People’s Democratic Republic of Algeria
Ministry of Higher Education and Scientific Research

University of 8 May 1945-Guelma-

Faculty of Mathematics, Computer Science and Science of Matter
Department of Computer Science

Master Thesis

Specialty: Computer Science

Option:

Computer system

Theme

Bias Mitigation in Healthcare-Based Machine Learning Systems

Presented by:

Hana Bordjiba

Supervised by:

Dr Djalila Boughareb

Jury Members:
Dr. Karima Benhamza, President
Dr. Leila Madi, Examiner

Acknowledgments

First and foremost, I would like to express my deepest gratitude to "Allah" for granting me strength, guidance, and perseverance throughout my academic journey.

I am incredibly grateful to my supervisor, Dr. Djalila Boughareb, for agreeing to be my supervisor and for her invaluable supervision and guidance throughout this research.

I would like to express my sincere appreciation. to the members of the jury for their time, expertise, and constructive feedback.

I want to express my heartfelt thanks to my parents, whose unwavering support and love have been the cornerstone of my success. To my family, thank you for your constant encouragement and for being my rock through this academic journey. I am also deeply grateful to my friends, mentors, and everyone who has accompanied me along the way. Your guidance, motivation, and belief in me have been invaluable. This achievement would not have been possible without each of you, and I am profoundly thankful for your presence in my life.

Dedications

I want to extend my deepest gratitude to my parents. Your unwavering support, endless patience, and unconditional love have been the foundation of my journey. Your encouragement has been my guiding light, and your sacrifices have made this achievement possible. I am profoundly thankful for everything you have done to help me reach this milestone.

To my cherished brothers "Otman" and "Abdallah" Your unwavering encouragement and steadfast support have been my anchor throughout this academic journey. Thank you for always being there in every way possible, guiding me with your wisdom and offering invaluable advice that transcends both academia and life itself.

To my partner in crime my brother "Skander", Thank you for being my constant companion on this journey, both figuratively and literally. Whether it was burning the midnight oil with me or chauffeuring me around like my very own personal Uber driver, you've been there every step of the way. Your support, patience, and unending supply of snacks made this possible.

To my dear auntie and her family, your unwavering love and support have been my foundation throughout this journey. I am forever grateful for the countless ways you have enriched my life.

To my little Princess "Alia", your laughter has been my constant source of energy and motivation throughout this journey.

To my best friend "Lyna", who is stuck with me through thick and thin, believed in me when I doubted myself, and somehow managed to make me laugh even in my bad days.

To all my familly and friends who have accompanied me on this academic journey.

Abstract

This study investigates the impact of oversampling techniques, specifically SVM-SMOTE and BorderlineSMOTE, on machine learning models for heart disease and diabetes risk prediction. Using Gradient Boosting Machine (GBM) and K-Nearest Neighbors (KNN) algorithms, we assess changes in accuracy, precision, recall, F1 score, Positive Predictive Value (PPV), Equal Opportunity Difference (EOD), Disparate Impact (DI), and Impact Ratio (IR) across diverse biomedical datasets.

In both heart disease and diabetes risk prediction tasks, SVM-SMOTE and BorderlineSMOTE proved effective in enhancing machine learning model performance. For heart disease prediction, SVM-SMOTE and BorderlineSMOTE improved GBM model accuracy to 0.85 and 0.85 from an initial 0.74, precision to 0.79 and 0.77 from 0.69, and recall to 0.88 and 0.92 from 0.75, respectively. KNN models also showed enhancements in accuracy (0.71 from 0.68), precision (0.70 from 0.59), and recall (0.72 from 0.62). In diabetes risk prediction, both techniques consistently boosted accuracy, precision, and F1 score metrics across GBM and KNN models. Notably, DI values improved significantly to 1.11 with both SVM-SMOTE and BorderlineSMOTE from an initial 0.43, indicating improved fairness in model predictions across demographic groups.

Overall, the strategic application of SVM-SMOTE and BorderlineSMOTE effectively addresses class imbalance challenges in biomedical datasets, enhancing both predictive accuracy and fairness in machine learning models. These results underscore the importance of tailored oversampling techniques in achieving robust and equitable healthcare predictions across diverse demographic groups.

Keywords: Bias, Unfairness, Mitigation, Healthcare, GBM, KNN, SVMSMOTE, BorderlineSMOTE

Résumé

Cette étude examine l'impact des techniques de suréchantillonnage, spécifiquement SVM-SMOTE et BorderlineSMOTE, sur les modèles d'apprentissage automatique pour la prédiction des maladies cardiaques et du risque de diabète. En utilisant les algorithmes Gradient Boosting Machine (GBM) et K-Nearest Neighbors (KNN), nous évaluons les changements dans les mesures d'exactitude, de précision, de rappel, de score F1, de valeur prédictive positive (PPV), de différence d'opportunité équitable (EOD), d'impact disparate (DI), et de ratio d'impact (IR) à travers divers ensembles de données biomédicales.

Dans les tâches de prédiction des maladies cardiaques et du risque de diabète, SVM-SMOTE et BorderlineSMOTE se sont avérés efficaces pour améliorer les performances des modèles d'apprentissage automatique. Pour la prédiction des maladies cardiaques, SVM-SMOTE et BorderlineSMOTE ont amélioré l'exactitude du modèle GBM à 0,85 et 0,85 à partir de 0,74 initial, la précision à 0,79 et 0,77 à partir de 0,69, et le rappel à 0,88 et 0,92 à partir de 0,75, respectivement. Les modèles KNN ont également montré des améliorations en termes d'exactitude (0,71 contre 0,68), de précision (0,70 contre 0,59), et de rappel (0,72 contre 0,62). Pour la prédiction du risque de diabète, les deux techniques ont régulièrement augmenté les mesures d'exactitude, de précision et de score F1 à travers les modèles GBM et KNN. Notamment, les valeurs de DI ont significativement augmenté à 1,11 avec SVM-SMOTE et BorderlineSMOTE à partir d'un initial de 0,43, indiquant une amélioration de l'équité dans les prédictions des modèles à travers les groupes démographiques.

Dans l'ensemble, l'application stratégique de SVM-SMOTE et BorderlineSMOTE adresse efficacement les défis liés aux déséquilibres de classes dans les ensembles de données biomédicales, améliorant à la fois l'exactitude prédictive et l'équité dans les modèles d'apprentissage automatique. Ces résultats soulignent l'importance des techniques de suréchantillonnage adaptées pour atteindre des prédictions de soins de santé robustes et équitables à travers des groupes démographiques diversifiés.

Mots clés : Biais, Injustice, Atténuation, Biomédical, GBM, KNN, SVMSMOTE, BorderlineSMOTE.

Contents

List of Tables

List of Figures

General introduction

Machine learning (ML) has revolutionized fields like healthcare by enabling data-driven insights and decision-making. However, alongside its benefits, ML models can unintentionally introduce biases that perpetuate inequalities. Bias in ML refers to systematic inaccuracies in predictions, often reflecting disparities in training data. Detecting and mitigating these biases is crucial for ensuring fair and trustworthy AI applications, particularly in sensitive domains like healthcare. This project focuses on developing strategies to identify and address bias in ML models applied to biomedical data, aiming to promote equitable and effective AI deployment in healthcare.

The problematic of this master thesis centers on the inherent biases that machine learning models can perpetuate when applied to real-world datasets. In many cases, these biases stem from uneven representation of demographic groups, such as gender or race, within the data. Additionally, biases can arise due to historical inequalities embedded in societal systems or errors in data collection processes. Left unchecked, these biases have the potential to lead to unfair outcomes, especially in critical domains like healthcare where accurate and equitable decision-making is paramount. Addressing these challenges is essential not only for ensuring the reliability and effectiveness of machine learning applications but also for upholding ethical standards and promoting trust in AI-driven solutions.

Our objective is to enhance the fairness and reliability of machine learning models applied to biomedical data by implementing effective bias mitigation strategies. This involves preprocessing the data to ensure suitability for training two different Models, GBM and KNN. To address class imbalance, we utilize SVM-SMOTE and Borderline SMOTE for oversampling. We focus on detecting and mitigating biases across demographic groups such as gender and race using metrics like disparate impact and statistical parity. By constructing model ensembles and refining them through post-processing techniques, we aim to improve overall model performance and fairness, particularly in critical applications like healthcare.

The core of this Master thesis comprises three main chapters. Chapter 1 offers a comprehensive overview of Bias and Unfairness Mitigation in Machine Learning Models. It covers fundamental definitions, types of bias, various fairness metrics, specific characteristics pertinent to biomedical models, challenges associated with bias mitigation in biomedical-based machine learning models, and techniques tailored for bias mitigation in machine learning models applied to biomedical data.

Chapter 2 details the methodology of our proposed architecture aimed at reducing bias in biomedical datasets using a machine learning approach. It includes a description of the dataset utilized for both training and evaluation purposes, outlines the preprocessing steps undertaken, and evaluates the performance metrics before and after the implementation of bias mitigation techniques.

