness metrics by race
metrics = {
    "selection_rate": selection_rate,
    "true_positive_rate": true_positive_rate,
    "false_positive_rate": false_positive_rate,
}
fair = MetricFrame(
    metrics=metrics,
    y_true=y_test,
    y_pred=y_pred,
    sensitive_features=sens_test
)
print("Fairness metrics by race:")
print(fair.by_group)

print("\nFairness disparities (max-min):")
print(fair.difference())

# Equalised Odds Difference
eod = max(
    fair.by_group["true_positive_rate"].max() -
fair.by_group["true_positive_rate"].min(),
    fair.by_group["false_positive_rate"].max() -
fair.by_group["false_positive_rate"].min()
)
print(f"\nEqualised Odds Difference: {eod:.4f}")

# Group-wise ROC AUC (using EGR ensemble probabilities)
group_auc = {}
for g in sens_test.unique():
    idx = sens_test == g
    group_auc[g] = roc_auc_score(y_test[idx], y_prob[idx])

print("\nGroup-wise ROC AUC Scores:")
for g, auc_val in group_auc.items():
    print(f"{g}: {auc_val:.4f}")
```

9.9 Analysis of RF model with threshold optimiser (full dataset)

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, roc_auc_score, f1_score,
classification_report
from fairlearn.metrics import MetricFrame, selection_rate,
true_positive_rate, false_positive_rate
from fairlearn.postprocessing import ThresholdOptimizer

RANDOM_STATE = 0

# Load and preprocess dataset
df = pd.read_csv("compas-scores-two-years.csv")
df["compas_screening_date"] = pd.to_datetime(df["compas_screening_date"],
errors="coerce")
df["out_custody"] = pd.to_datetime(df["out_custody"], errors="coerce")
df = df.dropna(subset=["compas_screening_date", "out_custody"])
df = df[
    (df["days_b_screening_arrest"] <= 30) &
    (df["days_b_screening_arrest"] >= -30) &
    (df["is_recid"] != -1) &
    (df["c_charge_degree"] != "O") &
    (df["score_text"] != "N/A")
].reset_index(drop=True)

# Features, target, and sensitive attribute
features = ["sex", "age", "race", "priors_count", "c_charge_degree"]
X = df[features]
y = df["two_year_recid"]
sens = df["race"]

# One-hot encode and split
X_enc = pd.get_dummies(X, columns=["sex", "race", "c_charge_degree"],
drop_first=True)
X_train, X_test, y_train, y_test, sens_train, sens_test =
train_test_split(
    X_enc, y, sens, test_size=0.30, stratify=y, random_state=RANDOM_STATE
)

# Train base RF
rf = RandomForestClassifier(random_state=RANDOM_STATE)
rf.fit(X_train, y_train)

# Base probabilities for ROC AUC (ranking unchanged by post-processing)
y_proba = rf.predict_proba(X_test)[:, 1]

# Threshold optimiser (equalised odds)
thresh_opt = ThresholdOptimizer(
    estimator=rf,
    constraints="equalized_odds",
    prefit=True
)
thresh_opt.fit(X_train, y_train, sensitive_features=sens_train)
```

```
# Post-processed predictions
y_pred = thresh_opt.predict(X_test, sensitive_features=sens_test)

# Performance metrics
print("Accuracy:", accuracy_score(y_test, y_pred))
print("ROC AUC:", roc_auc_score(y_test, y_proba))
print("F1 Score:", f1_score(y_test, y_pred))
print("\nClassification Report:\n", classification_report(y_test,
y_pred))

# Fairness metrics by race
metrics = {
    "selection_rate": selection_rate,
    "true_positive_rate": true_positive_rate,
    "false_positive_rate": false_positive_rate,
}
fairness_metrics = MetricFrame(
    metrics=metrics,
    y_true=y_test,
    y_pred=y_pred,
    sensitive_features=sens_test
)
print("Fairness metrics by race:")
print(fairness_metrics.by_group)

# Disparities (max-min) and EOD
print("\nFairness disparities (max-min):")
print(fairness_metrics.difference())
eod = max(
    fairness_metrics.by_group["true_positive_rate"].max() -
fairness_metrics.by_group["true_positive_rate"].min(),
    fairness_metrics.by_group["false_positive_rate"].max() -
fairness_metrics.by_group["false_positive_rate"].min()
)
print(f"\nEqualised Odds Difference: {eod:.4f}")

# Group-wise ROC AUC (using base probabilities)
group_auc = {}
for g in sens_test.unique():
    idx = sens_test == g
    group_auc[g] = roc_auc_score(y_test[idx], y_proba[idx])

print("\nGroup-wise ROC AUC Scores:")
for g, auc_val in group_auc.items():
    print(f"{g}: {auc_val:.4f}")
```

9.10 Analysis of baseline LR model (data-minimised dataset)

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, roc_auc_score, f1_score,
classification_report, roc_curve
from fairlearn.metrics import MetricFrame, selection_rate,
true_positive_rate, false_positive_rate

RANDOM_STATE = 0

# Load and preprocess dataset
df = pd.read_csv("compas-scores-two-years.csv")
df["compas_screening_date"] = pd.to_datetime(df["compas_screening_date"],
errors="coerce")
df["out_custody"] = pd.to_datetime(df["out_custody"], errors="coerce")
df = df.dropna(subset=["compas_screening_date", "out_custody"])
df = df[
    (df["days_b_screening_arrest"] <= 30) &
    (df["days_b_screening_arrest"] >= -30) &
    (df["is_recid"] != -1) &
    (df["c_charge_degree"] != "O") &
    (df["score_text"] != "N/A")
].reset_index(drop=True)