Chapter 3 focuses on the implementation phase of our proposed bias reduction system for biomedical datasets. This chapter details the practical execution of the methodology outlined in Chapter 2. It covers the actual application of machine learning algorithms, the integration of bias mitigation techniques, and the execution of experiments to validate the effectiveness of our approach

Chapter 1: Bias and Unfairness in Healthcare Models

1.1 Introduction

In the realm of machine learning, the omnipresent challenge of bias and unfairness looms large, casting shadows over the integrity and reliability of algorithmic decision-making. Defined as systematic errors or preferences, bias infiltrates data, algorithms, and user interactions, manifesting in disparities that undermine the principles of equality and justice. Within the biomedical domain, where machine learning holds profound potential for transformative healthcare advancements, the stakes of bias mitigation are particularly high. From data-driven approaches to clinical implementations, biomedical models navigate a complex landscape rife with challenges, from data bias to privacy concerns. In this chapter, we embark on an exploration of the multifaceted nature of bias, examining its various types, metrics for evaluation, and techniques for mitigation in the context of biomedical machine learning.

1.2 Definition of Bias

The concept of bias encompasses various definitions, which vary depending on the context, Bias can be defined as a systematic error or an inclination to favor one outcome over another unexpectedly [1]. The concept of bias is also applied to algorithms when they exhibit an undesired reliance on a particular attribute in the data, often associated with a demographic group. An unbiased algorithm ideally should not depend on any protected attributes of an individual, such as gender, race, or religion. When algorithmic bias results in differential treatment between patient groups, it can be deemed unfair from both legal and ethical perspectives [2].

1.3 Definition of Unfairness

Varied civilizations have varied definitions of injustice. Thus, the unfairness criterion is affected by user experience as well as cultural, social, historical, political, legal, and ethical factors. Social biases and statistical biases are the two main causes of unfairness.

The former refers to the difference between how the world should be and how it actuallyis, while the other type of bias refers to the difference between how the world is and howit is represented in the system [3].

1.4 Type of Bias

Bias can appear in various forms, some of which can result in unfairness across different learning tasks downstream. This bias can be categorized into three main types: data bias, algorithm bias, and user interaction bias.

1.4.1 Data Bias

When specific factors are omitted or human bias causes data to not correctly reflect the intended data, it is referred to as data bias. Ancestry, demography, socioeconomics, and methodological problems like quantifying illnesses or identifying treatment outcomes all contribute to this [W1].

1.4.2 Algorithmic Bias

A complicated concept to describe, algorithmic bias is a systematic error in computer systems that provides inaccurate data processing outcomes. It is not always an error; it is typically an algorithmic property that prevents it from being fair or objective. Alternatively, it could be described as a departure from a norm [4].

1.4.3 User interaction Bias

When a user applies self-selected prejudices and behaviors when interacting with data, output, outcomes, etc., user interaction bias can arise. The interface that exists between the automated system and the user can also cause it to develop [5].

1.5 Unfairness metrics

Several metrics can be used to measure unfairness. In this section, we describe the most common ones [6].

1.5.1 The Equalized odds

The Equalized odds (EO) given by equation (1.1) aims to ensure equal probability of positive and negative results for protected and unprotected groups for individuals in both classes [6].

$$EO = \frac{1}{2} * \left(\left|\frac{FP_P}{FP_P+TN_P} - \frac{FP_\mu}{FP_\mu+TN_\mu}\right| + \left|\frac{TP_P}{TP_P+FN_P} - \frac{TP_\mu}{TP_\mu+FN_\mu}\right|\right) \qquad (1.1)$$

1.5.2 The Equality of Opportunity

The measure Equal Of Opportunity (EOO) given by given by equation (1.2) provides that every individual in a binary classifier have an equal probability to get effective outcomes, which applies to both protected and unprotected groups [6].

$$EOO = \frac{TP_p}{TP_P+FN_p} - \frac{TP_u}{TP_u+FN_u} \qquad (1.2)$$

1.5.3 The Demographic Parity

Demographic Parity given by equation (1.3), formerly referred to as Statistical Parity, is a measure of fairness

that indicates the probability of a positive outcome [6].

$$DP = \frac{\mathrm{TP+FP}}{\mathrm{N}} \tag{1.3}$$

1.5.4 The Positive Predictive Value

The percentage of cases with positive test findings that are already patients is known as the positive predictive value (PPV) given by equation (1.4). It is the proportion of patients with valid diagnoses to all patients with positive test findings.

If a test is positive, this trait might indicate whether a person will actually be patient [36].

$$PPV = \frac{\mathrm{TP}}{\mathrm{TP+FP}} \tag{1.4}$$

1.5.5 The Equal Opportunity Difference

According to the author in [35], the Equal Opportunity Difference (EOD) given by equation (1.5) quantifies the disparity in true positive rates between the advantaged and disadvantaged groups.

$$\mathrm{EOD} = \left|TPR_{privileged} - TPR_{underprivileged}\right| \tag{1.5}$$

1.5.6 The Disparate Impact

The percentage of individuals who receive from both protected and unprotected groups is measured by the Disparate Impact (DI) given by equation (1.6). To be fair, it must equal 1 [6].

$$DI = \frac{\frac{TP_p+FP_p}{N_p}}{\frac{TP_u+FP_u}{N_u}} \tag{1.6}$$

1.5.7 The K-Nearest Neighbors Consistency

The similarity of sensitive attribute labels for similar instances is measured by the K-Nearest Neighbors Consistency (KNNC) fairness metric given by equation (1.7) [6].

$$\mathrm{KNNC} = 1 - \frac{1}{\mathrm{n}}\sum_{\mathrm{i}=1}^{\mathrm{n}} \left|\hat{\mathrm{y}}_{\mathrm{i}} - \frac{1}{\mathrm{k}}\sum_{\mathrm{j}\in \mathrm{N}_{\mathrm{k}}(\mathrm{x}_{\mathrm{i}})} \hat{\mathrm{y}}_{\mathrm{j}}\right| \qquad (1.7)$$

1.5.8 Absolute Balanced Accuracy Difference

The Absolute Balanced Accuracy Difference (ABAD) given by equation (1.8) is the difference in balanced accuracy in protected and unprotected groups, defined by Equation [6].

$$\mathrm{ABAD} = \left|\frac{1}{2}\left[TPR_p + TNR_p\right] - \left[TPR_u + TNR_u\right]\right| \qquad (1.8)$$

1.5.9 Absolute Average Odds Difference

The Absolute Average Odds Difference (AAOD) given by equation (1.9) is the absolute difference in TPR and FPR between different protected groups, defined by Equation [6].

$$\mathrm{AAOD} = \left|\frac{(FPR_u+FNR_p)-(TPR_u+TPR_p)}{2}\right| \qquad (1.9)$$

1.5.10 Absolute Equal Opportunity Rate Difference

Absolute Equal Opportunity Rate Difference (AEORD) given by equation (1.10) is the difference in recall scores (TPR) between the protected and unprotected groups. A value of 0 indicates equality of opportunity, defined by Equation [6].

$$AEORD = |TPR_p - TPR_u| \qquad (1.10)$$

1.5.11 Statistical Parity Difference

Statistical Parity Difference (SPD) given by equation (1.11) is the difference in SD between a protected and an unprotected group, defined by Equation [6].

$$\mathrm{SPD} = \frac{TP_P+FP_p}{N_p} - \frac{TP_u+FP_u}{N_u} \qquad (1.11)$$

1.5.12 Imbalance Ratio

The imbalance ratio (IR) given by equation (1.12) serves as a commonly employed measure for assessing the degree of class imbalance within a dataset. It is computed by dividing the sample size of the majority class by that of the minority class. In scenarios with multiple classes, Nmaj represents the sample size of the largest majority class, while Nmin denotes the sample size of the smallest minority class [47].

$$\mathrm{IR} = \frac{\mathrm{N_{maj}}}{\mathrm{N_{min}}} \qquad (1.12)$$

1.5.13 Confusion Matrix

The confusion matrix is a fundamental evaluation tool used to assess the performance of classification algorithms. It is applicable to both binary and multiclass classification tasks. Table 1.1 illustrates the confusion matrix.

		Predicted results	
		Positive (PP)	Negative (PN)
Actual Observations	Positive (P)	True Positive (TP)	False Negative (FN)
	Negative (N)	False Positive (FP)	True Negative (TN)

Table 1.1 Confusion matrix

1.5.14 Typical classification Metrics Used in Fairness Evaluation

Several metrics are used to measure the effectiveness of a ML model. The most common are: Accuracy, Precision, Recall, and F1-Score that can be calculated using equations (1.13-1.16) respectively.

Accuracy is a commonly used metric that assesses the ratio of correctly classified instances to the total number of instances.

$$Accurancy = \frac{TN+TP}{TN+TP+FN+FP} \qquad (1.13)$$

The percentage of true positive predictions out of all positive predictions is known as the precision. It can be calculated using the following formula:

$$Precision = \frac{TP}{TP+FP} \qquad (1.14)$$

Recall can be defined as the proportion of true positives with respect to all the positives that exist in the ground truth.

$$Recall = \frac{TP}{TP+FN} \qquad (1.15)$$

The F1-score is the harmonic mean of precision and recall, providing a single measure that balances both concerns.

$$F1 - score = \frac{2(recall \times Precision)}{recall+Precision} \qquad (1.16)$$

1.5.15 Other metrics

To comprehend fairness metrics reliant on true positives (TP), false positives (FP), true negatives (TN), and false negatives (FN), it is essential to establish the corresponding statistical measures as outlined in Table 1.2 [2].

Statistical Metrics	Equation
Positive Predictive Value (PPV)	PPV = TP/(TP + FP)

False Discovery Rate (FDR	FDR = FP/(TP + FP)
False Omission Rate (FOR)	FOR = FN/(TN + FN)
Negative Predictive Value (NPV)	NPV = TN/(TN + FN)
True Positive Rate (TPR)	TPR = TP/(TP + FN)
False Positive Rate (FPR)	FPR = FP/(FP + TN)
False Negative Rate (FNR)	FNR = FN/(TP + FN)
True Negative Rate (TNR)	TNR = TN/(FP + TN)

Table 1.2 Statistical Metrics

1.6 Characteristics of biomedical models

The characteristics of biomedical models discussed in this section stem from research on [7]. Biomedical models employ a data-driven approach, utilizing vast amounts of digital data such as electronic health records (EHRs), medical imaging scans, and genetic data to identify patterns and make predictions. They rely on machine-learning algorithms to analyze this data, uncovering patterns and relationships that may not be immediately apparent to human observers. In clinical settings, these models are essential tools, aiding healthcare professionals in tasks like case triage, diagnosis, decision-making, and risk prediction, necessitating rigorous validation and testing. Additionally, biomedical models focus on health science applications, enhancing patient care and outcomes through applications like EHRs, medical imaging, and genetic engineering. They handle both structured and unstructured data, ensuring accurate and efficient healthcare outcomes. Derived from successful real-world experiments, biomedical models have shown significant promise in improving healthcare delivery and patient outcomes in clinical settings.

1.7 Challenges of Bias Mitigation in Biomedical Based ML Models

The challenges in mitigating bias in biomedical machine learning models include data biases that favor certain demographics, measurement errors that lead to unreliable results, lack of contextual specificity across diverse populations, and ethical concerns regarding data privacy and algorithm fairness. Addressing these challenges is crucial for developing accurate, equitable, and ethically responsible AI applications in healthcare.

1. **Data Bias:** In healthcare, these biases in data can be more harmful. Many times, the data that is evaluated and used is biased toward particular demographics, which can have detrimental effects on the communities that are underrepresented [1].
2. **Measurement error bias:** Also known as, measurement bias is the discrepancy between observed and true values brought on by mistakes in self-reported measurements, laboratory factors, or equipment inaccuracy. These mistakes can arise in experimental and observational research settings, like in cardiovascular disease cohort studies. Ignoring measurement error, or naive analysis, can produce biased and misleading results, such as inconsistent regression parameter estimators and unacceptable conclusions about confidence intervals and hypothesis

testing [8].

3. **Lack of contextual specificity:** Diverse populations from different environments, cultures, and socioeconomic origins are served by health systems.
   This diversity should serve as the foundation for the development of a general AI model; however, data availability varies throughout groups. As a result of this imbalance, there is not enough information to make reliable projections for underrepresented groups, which emphasizes the need for more precise data collecting [9].
4. **Privacy and Ethical Concerns:** Data privacy is a major ethical concern, and big data research raises additional ethical issues with transparency, interpretability, and algorithmic impartiality [10].

1.8 Techniques of bias mitigation in ML models

Preprocessing, in-processing, and post processing are the three categories into which mathematical methods for mitigating bias—algorithmic debiasing—can be divided based on where they are used. When we talk about bias in these methods, we usually mean the statistical relationship between a protected attribute and predicted outcome [11].

1.8.1 Preprocessing

Preprocessing refers to the set of operations and techniques applied to data before it is used for modeling or analysis. Its primary objectives include data cleaning to remove noise and inconsistencies, data transformation to enhance interpretability or prepare features for modeling, and data augmentation to increase the volume or diversity of data available for training [3]. Preprocessing activities include:

- **Removing Sensitive Data:** Ensure that all sensitive information, such as customer data in a banking system, is removed or anonymized. This is essential to comply with privacy regulations, protecting individuals' privacy and preventing unauthorized access.
- **Generating Synthetic Data:** Create synthetic data from the original dataset to preserve privacy while maintaining data utility. Tools like the Trusted Model Executor (TME) from AIF360 can be used to generate synthetic datasets that mimic the statistical properties of the original data without revealing sensitive information.
- **Data Balancing:** Address class imbalance in datasets using techniques such as reweighting or methods like SMOTE (Synthetic Minority Oversampling Technique). These approaches are crucial for ensuring that machine learning models are trained accurately and perform well across all classes.
- **Mitigating Bias:** Implement various techniques to reduce biases in datasets. For instance, use Generative Adversarial Networks (GANs) for style transfer to increase demographic diversity in facial image datasets. This helps create more balanced and representative datasets, leading to fairer and more equitable model outcomes.

1.8.2 In-processing

In-processing in the context of machine learning refers to techniques and interventions applied during the training phase of a model to mitigate bias and improve fairness. Unlike preprocessing, which alters the data before training, in-processing modifies the learning algorithm itself or its parameters to address biases directly while the model is being trained. This approach ensures that the adjustments are made within the model's learning process, allowing it to learn in a way that inherently reduces the impact of sensitive attributes such as race, gender, or income on its predictions [3]. In-processing methods include:

- **Algorithm Adjustment:** Modifying the algorithm during training to achieve a balance between accuracy and fairness. For example, using Pareto Optimal solutions to maintain fairness metrics while minimizing accuracy loss.
- **Sensitive Attribute Neutralization:** Techniques that reduce a model's dependence on sensitive attributes by adjusting weights or introducing layers that counteract the influence of these attributes during training.
- **Adversarial Learning:** Incorporating adversarial networks that work against the main model to detect and reduce biases related to sensitive attributes. This method can involve adding adversarial layers to predict and mitigate the influence of sensitive attributes during model learning.
- **Balancing Methods:** Applying strategies such as reweighting or modifying loss functions to ensure that the learning process does not favor any particular group, thereby maintaining fairness across diverse populations.
- **Hybrid Techniques:** Combining in-processing with other fairness methods, such as using adversarial techniques in conjunction with decision trees or employing privileged information to train models while respecting sensitive data.

1.8.3 Post-processing

Post-processing in machine learning refers to techniques applied after a model has been trained to detect and mitigate biases in its predictions. Unlike preprocessing, which modifies the data before training, or in-processing, which adjusts the algorithm during training, post-processing focuses on altering the model's outputs to ensure fairness and reduce discrimination without changing the model itself or its training process [3].

- **Bias Detection and Adjustment:** Identifying unfair treatment of certain groups based on sensitive attributes (like race or gender) and adjusting the model's outputs to mitigate these biases.
- **Model Output Modification:** Altering decision thresholds or reclassifying outputs to reduce bias, such as adjusting a neural network's weights to minimize discrepancies between different groups.
- **Black-Box Model Mitigation:** Implementing fairness interventions without needing to understand or change the internal workings of complex models, making it suitable for deep learning models.
- **Bias Analysis in Predictions:** Using tools like Natural Language Processing (NLP) to detect and understand biases in the model's decisions, ensuring predictions are fair despite variations in data quality or types.

1.9 OVERSAMLING

Oversampling in machine learning is a data-level approach used to address class imbalance. It involves generating new data by replicating important samples of the minority class. This helps to increase the representation of the minority class in the dataset, ensuring that the machine learning model receives a balanced view of all classes during training [48].

This Figure 1.1 visually explains how oversampling works to address imbalances in datasets

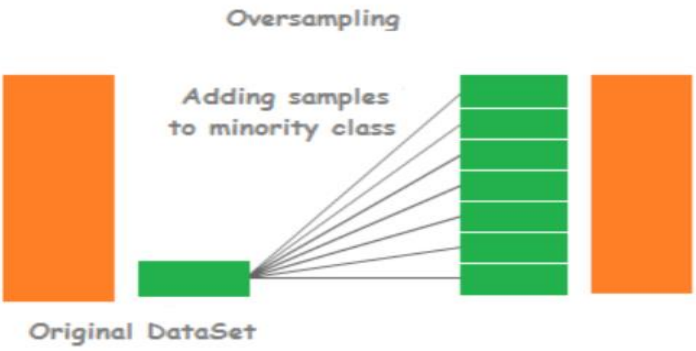


Figure 1.1 Illustration of Oversampling in Data Processing [50].

More advanced oversampling approaches, such as Synthetic Minority Oversampling Technique (SMOTE), Support Vector Machine SMOTE (SVM-SMOTE), and Borderline-SMOTE, have been developed in addition to basic oversampling techniques to produce synthetic samples in a more efficient and focused way.

1.9.1 SMOTE

Synthetic Minority Oversampling Technique (SMOTE), an over-sampling technique introduced in 2002, enhances the minority class by generating synthetic instances instead of simply duplicating existing ones. The process starts by selecting a sample from the minority class and finding its five nearest neighbors. If a 200% over-sampling is needed, two of these neighbors are chosen. A synthetic instance is then created by interpolating along the line between the selected neighbors and the original sample [37]. The following Pseudo-code details the SMOTE algorithm [49].