# Features, target, and sensitive attribute (race excluded from model
inputs)
features = ["sex", "age", "race", "priors_count", "c_charge_degree"]
X = df[features]
y = df["two_year_recid"]
sens = df["race"]

# One-hot encode inputs without race and split
X_enc = pd.get_dummies(X.drop(columns=["race"]), columns=["sex",
"c_charge_degree"], drop_first=True)
X_train, X_test, y_train, y_test, sens_train, sens_test =
train_test_split(
    X_enc, y, sens, test_size=0.30, stratify=y, random_state=RANDOM_STATE
)

# Train LR
lr = LogisticRegression(max_iter=1000, random_state=RANDOM_STATE)
lr.fit(X_train, y_train)

# Predictions and probabilities
y_pred = lr.predict(X_test)
y_prob = lr.predict_proba(X_test)[:, 1]

# Performance metrics
print("Accuracy:", accuracy_score(y_test, y_pred))
print("ROC AUC:", roc_auc_score(y_test, y_prob))
print("F1 Score:", f1_score(y_test, y_pred))
```

```
print("\nClassification Report:\n", classification_report(y_test,
y_pred))

# Fairness metrics by race
metrics = {
    "selection_rate": selection_rate,
    "true_positive_rate": true_positive_rate,
    "false_positive_rate": false_positive_rate,
}
fair = MetricFrame(
    metrics=metrics,
    y_true=y_test,
    y_pred=y_pred,
    sensitive_features=sens_test
)
print("Fairness metrics by race:")
print(fair.by_group)

# Disparities (max-min) and EOD
print("\nFairness disparities (max-min):")
print(fair.difference())
eod = max(
    fair.by_group["true_positive_rate"].max() -
fair.by_group["true_positive_rate"].min(),
    fair.by_group["false_positive_rate"].max() -
fair.by_group["false_positive_rate"].min()
)
print(f"\nEqualised Odds Difference: {eod:.4f}")

# Group-wise ROC AUC
group_auc = {}
for g in sens_test.unique():
    idx = sens_test == g
    group_auc[g] = roc_auc_score(y_test[idx], y_prob[idx])

print("\nGroup-wise ROC AUC Scores:")
for g, auc_val in group_auc.items():
    print(f"{g}: {auc_val:.4f}")

# ROC curve
fpr, tpr, _ = roc_curve(y_test, y_prob)
roc_auc = roc_auc_score(y_test, y_prob)
plt.figure(figsize=(6, 6))
plt.plot(fpr, tpr, lw=2, label=f"LR (AUC = {roc_auc:.3f})")
plt.plot([0, 1], [0, 1], "k--", lw=1, label="Chance")
plt.xlim(0, 1)
plt.ylim(0, 1.05)
plt.xlabel("False Positive Rate")
plt.ylabel("True Positive Rate")
plt.title("ROC curve - baseline LR (data-minimised)")
plt.legend(loc="lower right")
plt.tight_layout()
plt.show()

# Score distributions by race (KDE)
```

```
df_plot = pd.DataFrame({"race": sens_test.values, "score":
y_prob}).dropna()
race_order = ["African-American", "Caucasian", "Hispanic", "Other",
"Asian", "Native American"]
race_order = [r for r in race_order if r in df_plot["race"].unique()]

plt.figure(figsize=(12, 7))
for r in race_order:
    s = df_plot.loc[df_plot["race"] == r, "score"]
    if len(s) < 5:
        continue
    sns.kdeplot(s, label=r, fill=True, alpha=0.25, bw_adjust=1.0,
clip=(0, 1), common_norm=False)