Algorithm SMOTE

Begin

Input:

- Minority class samples (X_minority)
- Number of synthetic samples to generate (N)
- Number of nearest neighbors to consider (k)

Output:

- Synthetic samples (X_synthetic)

```
Procedure SMOTE(X_minority, N, k):
X_synthetic = [ ]
for i 1 to N do:
random_sample = randomly select a sample from X_minority
```

```
5.  nearest_neighbors = find k-nearest neighbors of random_sample in X_ minority
    # randomly select one of the nearest neighbors
6.  neighbor = randomly select a neighbor from nearest_neighbors
    # Generate synthetic sample
7.  synthetic_sample = random_sample + random () * (neighbor - random_sample)
8.  X_synthetic.append(synthetic_sample)
9.  return X_synthetic
10. Stop algorithm
11. End.
```

X_minority: Input data containing samples from the minority class.
N: Number of synthetic samples to generate.
k: Number of nearest neighbors to consider.

1.9.2 SVMSMOTE

The Support Vector Machine SMOTE (SVM SMOTE) is a boundary-defining technique that uses SVM separation formulas in conjunction with extrapolation and interpolation. SVM SMOTE employs support vectors in place of the k-neighborhood computational interpolation used in the SMOTE technique. This allows for additional polarization for the minority class and interpolation for the majority class, resulting in a balanced sample. Each minority class support vector will have synthetic data generated at random along the lines connecting it with some of its closest neighbors and the following pseudo-code details the SVMSMOTE technique [49].

Algorithm SVM-SMOTE

Begin

Input:

- X_train: Feature matrix of the training set
- Y_train: Corresponding labels of the training set
- n_neighbors: Number of nearest neighbors to consider in SMOTE
- svm_kernel: Kernel function for SVM (e.g., linear, RBF)
- svm_C: Penalty parameter for SVM
- over_sampling_ratio: Ratio of over-sampling for the minority class

Output:

- X_resampled: Resampled feature matrix
- Y_resampled: Corresponding resampled labels

```
1. Apply SVM to the original imbalanced dataset to train a classification model.
2. Identify the minority class samples.
3. For each minority class sample (x_i, y_i):
4. Find its n_neighbors nearest neighbors within the same class.
5. Apply SMOTE to generate synthetic samples, considering n_neighbors and
   over_sampling_ratio.
6. Add the synthetic samples to the list of synthetic samples.
```

```
7. Combine the original training data with the synthetic samples:
8. Append the synthetic samples to the original feature matrix (X_train) and their
   corresponding labels to Y_train.
9. Return the resampled feature matrix (X_resampled) and corresponding labels
   (Y_resampled).
10. Stop algorithm
End.
```

1.9.3 BorderlineSMOTE

A version of the SMOTE algorithm called BorderlineSMOTE was created to reduce some of its flaws. For instance, SMOTE may produce synthetic instances utilizing majority class samples when minority class samples are outliers within the majority class, which might result in inaccurate results. This problem is reduced by BorderlineSMOTE, which removes any minority sample that is completely surrounded by samples from the majority class as noise and does not use them in the creation of synthetic instances. Additionally, it recognizes some samples as border points, whose neighbors are members of the majority and minority classes, and builds synthetic instances only from these border points [37].

In Borderline-SMOTE, instances in the minority class are categorized into three groups: NOISE, DANGER, and SAFE.

- NOISE instances are rare and likely incorrect, situated in regions dominated by majority class instances.
- DANGER instances are located near class boundaries, often overlapping with majority instances.
- SAFE instances are more easily identifiable and serve as the primary representatives of the minority class

The following Pseudo-code details the BorderlineSMOTE [51].

```
Algorithm BorderlineSMOTE
Begin
Input:
  • P number of minority class sample;
  • S% amount of synthetic to be generated;
  • M number of nearest neighbors to create the borderline subset;
  • k Number of nearest neighbors
Output:
  • $(S/100)^{*}P'$ synthetic samples
  1. Create function MinDanger ()
     {For $i \leftarrow 1$ to P
     Compute M nearest neighbors of each minority instance and other instances from the
     dataset.
      Check number of Majority instance M' within the Mnn
      IF M/2<M'<M Add instance P to borderline subset P' End IF
     End For}
```

```
    2. ComputKNN ($i \leftarrow 1$ $to$ $P', P'_i, P_j$)
    3. $N_s = (S/100)^* P'$
       While $N_s \neq 0$
    4. GenerateS $P'_i, P_j$
       $N_s = N_s - 1$
       End while
5. Return
6. End.
```

Table 1.3: illustrates a comparison between these variants, highlighting their enhanced findings, research gaps, and limitations [38].

Method	Findings	Research Gaps	Limitations
SMOTE	Enhanced minority class classification accuracy	Limited research on high-dimensional datasets	Lack investigation of parameter sensitivity
SVMSMOTE	Enhanced classification performance with support vector machines and SMOTE oversampling	Limited research on the effects of scalability and parameter selection	High processing costs for big datasets
BorderlineSMOTE	improved results with unbalanced datasets	Absence of research on using combined with other oversampling methods	high processing costs for big datasets

Table 1.3 Comparative Analysis of SMOTE, SVMSMOTE and BorderlineSMOTE Techniques

1.10 Gradient Boosting Machine

The term “gradient” describes the residual inaccuracy that is found after creating a model. “Boosting” means getting better. GBM, or gradient boosting machine, is the term for the method. A technique for progressively improving (reducing) inaccuracy is gradient boosting [39].

Gradient Boosting Machine (GBM) is a highly effective supervised learning algorithm that combines multiple weak learners into a robust ensemble, achieving excellent predictive performance. It excels in various prediction tasks, including spam filtering, online advertising, fraud detection, anomaly detection, and computational physics, such as the discovery of the Higgs Boson. GBM is frequently among the top algorithms in Kaggle competitions and the KDD Cup [40].

GBM is adept at handling heterogeneous datasets, which may include highly correlated data, missing data, and categorical data. It constructs an additive model, leading to interpretable results. Moreover, GBM is user-friendly, with numerous publicly available implementations, including scikit-learn, Spark MLLib, LightGBM, XGBoost, and TensorFlow Boosted Trees.

In the context of a supervised learning problem with n training examples, where x_i represents the feature vector of the i-Th example in$\mathbb{R}^p$, and y_i is the corresponding label (in classification) or continuous response (in regression), the classical version of Gradient Boosting Machine (GBM) predicts $f(x)$ for a feature vector x using an additive model of the form:

$$f(x) \coloneqq \sum_{m=1}^{M} \beta_{jm} b(x;\tau_{jm}) \text{ (1)}$$

This configuration is a simple function of the feature vector indexed by a parameter τ, where each basic function $b(x;\tau)$ belonging to $\mathbb{R}$ (referred to as a weak learner) is. The parameters τ_{jm} and coefficients β_{jm} are chosen flexibly to improve data integrity according to a particular standard, as shown below. In practical applications, weak learners are frequently employed in the following ways: wavelet functions, support vector machines, one-level decision trees (sometimes called tree stumps), and classification and regression trees (CART).

With a size of K, we suppose a limited set of weak learners. However, in many real-world scenarios, like the ones described above, K can grow exponentially enormous, resulting in computing complexity.

Let $\ell(y, f(x))$ represent the data fidelity measure for the loss function, which is considered differentiable in the second coordinate, at the observation(y,x) . A key goal in machine learning is to find a function f that minimizes the anticipated loss $\mathbb{E}_p(\ell(y,f(x)))$, where the expectation is taken out the unknown distribution of (y,x), represented by P. Using an algorithm like the Gradient Boosting Machine (GBM), one method to roughly reduce the empirical loss is one way to accomplish this goal.

GBM is a method that effectively minimizes the empirical loss in order to get a good estimate of f:

$$\underset{f}{min} \sum_{i=1}^{n} \ell(y, f(x_i)) \text{ (2)}$$

GBM (Gradient Boosting Machines) minimizes a loss function by iteratively updating predictions. It starts from a null model (f_0) and computes pseudo-residuals r^mat each iteration.

The pseudo-residuals are the negative gradients of the loss function with respect to the prediction: $r^m = -\partial\ell(y_i, f^m(x_i))/\partial f^m(x_i)$ for $i = 1, \ldots, n$

BM then finds the best weak-learner that fits these residuals in a least squares sense:

$$j_m = \underset{j \in [K]}{\arg\min} \ \underset{\sigma}{\min} \sum_{i=1}^{n} (r_i^m - \sigma b(x_i;\tau_j))^2 \text{ (3)}$$

The notation “[K]” represents a shorthand for the set {1… K}. in cases where there are ties in the “argmin” operation (as mentioned in equation (3)), we choose the element with the smallest index [40]. The GBM Algorithm is described bellows.

```
**Algorithm** Gradient Boosting Machine (GBM)
**Initialization.** Initialize with $f^0(x) = 0$
For $m = 0, \dots, M-1$ do:
    1. Compute pseudo-residual $r^m = -[\frac{\partial \ell(y_i, f^m(x_i))}{\partial f^m(x_i)}]_{i=1,\dots,n}$
    2. Find the best weak-learner $j_m = \underset{j\in[K]}{\arg\min} \; \underset{\sigma}{\min} \; \sum_{i=1}^{n}(r_i^m - \sigma b(x_i;\tau_j))^2$.
    3. Choose the step-size pm by line-search:
       $\rho_m = \arg min_{j\in[K]} \min_{\sigma} \sum_{i=1}^{n}(r_i^m - \sigma b(x_i;\tau_j))^2$.
    4. Update the model $f^{m+1}(x) = f^m(x) + \rho_m b(x;\tau_{jm})$
**Output.** $\boldsymbol{f^M(x)}$
```

1.11. K-nearest neighbors (KNN)

The K-nearest neighbors (KNN) classification approach is well-known for its effectiveness and simplicity. It is frequently chosen due to its ease of interpretation and speed of the computation. The selection of the parameter "k" is a crucial factor that impacts the efficiency of this algorithm. [45]

The K-nearest neighbors (KNN) algorithm extends the concept of nearest neighbor rules by considering the class labels of multiple neighboring samples. Among the k closest samples, it chooses the class label that is closest to the one under test during the decision-making stage. The KNN increases this to k samples, allowing for the exploitation of additional information, in contrast to the nearest neighbor rule, which only takes the closest sample into consideration. This modification adds more context, which improves the algorithm's speed. KNN is simpler and more straightforward in its approach than other classification algorithms that have separate training phases since it does not require a separate learning approach[46].

If y represents the nearest neighbor instance of x within set E, then the category of $y$becomes the decision outcome, following the nearest neighbor rule. Given an unknown category sample X, the decision process can be specified as follows:

$$g_j(X) = \min g_i \quad i = 1,2,\dots,C$$

Then the decision result is X ∈ W

The nearest neighbor rule is introduced from two perspectives: convergence and generalization error. When testing the same point x, the nearest neighbor x' obtained from two different training sets with distinct samples can vary. Since the classification outcome hinges on the category label of the nearest neighbor, it leads to the conditional error rate, denoted as

$P(e|x,x')$, dependent on both x and x'. Here, the average of x' can be calculated as follows:

$$P(e|x,) = \int P(e|x,x')P(x'|x)\,dx'$$

Among these equations, $P(x'|x)$ represents a conditional probability density function. Assuming $P(.)$is a continuous non-zero function, the probability that any point falls within the x-centered hypersphere S is calculated as follows:

$$P_s = \int P(x')\, dx'$$

The probability that all n samples fall outside the hypersphere is $(1 - P_s)^*$. if $n \to \infty$, this probability tends to zero. Consequently, if the nearest neighbor x0 converges to the point x being measured according to this probability, then $P(e|x)$ infinitely approaches the Dirac function. Similarly, if the KNN decision rule is followed, k neighbors converge to the point x being measured.

The error rate of the nearest neighbor can be interpreted as the probability that the point x being measured differs from the category c of the nearest neighbor point x', and the error rate is calculated as follows:

$$P(error) = 1 - \sum_{c \in Y} P(c|x)P(c|x')$$

Here, the assumption is that each sample is independently and equally distributed. A sample x can always be found within the d distance range around x, thus the Bayesian classifier can be expressed as:

$$c^* = \arg max_{c \in Y} P^2(c^*|x)$$

At this point, the inequality holds:

$$P(error) \leq 1 - \sum\nolimits_{c \in Y} P^2(C^*|x)$$

This leads to the conclusion that the nearest neighbor rule, while simple in its construction, also ensures that the generalized error rate is no more than twice the Bayesian error rate. The KNN Algorithm is described bellows.

```
Algorithm  The k-nearest neighbors classification algorithm
Input:
D: a set of training samples $\{(x_1, y_1), \dots., (x_n, y_n)\}$
k: the number of nearest neighbors
d$(x, y)$a distance metric
$x$ : a test sample
for each training sample $(X_i, y_i) \in D$ do
Compute d$(x, x_i)$ , the distance between $x$and $x_i$
Let N⊆D be the set of training samples with the k smallest distances d$(x, x_i)$
return the majority label of the samples in N
```

1.12. Related works

The authors in [12] evaluated implicit bias in healthcare using metrics such as Accuracy, Equal opportunity, and Predictive equity. They introduced HOUSES as a new feature for individual-level socioeconomic status (SES) measurement. Their study revealed that asthmatic children with lower SES had larger balanced error rates than those with higher SES. Also, focusing on implicit algorithmic bias, [13] utilized statistical parity and calibration-in-the-large. They found that predictive models struggle to classify minority class instances correctly and fairly.

Another study on implicit algorithmic bias, using Statistical parity, showed that performance decreases for mortality prediction in minority racial and socioeconomic groups when comparing the entire cohort to subpopulations [14].

The study [15] also examined implicit algorithmic bias through Statistical parity. They discovered performance differences among different groups, highlighting the reliance of models on different racial attributes.

In their work Raza et Bashir [16] investigated implicit algorithmic bias using Statistical parity. They explored performance differences across various groups and highlighted the impact of different racial attributes on model predictions. Their findings contribute to our understanding of bias mitigation in machine learning models.

The study by Allen et al [17]. focuses on mitigating racial bias in early warning and mortality scoring systems. Specifically, they address disparities between White and non-White racial groups. To achieve this, the authors employ a preprocessing reweighting approach based on probability. They assign individual weights to training examples based on mortality status and race within each age stratum. By doing so, they effectively reduce racial bias and enhance the accuracy of existing mortality score predictors.

Similarly, another study [18] tackles class bias caused by data sparsity. They apply preprocessing resampling techniques to improve prediction results.

The study conducted by Lee et al [19] focuses on mitigating healthcare disparities caused by imbalanced ophthalmic clinical data among racial groups. They propose an in-processing transfer learning approach using Mean Absolute Error (MAE) and Mean Squared Error (MSE). The two-step method involves transferring information from a source domain by fitting a linear model on a combined dataset, followed by bias correction through fitting the contrast solely on the target domain. This approach outperforms conventional methods in terms of both accuracy and equity.

In [20] authors address bias in risk assessment predictive models due to systematic bias in training data. Their preprocessing methods include resampling and blinding to achieve equal opportunity and statistical parity. They propose three strategies: removing protected attributes, resampling the imbalanced training dataset by sample size, and resampling by case proportion of people with cardiovascular disease (CVD) outcomes. While removing protected attributes and simple resampling did not significantly reduce bias, resampling by case proportion effectively reduced bias for gender groups without compromising overall accuracy.

In Zhu et al’s study [21], the researchers focused on addressing bias in hospital readmission prediction models due to skewed population distributions. They utilized a preprocessing technique called localized sampling, which involved resampling instances based on locality assessed through LDA embedding. The evaluation metric employed was AUROC. The research aimed to overcome the challenge of imbalanced samples, particularly in cases where re-admission patients constituted a small proportion of the population. Their localized sampling approach successfully tackled the sample imbalance issue, resulting in more effective hospital readmission predictions.

In Ref. [22], authors employed a bias handling approach in their study, focusing on preprocessing techniques such as resampling, specifically using SMOTE (Synthetic Minority Over-sampling Technique) combined with Neural Network. Their evaluation metric of choice was accuracy. Addressing the research question of whether imbalanced datasets lead to bias in the Istitaah classification system, they found that SMOTE effectively oversampled minority classes by generating new instances, thereby creating a more balanced dataset. The combination of SMOTE and Neural Network yielded the most accurate classification results, indicating its effectiveness in mitigating bias caused by imbalanced data.

In Hee's study [23], the focus was on addressing bias in clinical data for mortality prediction through preprocessing techniques. They employed resampling, specifically stratified random sampling, to handle bias. Their evaluation utilized AUROC and accuracy metrics. The research aimed to mitigate bias in reused clinical data, emphasizing the importance of data quality assurance methods. Their approach involved Clinical Data Quality Assessment (CDQA) and Mortality Data Quality Assessment (MDQA) to identify relevant variables for stratified sampling. The results showed that CDQA and MDQA effectively stratified sampled inputs, leading to improvements in predictive performance, as evidenced by increased AUC and accuracy.

In this study, [24]. Address bias in machine learning models caused by missingness not at random. They employ a preprocessing approach involving transformation through multiple imputation to handle bias. Evaluation is conducted using percent bias as the metric. The study compares different bias handling methods, specifically multiple imputation using chain equation, random forest, and denoising autoencoder for imputing missing values. The results indicate that the denoising autoencoder method does not demonstrate superior performance compared to traditional multiple imputation techniques.

The study of Yin et al [25] present a novel approach to addressing bias in predictive modeling, focusing on preprocessing techniques and evaluation metrics such as AUPRC and MAE. They identify the challenge of traditional models being hindered by observational bias and partial observations, leading to degraded performance. Their proposed method involves utilizing three subnetworks to impute missing data through propensity score adjustment. Results indicate that this model surpasses existing methods in tasks like binary data imputation, disease progression modeling, and mortality prediction, showcasing its efficacy in handling bias and improving predictive accuracy.