plt.title("Prediction score distributions by race")
plt.xlabel("Predicted score")
plt.ylabel("Density")
plt.xlim(0, 1)
plt.legend(title="race")
plt.tight_layout()
plt.show()
```

9.11 Analysis of baseline RF model (data-minimised dataset)

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import (
    accuracy_score, roc_auc_score, f1_score, classification_report,
roc_curve
)
from fairlearn.metrics import MetricFrame, selection_rate,
true_positive_rate, false_positive_rate

RANDOM_STATE = 0

# Load and preprocess dataset
df = pd.read_csv("compas-scores-two-years.csv")
df["compas_screening_date"] = pd.to_datetime(df["compas_screening_date"],
errors="coerce")
df["out_custody"] = pd.to_datetime(df["out_custody"], errors="coerce")
df = df.dropna(subset=["compas_screening_date", "out_custody"])
df = df[
    (df["days_b_screening_arrest"] <= 30) &
    (df["days_b_screening_arrest"] >= -30) &
    (df["is_recid"] != -1) &
    (df["c_charge_degree"] != "O") &
    (df["score_text"] != "N/A")
].reset_index(drop=True)

# Features, target, and sensitive attribute (race excluded from model
inputs)
X = df[["sex", "age", "race", "priors_count", "c_charge_degree"]]
y = df["two_year_recid"]
sens = df["race"]

# One-hot encode inputs without race and split
X_enc = pd.get_dummies(X.drop(columns=["race"]), columns=["sex",
"c_charge_degree"], drop_first=True)
X_train, X_test, y_train, y_test, sens_train, sens_test =
train_test_split(
    X_enc, y, sens, test_size=0.30, stratify=y, random_state=RANDOM_STATE
)

# Train RF
rf = RandomForestClassifier(random_state=RANDOM_STATE)
rf.fit(X_train, y_train)

# Predictions and probabilities
y_pred = rf.predict(X_test)
y_proba = rf.predict_proba(X_test)[:, 1]

# Performance metrics
print("Accuracy:", accuracy_score(y_test, y_pred))
print("ROC AUC:", roc_auc_score(y_test, y_proba))
```

```
print("F1 Score:", f1_score(y_test, y_pred))
print("\nClassification Report:\n", classification_report(y_test,
y_pred))

# Fairness metrics by race
metrics = {
    "selection_rate": selection_rate,
    "true_positive_rate": true_positive_rate,
    "false_positive_rate": false_positive_rate,
}
fair = MetricFrame(
    metrics=metrics,
    y_true=y_test,
    y_pred=y_pred,
    sensitive_features=sens_test
)
print("Fairness metrics by race:")
print(fair.by_group)

# Disparities (max-min) and EOD
print("\nFairness disparities (max-min):")
print(fair.difference())
eod = max(
    fair.by_group["true_positive_rate"].max() -
fair.by_group["true_positive_rate"].min(),
    fair.by_group["false_positive_rate"].max() -
fair.by_group["false_positive_rate"].min()
)
print(f"\nEqualised Odds Difference: {eod:.4f}")

# Group-wise ROC AUC
group_auc = {}
for g in sens_test.unique():
    idx = sens_test == g
    group_auc[g] = roc_auc_score(y_test[idx], y_proba[idx])

print("\nGroup-wise ROC AUC Scores:")
for g, auc_val in group_auc.items():
    print(f"{g}: {auc_val:.4f}")

# ROC curve
fpr_rf, tpr_rf, _ = roc_curve(y_test, y_proba)
auc_rf = roc_auc_score(y_test, y_proba)
plt.figure(figsize=(6, 6))
plt.plot(fpr_rf, tpr_rf, label=f"RF (AUC = {auc_rf:.3f})")
plt.plot([0, 1], [0, 1], "k--", label="Chance")
plt.xlabel("False Positive Rate")
plt.ylabel("True Positive Rate")
plt.title("ROC curve - baseline RF (data-minimised)")
plt.legend(loc="lower right")
plt.tight_layout()
plt.show()

# Score distributions by race (KDE)
df_plot = pd.DataFrame({"race": sens_test.values, "score":
y_proba}).dropna()
```

```
race_order = ["African-American", "Caucasian", "Hispanic", "Other",
"Asian", "Native American"]
race_order = [r for r in race_order if r in df_plot["race"].unique()]

plt.figure(figsize=(12, 7))
for r in race_order:
    s = df_plot.loc[df_plot["race"] == r, "score"]
    if len(s) < 5:
        continue
    sns.kdeplot(s, label=r, fill=True, alpha=0.25, bw_adjust=1.0,
clip=(0, 1), common_norm=False)

plt.title("Prediction score distributions by race")
plt.xlabel("Predicted score")
plt.ylabel("Density")
plt.xlim(0, 1)
plt.legend(title="race")
plt.tight_layout()
plt.show()
```

9.12 Analysis of LR model with calibration (Platt scaling and isotonic regression, data-minimised dataset)

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.calibration import CalibratedClassifierCV
from sklearn.metrics import accuracy_score, roc_auc_score, f1_score,
classification_report
from fairlearn.metrics import MetricFrame, selection_rate,
true_positive_rate, false_positive_rate

RANDOM_STATE = 0

# Load and preprocess dataset
df = pd.read_csv("compas-scores-two-years.csv")
df["compas_screening_date"] = pd.to_datetime(df["compas_screening_date"],
errors="coerce")
df["out_custody"] = pd.to_datetime(df["out_custody"], errors="coerce")
df = df.dropna(subset=["compas_screening_date", "out_custody"])
df = df[
    (df["days_b_screening_arrest"] <= 30) &
    (df["days_b_screening_arrest"] >= -30) &
    (df["is_recid"] != -1) &
    (df["c_charge_degree"] != "O") &
    (df["score_text"] != "N/A")
].reset_index(drop=True)

# Features, target, and sensitive attribute (race excluded from model
inputs)
X = df[["sex", "age", "race", "priors_count", "c_charge_degree"]]
y = df["two_year_recid"]
sens = df["race"]

# One-hot encode inputs without race and split
X_enc = pd.get_dummies(X.drop(columns=["race"]), columns=["sex",
"c_charge_degree"], drop_first=True)
X_train, X_test, y_train, y_test, sens_train, sens_test =
train_test_split(
    X_enc, y, sens, test_size=0.30, stratify=y, random_state=RANDOM_STATE
)

# Base LR and probability calibration
# Use method='sigmoid' for Platt scaling or method='isotonic' for
isotonic regression.
base_lr = LogisticRegression(max_iter=1000, random_state=RANDOM_STATE)
calibrated_lr = CalibratedClassifierCV(estimator=base_lr,
method="isotonic", cv=5)
calibrated_lr.fit(X_train, y_train)