In [26] Davoudi et al. employed an in-processing approach known as reweighting to address bias in predictive models, focusing on mitigating systematic outcome predictions favoring certain socioeconomic groups. Their evaluation centered on metrics including Equal Opportunity, Predictive Equality, and Statistical Parity. The study aimed to rectify predictive disparities among groups. Their method involved adjusting observation weights in attribute-outcome combinations during model training. While reweighing successfully reduced bias in certain instances, it inadvertently introduced bias in scenarios where none initially existed.

In the study introduced by [27], authors tackle bias in ranking models through a postprocessing approach known as transformation (xOrder). Focused on addressing the issue of ranking positive instances higher than negative ones with poor fairness, they introduce the xAUC evaluation metric. Their research aims to mitigate systematic disparities across various protected groups caused by biased rankings. Their proposed framework employs dynamic programming to adjust ranking scores, optimizing the ordering by minimizing an objective that combines algorithm utility loss and ranking disparity. The results demonstrate that the framework consistently achieves a better balance between algorithm utility and ranking fairness across diverse datasets and metrics.

In the study [28], the focus was on addressing temporal bias in longitudinal Electronic Health Record (EHR) data, particularly concerning patients with varying disease progression states. Their approach involves preprocessing through time alignment transformation to mitigate bias. Evaluation metrics such as AUROC, AUPRC, and F1-score were utilized to assess the effectiveness of their method. By aligning patients' timelines to a shared reference point instead of solely relying on hospital or ICU admission time as the starting point, they found that this registration technique notably improved mortality prediction, yielding an enhancement of at least 1-2% across evaluation metrics.

Wolk et al [29], propose a bias handling approach aimed at improving the identification of high-risk septic shock patients, addressing the limitations of general scoring systems due to covariate shift and systematic bias. Their preprocessing strategy involves relabeling through domain adaptation. Evaluation metrics such as AUROC, Accuracy, Sensitivity, Specificity, and F1-score were utilized. The study introduces a VRNN-based Adversarial Domain Separation model, which effectively separates globalshared representations from local information across domains. The results demonstrate superior performance compared to existing domain adaptation methods, effectively mitigating both covariate shift and systematic bias in identifying high-risk septic shock patients.

Abay et al's study presents a novel framework addressing both federated learning (FL) and fairness concerns in machine learning models. Their study primarily utilizes the Adult dataset and COMPAS dataset for analysis. Despite the comprehensive approach, the paper's limitation lies in its focus solely on binary classification tasks, neglecting biased data within FL training data. The authors employ various methods to tackle bias, including local reweighing, global reweighing with privacy considerations, and federated bias removal. They also incorporate pre-processing and in-processing techniques to mitigate bias effectively. Accuracy assessments encompass fairness metrics, with a particular emphasis on an 11.5% threshold, alongside traditional metrics such as false positives, false negatives, and receiver operating characteristic curves.

1.13. Conclusion

Bias remains a significant challenge in biomedical applications, impacting the fairness and accuracy of healthcare outcomes. Addressing biases in both data and algorithms is crucial for developing unbiased AI systems. Techniques such as data preprocessing, algorithmic adjustments and post-processing methods offer potential solutions to mitigate biases. However, ongoing research and ethical considerations are essential to ensure these advancements promote equitable healthcare access, respect privacy, and maintain trust in clinical settings.

Chapter 2: Methodology

2.1 Introduction

Machine learning techniques are instrumental in enhancing predictive accuracy across diverse fields, particularly in healthcare and other domains requiring early detection and intervention. However, the challenge of class imbalance, where certain data classes are underrepresented, often compromises model effectiveness by introducing bias towards majority classes. To address this, oversampling methods like SMOTE variants generate synthetic data for minority classes, thereby balancing dataset distributions and improving overall prediction accuracy. Evaluation metrics such as accuracy, precision, recall, and F1 score gauge model performance, while fairness metrics like EOD, DI, and IR assess equitable treatment across different data subsets. Integrating these techniques ensures more robust and fair machine learning models, suitable for varied real-world applications through continual validation and refinement

2.2 Data Collection

Two distinct datasets were utilized for this study: The Heart Disease Prediction dataset [W2], containing 270 instances with 14 attributes, and the Early Stage Diabetes Risk Prediction Dataset [W3], comprising 520 instances with 17 attributes.

Heart Disease Prediction Dataset

- Age: Age (years)
- Sex: Gender (Female/Male), (the sensitive attribute)
- Chest-pain: chest pain type
- Rest-bp: resting blood pressure
- Cholesterol: serum cholestoral (mg/dl)
- FBS over 120: fasting blood sugar > 120 mg/dl
- EKG results: resting electrocardiographic results
- Max HR: maximum heart rate achieved
- Exercise angina: exercise induced angina
- ST depression: ST depression induced by exercise relative to rest
- Slope of ST: the slope of the peak exercise ST segment
- Number of vessels fluro: number of major vessels (0-3) colored by fluoroscopy
- Thallium: thal: 3 = normal; 6 = fixed defect; 7 = reversable defect

- Heart Disease: Class variable (0 or 1).

The bias is detected in the attribute ‘gender’ (the sensitive attribute), with the privileged group being female.

Early Stage Diabetes Risk Prediction Dataset

This dataset comprises crucial data on the signs and symptoms of individuals either exhibiting early signs of diabetes or at risk of developing the disease. It encompasses a wide range of variables, offering valuable insights into potential early indicators of diabetes onset. The dataset includes diverse information, from demographic details to specific diabetes-related symptoms. The data was collected via direct patient questionnaires at the Sylhet Diabetes Hospital in Sylhet, Bangladesh, and has been validated by a medical doctor.

- Age (1-20 to 65): Age range of the individuals.
- Sex (1. Male, 2. Female): Gender information. (Male 63% Female 37%)
- Polyuria (1. Yes, 2. No): Presence of excessive urination.
- Polydipsia (1. Yes, 2. No): Excessive thirst.
- Sudden Weight Loss (1. Yes, 2. No): Abrupt weight loss.
- Weakness (1. Yes, 2. No): Generalized weakness.
- Polyphagia (1. Yes, 2. No): Excessive hunger.
- Genital Thrush (1. Yes, 2. No): Presence of genital thrush.
- Visual Blurring (1. Yes, 2. No): Blurring of vision.
- Itching (1. Yes, 2. No): Presence of itching.
- Irritability (1. Yes, 2. No): Display of irritability.
- Delayed Healing (1. Yes, 2. No): Delayed wound healing.
- Partial Paresis (1. Yes, 2. No): Partial loss of voluntary movement.
- Muscle Stiffness (1. Yes, 2. No): Presence of muscle stiffness.
- Alopecia (1. Yes, 2. No): Hair loss.
- Obesity (1. Yes, 2. No): Presence of obesity.
- Class (1. Positive, 2. Negative) : Diabetes classification.

The bias is detected in the attribute ‘gender’ (the sensitive attribute), with the privileged group being Male.

To enhance the fairness metrics and overall performance of our biomedical machine learning models, we utilized two distinct datasets. By employing multiple datasets for comparison, we aimed to assess the resilience of our models and evaluate their performance across different data sources. This comparative approach allowed us to identify any discrepancies or biases inherent in individual datasets and to ensure more reliable and generalizable results. The use of multiple datasets provided a broader perspective, enabling us to make more informed decisions about model performance and potential improvements.

CSV files: A CSV (Comma Separated Values) file is a commonly used format for importing and exporting data in spreadsheets and databases. Although it lacks a standardized definition, it typically consists of data separated by commas, with each row representing a record and each comma separating individual fields within that record. While different applications may use varying delimiters and quoting characters, the overall structure remains similar, enabling the creation of modules to efficiently manipulate such data. The csv module in programming languages like Python implements classes to handle reading and writing CSV files, allowing programmers to interact with CSV data without needing to understand the specific details of its format [W4].

2.3 Attribute Distribution

Figure 2.1 illustrates the Heart Disease Prediction dataset, presenting a pie chart that depicts both the distribution of outcomes and gender demographics. The chart highlights an overall imbalance with 55.6% positive cases ('1') and 44.4% negative cases ('0'). Additionally, it shows a gender disparity where the male segment ('1') is notably larger than the female segment ('0'). This dual insight suggests potential biases in machine learning models towards predicting negative outcomes and favoring male subjects, necessitating strategies for achieving fairer predictions across genders.

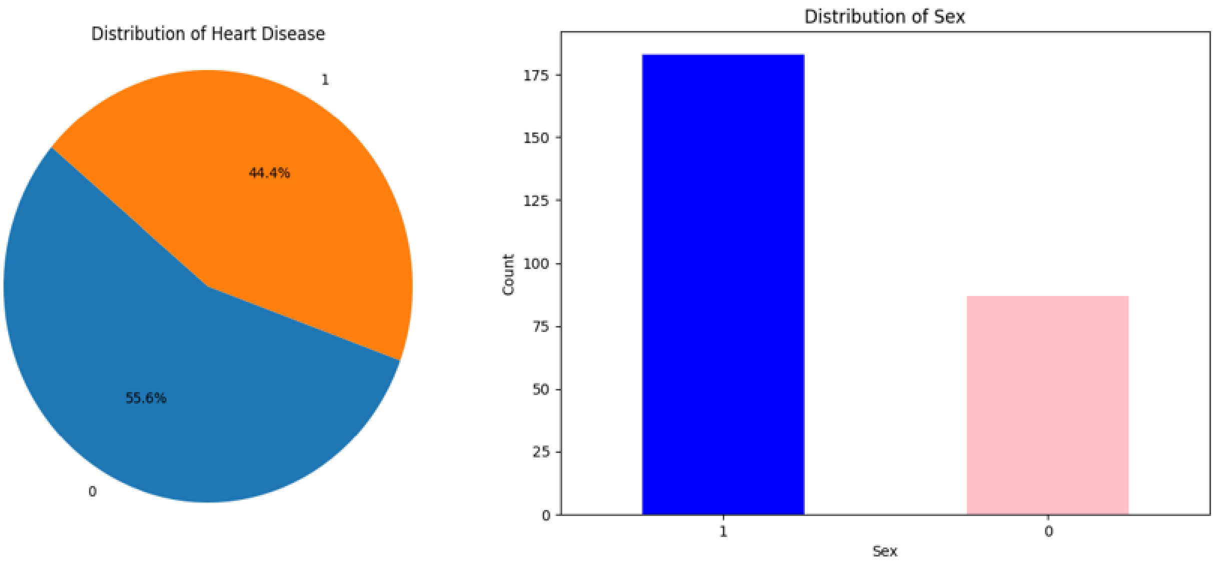


Figure 2.1: Outcome and Gender distribution in Heart Disease Prediction Dataset

In the Diabetes Risk Prediction Dataset (Figure 2.2), the pie chart illustrates both the distribution of outcomes and gender demographics. The chart reveals a notable imbalance with 61.5% positive cases ('1') and 38.5% negative cases ('0'). Furthermore, it highlights a gender disparity where males ('0') significantly outnumber females ('1'). This dual observation suggests a potential bias in machine learning models towards predicting positive outcomes and favoring male subjects, necessitating strategies to ensure equitable predictions across genders.

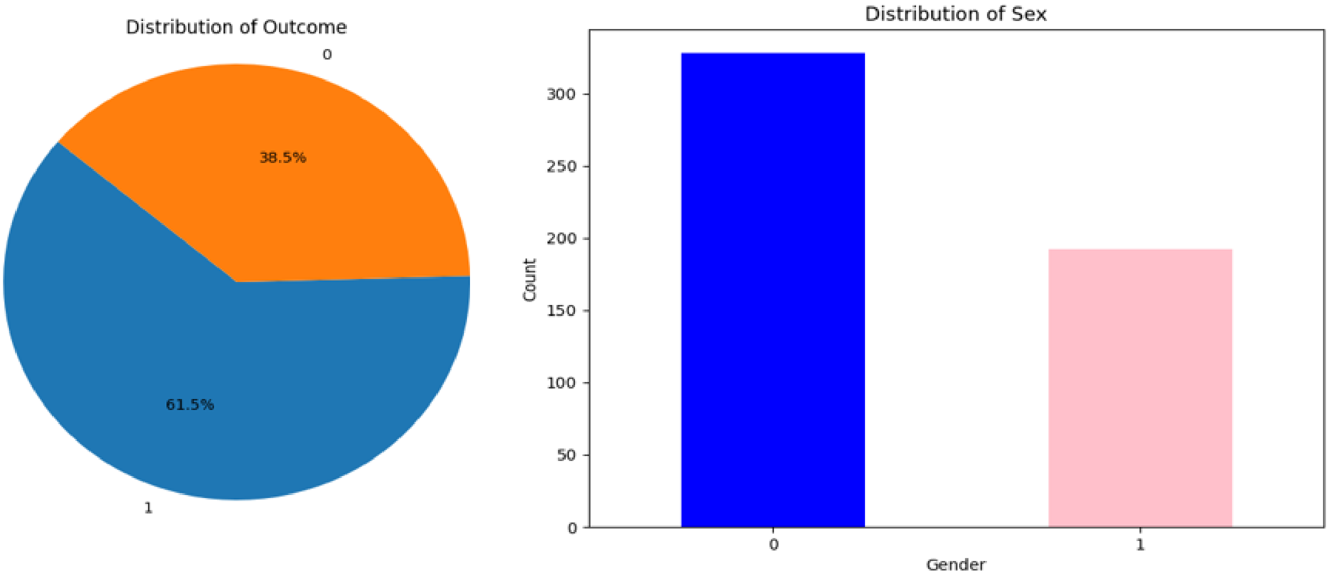


Figure 2.2: Outcome and Gender distribution in Diabetes Risk Prediction Dataset

Across both datasets, addressing these imbalances is critical to prevent bias and to ensure that machine-learning models make equitable predictions for all classes.

2.4 Proposed approach

In this section, we propose an architecture for a biomedical dataset bias reduction system using a machine learning approach, as illustrated in (Figure 2.3). This system aims to effectively detect and mitigate dataset bias, thereby enhancing the equity and reliability of machine learning models in biomedical applications.

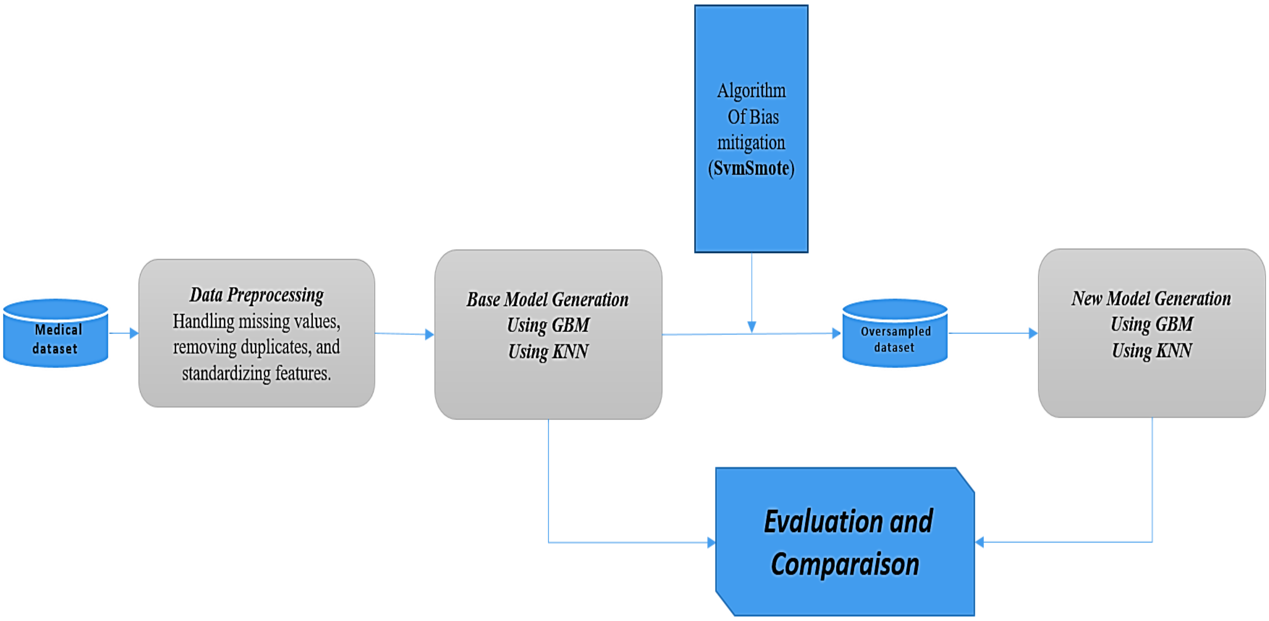


Figure 2.3: Proposed approach for dataset bias mitigation.

Steps:

Data Preprocessing:

- Clean and preprocess the biomedical dataset. This step involves handling missing values, removing duplicates, and standardizing features.

Base Model Generation:

- Train a Gradient Boosting Machine (GBM) model or KNN model on dataset.
- Evaluate its performance using metrics such as accuracy, precision, recall, F1 score, positive predictive value (PPV), equal opportunity difference (EOD), Impact Ratio (IR), and disparate impact (DI).

Bias Detection:

- Investigate potential sources of bias in the dataset

Bias Mitigation Techniques:

- We have chosen oversampling with SVM-SMOTE and BorderlineSMOTE to address class imbalance. SVM-SMOTE and BorderlineSMOTE generates synthetic samples for the minority class while preserving the decision boundary, which helps in reducing bias.

New Model Generation:

- Post-Oversampling, Train another GBM model or KNN model on the oversampled dataset.
- Calculate the same evaluation metrics (accuracy, precision, recall, F1 score, PPV, EOD, DI, IR) for this model.

Comparison:

- Compare the metrics before and after oversampling:

Did oversampling improve fairness (reduced bias)?

Did it impact overall model performance (accuracy, etc.)?

2.4.1 Preprocessing

- ***Checking messing values***

Dealing with missing data is a common challenge in medical research, and while there are several strategies available to handle it, it is reassuring to note that neither of our datasets currently contains any missing values. This absence of missing data simplifies the analysis process and ensures that our findings are based on complete information. However, it is always prudent to remain vigilant for potential missing data issues as datasets evolve or new data is collected, as addressing them promptly is essential for maintaining the integrity and reliability of our research findings.