# Predictions and calibrated probabilities
y_pred = calibrated_lr.predict(X_test)
y_prob = calibrated_lr.predict_proba(X_test)[:, 1]
```

```
# Performance metrics
print("Accuracy:", accuracy_score(y_test, y_pred))
print("ROC AUC:", roc_auc_score(y_test, y_prob))
print("F1 Score:", f1_score(y_test, y_pred))
print("\nClassification Report:\n", classification_report(y_test,
y_pred))

# Fairness metrics by race
metrics = {
    "selection_rate": selection_rate,
    "true_positive_rate": true_positive_rate,
    "false_positive_rate": false_positive_rate,
}
fair = MetricFrame(
    metrics=metrics,
    y_true=y_test,
    y_pred=y_pred,
    sensitive_features=sens_test
)
print("Fairness metrics by race:")
print(fair.by_group)

# Disparities (max-min) and EOD
print("\nFairness disparities (max-min):")
print(fair.difference())
eod = max(
    fair.by_group["true_positive_rate"].max() -
fair.by_group["true_positive_rate"].min(),
    fair.by_group["false_positive_rate"].max() -
fair.by_group["false_positive_rate"].min()
)
print(f"\nEqualised Odds Difference: {eod:.4f}")

# Group-wise ROC AUC
group_auc = {}
for g in sens_test.unique():
    idx = sens_test == g
    group_auc[g] = roc_auc_score(y_test[idx], y_prob[idx])

print("\nGroup-wise ROC AUC Scores:")
for g, auc_val in group_auc.items():
    print(f"{g}: {auc_val:.4f}")
```

9.13 Analysis of LR with threshold optimiser (data-minimised dataset)

```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import (
    accuracy_score, roc_auc_score, f1_score, classification_report
)
from fairlearn.metrics import MetricFrame, selection_rate,
true_positive_rate, false_positive_rate
from fairlearn.postprocessing import ThresholdOptimizer
import matplotlib.pyplot as plt

RANDOM_STATE = 0

# Load COMPAS dataset
df = pd.read_csv("compas-scores-two-years.csv")

# Convert dates and remove rows with missing or invalid values
df["compas_screening_date"] = pd.to_datetime(df["compas_screening_date"],
errors='coerce')
df["out_custody"] = pd.to_datetime(df["out_custody"], errors='coerce')
df = df.dropna(subset=["compas_screening_date", "out_custody"])
df = df[
    (df["days_b_screening_arrest"] <= 30) &
    (df["days_b_screening_arrest"] >= -30) &
    (df["is_recid"] != -1) &
    (df["c_charge_degree"] != "O") &
    (df["score_text"] != "N/A")
].reset_index(drop=True)

# Select features and target
features = ["sex", "age", "race", "priors_count", "c_charge_degree"]
X = df[features]
y = df["two_year_recid"]

# Store sensitive attribute separately
sensitive_attr = df["race"]

# One-hot encode features, excluding race from training inputs
X_encoded = pd.get_dummies(X.drop(columns=["race"]), columns=["sex",
"c_charge_degree"], drop_first=True)

# Train/test split
X_train, X_test, y_train, y_test, sens_train, sens_test =
train_test_split(
    X_encoded, y, sensitive_attr, test_size=0.3, stratify=y,
random_state=RANDOM_STATE
)

# Train logistic regression model
lr = LogisticRegression(max_iter=1000, random_state=RANDOM_STATE)
lr.fit(X_train, y_train)
```

```
# Apply threshold optimisation with Equalized Odds constraint
threshold_op = ThresholdOptimizer(
    estimator=lr,
    constraints="equalized_odds",
    predict_method="predict_proba",
    prefit=True
)
threshold_op.fit(X_train, y_train, sensitive_features=sens_train)

# Predictions and probabilities
y_pred = threshold_op.predict(X_test, sensitive_features=sens_test)
y_prob = lr.predict_proba(X_test)[:, 1]

# Performance metrics
print("Accuracy:", accuracy_score(y_test, y_pred))
print("ROC AUC:", roc_auc_score(y_test, y_prob))
print("F1 Score:", f1_score(y_test, y_pred))
print("\nClassification Report:\n", classification_report(y_test,
y_pred))

# Fairness metrics by race
metrics = {
    'selection_rate': selection_rate,
    'true_positive_rate': true_positive_rate,
    'false_positive_rate': false_positive_rate,
}
fairness_metrics = MetricFrame(
    metrics=metrics,
    y_true=y_test,
    y_pred=y_pred,
    sensitive_features=sens_test
)
print("Fairness Metrics by Race:")
display(fairness_metrics.by_group)

# Disparities and Equalised Odds Difference
print("\nFairness Disparities (Max-Min):")
display(fairness_metrics.difference())
equalised_odds_diff = max(
    fairness_metrics.by_group['true_positive_rate'].max() -
fairness_metrics.by_group['true_positive_rate'].min(),
    fairness_metrics.by_group['false_positive_rate'].max() -
fairness_metrics.by_group['false_positive_rate'].min()
)
print(f"\nEqualised Odds Difference: {equalised_odds_diff:.4f}")

# Group-wise ROC AUC
group_auc_scores = {}
for group in sens_test.unique():
    group_idx = sens_test == group
    group_auc_scores[group] = roc_auc_score(y_test[group_idx],
y_prob[group_idx])
print("\nGroup-wise ROC AUC Scores:")
for group, auc in group_auc_scores.items():
    print(f"{group}: {auc:.4f}")
```

9.14 Analysis of RF model with calibration (Platt scaling and isotonic regression, data-minimised dataset)

```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.calibration import CalibratedClassifierCV
from sklearn.metrics import (
    accuracy_score, roc_auc_score, f1_score, classification_report
)
from fairlearn.metrics import MetricFrame, selection_rate,
true_positive_rate, false_positive_rate
import matplotlib.pyplot as plt

RANDOM_STATE = 0

# Load COMPAS dataset
df = pd.read_csv("compas-scores-two-years.csv")

# Convert dates and remove rows with missing or invalid values
df["compas_screening_date"] = pd.to_datetime(df["compas_screening_date"],
errors='coerce')
df["out_custody"] = pd.to_datetime(df["out_custody"], errors='coerce')
df = df.dropna(subset=["compas_screening_date", "out_custody"])
df = df[
    (df["days_b_screening_arrest"] <= 30) &
    (df["days_b_screening_arrest"] >= -30) &
    (df["is_recid"] != -1) &
    (df["c_charge_degree"] != "O") &
    (df["score_text"] != "N/A")
].reset_index(drop=True)

# Select features and target
X = df[["sex", "age", "race", "priors_count", "c_charge_degree"]]
y = df["two_year_recid"]

# Store sensitive attribute separately
sensitive_attr = df["race"]

# One-hot encode features, excluding race from training inputs
X_encoded = pd.get_dummies(X.drop(columns=["race"]), columns=["sex",
"c_charge_degree"], drop_first=True)

# Train/test split
X_train, X_test, y_train, y_test, sens_train, sens_test =
train_test_split(
    X_encoded, y, sensitive_attr, test_size=0.3, stratify=y,
random_state=RANDOM_STATE
)

# Define base model
base_rf = RandomForestClassifier(random_state=RANDOM_STATE)
```

```
# Apply calibration using cross-validation
# Use method='sigmoid' for Platt scaling or method='isotonic' for
isotonic regression.
calibrated_rf = CalibratedClassifierCV(estimator=base_rf,
method='sigmoid', cv=5)
calibrated_rf.fit(X_train, y_train)

# Predictions and calibrated probabilities
y_pred = calibrated_rf.predict(X_test)
y_proba = calibrated_rf.predict_proba(X_test)[:, 1]

# Performance metrics
print("Accuracy:", accuracy_score(y_test, y_pred))
print("ROC AUC:", roc_auc_score(y_test, y_proba))
print("F1 Score:", f1_score(y_test, y_pred))
print("\nClassification Report:\n", classification_report(y_test,
y_pred))

# Fairness metrics by race
metrics = {
    'selection_rate': selection_rate,
    'true_positive_rate': true_positive_rate,
    'false_positive_rate': false_positive_rate,
}
fairness_metrics = MetricFrame(
    metrics=metrics,
    y_true=y_test,
    y_pred=y_pred,
    sensitive_features=sens_test
)
print("Fairness Metrics by Race:")
display(fairness_metrics.by_group)

# Disparities and Equalised Odds Difference
print("\nFairness Disparities (Max-Min):")
display(fairness_metrics.difference())
equalised_odds_diff = max(
    fairness_metrics.by_group['true_positive_rate'].max() -
fairness_metrics.by_group['true_positive_rate'].min(),
    fairness_metrics.by_group['false_positive_rate'].max() -
fairness_metrics.by_group['false_positive_rate'].min()
)
print(f"\nEqualised Odds Difference: {equalised_odds_diff:.4f}")

# Group-wise ROC AUC
group_auc_scores = {}
for group in sens_test.unique():
    group_idx = sens_test == group
    group_auc_scores[group] = roc_auc_score(y_test[group_idx],
y_proba[group_idx])
print("\nGroup-wise ROC AUC Scores:")
for group, auc in group_auc_scores.items():
    print(f"{group}: {auc:.4f}")
```

9.15 Analysis of RF model with threshold optimiser (data-minimised dataset)

```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import (
    accuracy_score, roc_auc_score, f1_score, classification_report
)
from fairlearn.metrics import MetricFrame, selection_rate,
true_positive_rate, false_positive_rate
from fairlearn.postprocessing import ThresholdOptimizer
import matplotlib.pyplot as plt

RANDOM_STATE = 0

# Load COMPAS dataset
df = pd.read_csv("compas-scores-two-years.csv")

# Convert dates and remove rows with missing or invalid values
df["compas_screening_date"] = pd.to_datetime(df["compas_screening_date"],
errors='coerce')
df["out_custody"] = pd.to_datetime(df["out_custody"], errors='coerce')
df = df.dropna(subset=["compas_screening_date", "out_custody"])
df = df[
    (df["days_b_screening_arrest"] <= 30) &
    (df["days_b_screening_arrest"] >= -30) &
    (df["is_recid"] != -1) &
    (df["c_charge_degree"] != "O") &
    (df["score_text"] != "N/A")
].reset_index(drop=True)

# Select features and target
X = df[["sex", "age", "race", "priors_count", "c_charge_degree"]]
y = df["two_year_recid"]

# Store sensitive attribute separately
sensitive_attr = df["race"]

# One-hot encode features, excluding race from training inputs
X_encoded = pd.get_dummies(X.drop(columns=["race"]), columns=["sex",
"c_charge_degree"], drop_first=True)

# Train/test split
X_train, X_test, y_train, y_test, sens_train, sens_test =
train_test_split(
    X_encoded, y, sensitive_attr, test_size=0.3, stratify=y,
random_state=RANDOM_STATE
)

# Train base Random Forest model
rf_clf = RandomForestClassifier(random_state=RANDOM_STATE)
rf_clf.fit(X_train, y_train)

# Predicted probabilities for ROC AUC (before threshold optimisation)
```

```
y_proba = rf_clf.predict_proba(X_test)[:, 1]

# Apply threshold optimisation for Equalized Odds
threshold_opt = ThresholdOptimizer(
    estimator=rf_clf,
    constraints="equalized_odds",
    prefit=True,
    predict_method="predict_proba"
)
threshold_opt.fit(X_train, y_train, sensitive_features=sens_train)

# Predict using fairness-optimised thresholds
y_pred = threshold_opt.predict(X_test, sensitive_features=sens_test)

# Performance metrics
print("Accuracy:", accuracy_score(y_test, y_pred))
print("ROC AUC:", roc_auc_score(y_test, y_proba))
print("F1 Score:", f1_score(y_test, y_pred))
print("\nClassification Report:\n", classification_report(y_test,
y_pred))

# Fairness metrics by race
metrics = {
    'selection_rate': selection_rate,
    'true_positive_rate': true_positive_rate,
    'false_positive_rate': false_positive_rate,
}
fairness_metrics = MetricFrame(
    metrics=metrics,
    y_true=y_test,
    y_pred=y_pred,
    sensitive_features=sens_test
)
print("Fairness Metrics by Race:")
display(fairness_metrics.by_group)

# Disparities and Equalised Odds Difference
print("\nFairness Disparities (Max-Min):")
display(fairness_metrics.difference())
equalised_odds_diff = max(
    fairness_metrics.by_group['true_positive_rate'].max() -
fairness_metrics.by_group['true_positive_rate'].min(),
    fairness_metrics.by_group['false_positive_rate'].max() -
fairness_metrics.by_group['false_positive_rate'].min()
)
print(f"\nEqualised Odds Difference: {equalised_odds_diff:.4f}")

# Group-wise ROC AUC
group_auc_scores = {}
for group in sens_test.unique():
    group_idx = sens_test == group
    group_auc_scores[group] = roc_auc_score(y_test[group_idx],
y_proba[group_idx])
print("\nGroup-wise ROC AUC Scores:")
for group, auc in group_auc_scores.items():
    print(f"{group}: {auc:.4f}")
```

9.16 Analysis of LR model with reweighting, EGR, and threshold optimiser (full dataset)

```
import numpy as np
import pandas as pd
from IPython.display import display

from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.base import BaseEstimator, ClassifierMixin
from sklearn.metrics import accuracy_score, roc_auc_score, f1_score,
classification_report

from fairlearn.metrics import MetricFrame, selection_rate,
true_positive_rate, false_positive_rate
from fairlearn.reductions import ExponentiatedGradient, EqualizedOdds
from fairlearn.postprocessing import ThresholdOptimizer

RANDOM_STATE = 0
np.random.seed(RANDOM_STATE)

# Load and preprocess COMPAS
df = pd.read_csv("compas-scores-two-years.csv")
df["compas_screening_date"] = pd.to_datetime(df["compas_screening_date"],
errors="coerce")
df["out_custody"] = pd.to_datetime(df["out_custody"], errors="coerce")
df = df.dropna(subset=["compas_screening_date", "out_custody"])
df = df[
    (df["days_b_screening_arrest"] <= 30) &
    (df["days_b_screening_arrest"] >= -30) &
    (df["is_recid"] != -1) &
    (df["c_charge_degree"] != "O") &
    (df["score_text"] != "N/A")
].reset_index(drop=True)

# Features, target, and sensitive attribute
X = df[["sex", "age", "race", "priors_count", "c_charge_degree"]]
y = df["two_year_recid"].astype(int)
sens = df["race"]

# One-hot encode (keep race dummies for full-data setting)
X_enc = pd.get_dummies(X, columns=["sex", "race", "c_charge_degree"],
drop_first=True)

# Train/test split
X_tr, X_te, y_tr, y_te, sens_tr, sens_te = train_test_split(
    X_enc, y, sens, test_size=0.30, stratify=y, random_state=RANDOM_STATE
)

# Reweighting: compute P(r)P(y)/P(r,y) on train, then resample
train_df = pd.DataFrame({"label": y_tr.values, "race": sens_tr.values})
p_r = train_df["race"].value_counts(normalize=True)
p_y = train_df["label"].value_counts(normalize=True)
p_ry = train_df.groupby(["race", "label"]).size() / len(train_df)
```

```
eps = 1e-12

w = np.empty(len(train_df), dtype=float)
for i in range(len(train_df)):
    r = train_df.iloc[i]["race"]
    l = train_df.iloc[i]["label"]
    num = p_r.loc[r] * p_y.loc[l]
    den = p_ry.loc[(r, l)] if (r, l) in p_ry.index else eps
    w[i] = num / max(den, eps)

probs = w / w.sum()
idx = np.random.choice(np.arange(len(X_tr)), size=len(X_tr),
replace=True, p=probs)
X_tr_rw = X_tr.iloc[idx].reset_index(drop=True)
y_tr_rw = y_tr.iloc[idx].reset_index(drop=True)
sens_tr_rw = sens_tr.iloc[idx].reset_index(drop=True)

# EGR (Equalised Odds) with LR base estimator
base_lr = LogisticRegression(max_iter=1000, random_state=RANDOM_STATE,
solver="lbfgs")
constraint = EqualizedOdds()
egr = ExponentiatedGradient(estimator=base_lr, constraints=constraint)
egr.fit(X_tr_rw, y_tr_rw, sensitive_features=sens_tr_rw)

# Wrapper to expose predict_proba for the EGR ensemble
class EGRAveragedEstimator(BaseEstimator, ClassifierMixin):
    def __init__(self, expgrad_model):
        self.expgrad = expgrad_model

    def fit(self, X=None, y=None):
        self.is_fitted_ = True
        self.classes_ = np.array([0, 1])
        return self

    def predict_proba(self, X):
        probs = np.zeros((len(X), 2), dtype=float)
        for h, w in zip(self.expgrad.predictors_, self.expgrad.weights_):
            if hasattr(h, "predict_proba"):
                probs += w * h.predict_proba(X)
            else:
                scores = h.decision_function(X)
                p1 = 1.0 / (1.0 + np.exp(-scores))
                probs[:, 1] += w * p1
                probs[:, 0] += w * (1 - p1)
        sums = probs.sum(axis=1, keepdims=True)
        sums[sums == 0] = 1.0
        return probs / sums

    def predict(self, X):
        return (self.predict_proba(X)[:, 1] >= 0.5).astype(int)

wrapped_egr = EGRAveragedEstimator(egr).fit()

# Threshold optimiser (Equalised Odds) on top of EGR
thresh_opt = ThresholdOptimizer(
    estimator=wrapped_egr,
```

```
    constraints="equalized_odds",
    predict_method="predict_proba",
    prefit=True,
    flip=True
)
thresh_opt.fit(X_tr_rw, y_tr_rw, sensitive_features=sens_tr_rw)

# Predictions and scores
y_pred = thresh_opt.predict(X_te, sensitive_features=sens_te)
y_prob = wrapped_egr.predict_proba(X_te)[:, 1]

print("Accuracy:", accuracy_score(y_te, y_pred))
print("ROC AUC:", roc_auc_score(y_te, y_prob))
print("F1 Score:", f1_score(y_te, y_pred))
print("\nClassification Report:\n", classification_report(y_te, y_pred))

# Fairness metrics by race
metrics = {
    "selection_rate": selection_rate,
    "true_positive_rate": true_positive_rate,
    "false_positive_rate": false_positive_rate,
}
fair = MetricFrame(metrics=metrics, y_true=y_te, y_pred=y_pred,
sensitive_features=sens_te)
print("Fairness Metrics by Race:")
display(fair.by_group)

# Disparities and EOD
print("\nFairness Disparities (Max-Min):")
display(fair.difference())
eod = max(
    fair.by_group["true_positive_rate"].max() -
fair.by_group["true_positive_rate"].min(),
    fair.by_group["false_positive_rate"].max() -
fair.by_group["false_positive_rate"].min()
)
print(f"\nEqualised Odds Difference: {eod:.4f}")

# Group-wise ROC AUC
group_auc = {}
for g in sens_te.unique():
    idx = sens_te == g
    group_auc[g] = roc_auc_score(y_te[idx], y_prob[idx])
print("\nGroup-wise ROC AUC Scores:")
for g, auc in group_auc.items():
    print(f"{g}: {auc:.4f}")
```

9.17 Analysis of RF model with reweighting, EGR, and threshold optimiser (full dataset)

```
import numpy as np
import pandas as pd
from IPython.display import display

from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.base import BaseEstimator, ClassifierMixin
from sklearn.metrics import accuracy_score, roc_auc_score, f1_score,
classification_report

from fairlearn.metrics import MetricFrame, selection_rate,
true_positive_rate, false_positive_rate
from fairlearn.reductions import ExponentiatedGradient, EqualizedOdds
from fairlearn.postprocessing import ThresholdOptimizer

RANDOM_STATE = 0
np.random.seed(RANDOM_STATE)

# Load and preprocess COMPAS
df = pd.read_csv("compas-scores-two-years.csv")
df["compas_screening_date"] = pd.to_datetime(df["compas_screening_date"],
errors="coerce")
df["out_custody"] = pd.to_datetime(df["out_custody"], errors="coerce")
df = df.dropna(subset=["compas_screening_date", "out_custody"])
df = df[
    (df["days_b_screening_arrest"] <= 30) &
    (df["days_b_screening_arrest"] >= -30) &
    (df["is_recid"] != -1) &
    (df["c_charge_degree"] != "O") &
    (df["score_text"] != "N/A")
].reset_index(drop=True)

# Features, target, sensitive attribute
X = df[["sex", "age", "race", "priors_count", "c_charge_degree"]]
y = df["two_year_recid"].astype(int)
sens = df["race"]

# One-hot encode and split
X_enc = pd.get_dummies(X, columns=["sex", "race", "c_charge_degree"],
drop_first=True)
X_tr, X_te, y_tr, y_te, a_tr, a_te = train_test_split(
    X_enc, y, sens, test_size=0.30, stratify=y, random_state=RANDOM_STATE
)

# Reweighting via resampling: w = P(r)P(y)/P(r,y)
tmp = pd.DataFrame({"y": y_tr.values, "r": a_tr.values})
p_r = tmp["r"].value_counts(normalize=True)
p_y = tmp["y"].value_counts(normalize=True)
p_ry = tmp.groupby(["r", "y"]).size() / len(tmp)
eps = 1e-12
w = np.empty(len(tmp), dtype=float)
```

```
for i in range(len(tmp)):
    r = tmp.iloc[i]["r"]
    yy = tmp.iloc[i]["y"]
    w[i] = (p_r.loc[r] * p_y.loc[yy]) / max(p_ry.loc[(r, yy)] if (r, yy)
in p_ry.index else eps, eps)

prob = w / w.sum()
idx = np.random.choice(np.arange(len(X_tr)), size=len(X_tr),
replace=True, p=prob)
X_tr_rw = X_tr.iloc[idx].reset_index(drop=True)
y_tr_rw = y_tr.iloc[idx].reset_index(drop=True)
a_tr_rw = a_tr.iloc[idx].reset_index(drop=True)

# EGR with Equalised Odds using RF base estimator
rf_base = RandomForestClassifier(
    n_estimators=500,
    min_samples_leaf=2,
    random_state=RANDOM_STATE,
    n_jobs=-1
)
egr = ExponentiatedGradient(estimator=rf_base,
constraints=EqualizedOdds())
egr.fit(X_tr_rw, y_tr_rw, sensitive_features=a_tr_rw)

# Wrapper to expose averaged predict_proba from EGR ensemble
class EGRAveragedEstimator(BaseEstimator, ClassifierMixin):
    def __init__(self, egr_model):
        self.egr = egr_model

    def fit(self, X=None, y=None):
        self.is_fitted_ = True
        self.classes_ = np.array([0, 1])
        return self

    def predict_proba(self, X):
        probs = np.zeros((len(X), 2))
        for h, w in zip(self.egr.predictors_, self.egr.weights_):
            probs += w * h.predict_proba(X)
        row_sums = probs.sum(axis=1, keepdims=True)
        row_sums[row_sums == 0] = 1.0
        return probs / row_sums

    def predict(self, X):
        return (self.predict_proba(X)[:, 1] >= 0.5).astype(int)

wrapped = EGRAveragedEstimator(egr).fit()

# Threshold Optimizer (Equalised Odds)
thresh = ThresholdOptimizer(
    estimator=wrapped,
    constraints="equalized_odds",
    predict_method="predict_proba",
    prefit=True,
    flip=True
)
thresh.fit(X_tr_rw, y_tr_rw, sensitive_features=a_tr_rw)
```

```
# Predictions and scores
y_pred = thresh.predict(X_te, sensitive_features=a_te)
y_prob = wrapped.predict_proba(X_te)[:, 1]

print("Accuracy:", accuracy_score(y_te, y_pred))
print("ROC AUC:", roc_auc_score(y_te, y_prob))
print("F1 Score:", f1_score(y_te, y_pred))
print("\nClassification Report:\n", classification_report(y_te, y_pred))

# Fairness metrics by race
metrics = {
    "selection_rate": selection_rate,
    "true_positive_rate": true_positive_rate,
    "false_positive_rate": false_positive_rate,
}
fair = MetricFrame(metrics=metrics, y_true=y_te, y_pred=y_pred,
sensitive_features=a_te)
print("Fairness Metrics by Race:")
display(fair.by_group)

print("\nFairness Disparities (Max-Min):")
display(fair.difference())

# Equalised Odds Difference
eod = max(
    fair.by_group["true_positive_rate"].max() -
fair.by_group["true_positive_rate"].min(),
    fair.by_group["false_positive_rate"].max() -
fair.by_group["false_positive_rate"].min()
)
print(f"\nEqualised Odds Difference: {eod:.4f}")

# Group-wise ROC AUC (using wrapped EGR probabilities)
group_auc = {}
for g in a_te.unique():
    idx = a_te == g
    group_auc[g] = roc_auc_score(y_te[idx], y_prob[idx])
print("\nGroup-wise ROC AUC Scores:")
for g, auc_val in group_auc.items():
    print(f"{g}: {auc_val:.4f}")
```

9.18 Statement on AI usage

AI tools were used during the preparation of this thesis to provide support in several areas.

Firstly, AI was used for code debugging and optimisation, particularly the Python code relevant to Chapters 3-6. This code was used to train ML classifiers, apply bias mitigation methods, and compute evaluation metrics. AI's assistance involved resolving identified issues in the syntax, logic, or implementation. All suggestions made by AI were reviewed and validated before being included in this paper.

Secondly, AI was used to provide creative input throughout the thesis. This included suggesting potential names for figures and tables, offering alternative ways to phrase section headings, and proposing approaches for visualising models. This type of support was used whenever the author faced a creative roadblock. In these cases, AI only served as a source of inspiration and never made final decisions.

Thirdly, AI was consulted for general feedback on content completeness and coherence throughout each chapter. This included asking whether key elements might be missing, if certain explanations required greater clarity, or if the flow of ideas could be improved. Any feedback provided was considered solely as input for review. Suggestions were only addressed if deemed necessary by the author's own judgment and if they aligned with the thesis objectives. AI did not make final decisions or directly implement changes. Its role was advisory.

In all cases, the final decisions regarding structure, content, and presentation were made solely by the author.