- ***Data Normalization***

Normalization involves adjusting numerical data so that it fits within a predetermined range, typically 0 to 1 or -1 to 1. This method is particularly advantageous for data mining tasks such as classification, clustering, and artificial neural networks.

We applied the widely adopted normalization method called min-max normalization, which can be represented mathematically as follows:

$$x_{norm} = (x - min)/(max - min) \tag{2.1}$$

2.4.2 Model Generation :

- ***Gradient Boosting Machines (GBM)***

In this stage, we used GBM to generate an Ensemble Learning Model. The training parameters are described in table 2.1.

Parameter	Typical Values
n_estimators	100-500 (default: 100)
learning_rate	0.01-0.1 (default: 0.1)
max_depth	3-10 (default: 3)
min_samples_split	2-10 (default: 2)
min_samples_leaf	1-10 (default: 1)
subsample	0.5-1.0 (default: 1.0)
random_state	Integer (e.g., 42) default (None)
alpha	0.9 (default for 'quantile')
validation_fraction	0.1-0.2 (default: 0.1)
n_iter_no_change	Integer (e.g., 10) default (None)
init	An estimator object implementing 'fit' method default (None)

Table 2.1: Training and Testing Parameters for GBM Ensemble Learning Model

- ***K-Nearest Neighbor (KNN)***

We used also KNN to generate a Machine Learning Model. The training and test parameters are described in table 2.2.

Parameter	Description
random_state	None, integer (e.g., 42), or np.random.RandomState instance default (None)
n_neighbors	Integer (default: 5), we used “7”
weights	‘uniform’,‘distance’, or custom function (default: uniform)
algorithm	‘auto’, ‘ball_tree’, ‘kd_tree’, ‘brute’(default: auto)
leaf_size	Integer (default: 30)
P	Integer (default: 2)
Metric	String or callable (e.g., ‘minkowski’)(default: minkowski)
metric_params	Dictionary or None (default: None)
n_jobs	Integer (default: 1)

Table 2.2: Training and Testing Parameters for KNN Machine Learning Model

2.4.4 Bias Mitigation Techniques:

To address the imbalances and potential biases in our health outcome prediction datasets, we applied advanced oversampling techniques, specifically SVM-SMOTE and BorderlineSMOTE. These methods were used to correct both class and gender disparities, aiming to produce fairer and more accurate predictions.

Initially, the Heart Disease Prediction Dataset had a notable imbalance in both gender and target class distributions. Out of the total instances, there were 183 females and 87 males. Additionally, the target classes were imbalanced, with 120 instances indicating the presence of heart disease and 150 indicating its absence. Such skewed distributions posed a risk of the model developing biases, either favoring the majority class or displaying gender bias against the less represented groups.

After applying SVM-SMOTE and BorderlineSMOTE, we achieved a balanced class distribution with 150 instances each for both the presence and absence of heart disease. The gender distribution post-oversampling improved to 183 females and 117 males. While still slightly imbalanced in terms of gender, this represents a more equitable distribution compared to the initial state, helping to mitigate potential biases.

The Diabetes Risk Prediction Dataset also exhibited imbalances, with 192 females and 328 males, and 320 instances indicating a positive risk for diabetes against 200 indicating no risk. Such disparities could lead to biased predictions favoring the majority class or gender.

Post oversampling, we balanced the class distribution to 320 instances each for positive and negative diabetes outcomes. The gender distribution was adjusted to 312 females and 328 males, significantly improving the representation equity compared to the original dataset.

SVM-SMOTE and BorderlineSMOTE enhance dataset balance by generating synthetic samples for the minority class, carefully considering the decision boundary. This not only corrects class imbalances but also ensures a more equitable gender representation. By doing so, these techniques reduce potential biases and promote fairer, more inclusive predictions across different demographic groups.

The following Figure 2.4 and Figure 2.5 illustrate the outcome and gender distribution in the Heart Disease and Diabetes Risk Prediction datasets after applying SVM-SMOTE and BorderlineSMOTE, highlighting the improved balance achieved through these techniques:

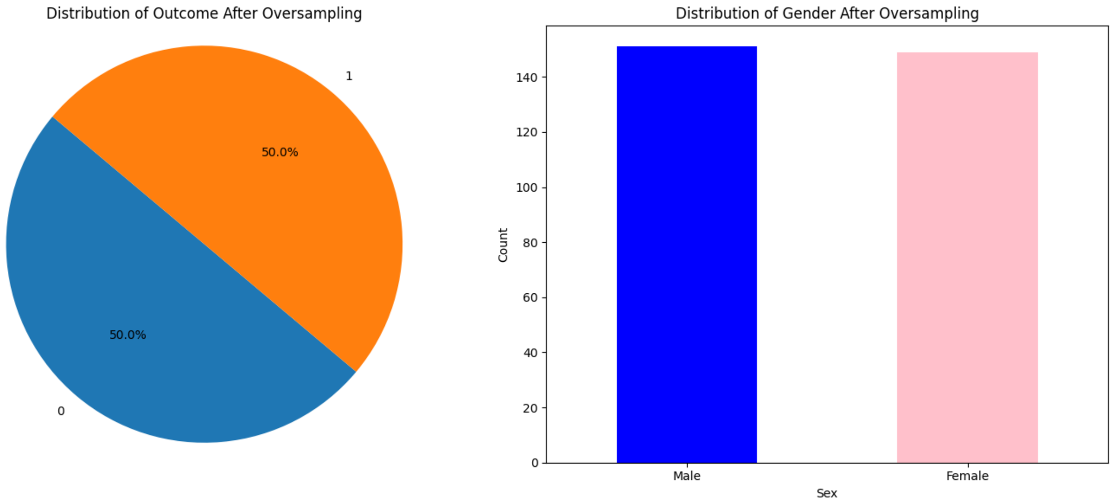


Figure 2.4: Outcome and Gender distribution after SVM-SMOTE and BorderlineSMOTE Heart Disease Dataset

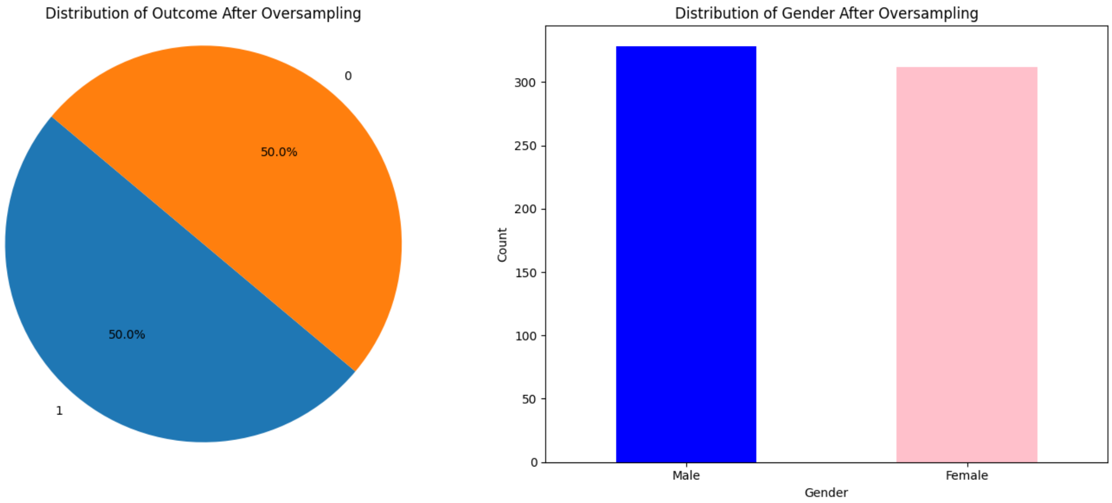


Figure 2.5: Outcome and Gender distribution after SVM-SMOTE and BorderlineSMOTE (Diabetes Risk Prediction Dataset)

2.4.5 New Model Generation

Following the application of SVM-SMOTE and BorderlineSMOTE to oversample the Heart Disease Prediction and Diabetes Risk Prediction datasets, Gradient Boosting Machine (GBM) models were trained on each oversampled dataset variant. Specifically, separate GBM models were trained on the datasets oversampled using SVM-SMOTE and BorderlineSMOTE for both health prediction scenarios. Similarly, K-Nearest Neighbors (KNN) models were trained on these datasets post-oversampling, with one set trained on SVM-SMOTE oversampled data and another on BorderlineSMOTE oversampled data for each health prediction dataset.

After applying SVM-SMOTE and BorderlineSMOTE to oversample the datasets for heart disease and diabetes risk prediction, various models including Gradient Boosting Machines (GBM) and K-Nearest Neighbors (KNN) were trained and evaluated. Comprehensive metrics such as Accuracy, Precision, Recall, F1 score, positive predictive value (PPV), Equal Opportunity Difference (EOD), Disparate Impact (DI), and Imbalance Ratio (IR) were used to assess each model's performance and fairness across demographic groups. This thorough evaluation highlighted the impact of oversampling techniques on both model effectiveness and equity, ensuring that predictions were both accurate and unbiased in healthcare decision-making. This approach emphasizes the importance of advanced oversampling methods in creating balanced datasets, thereby improving the overall predictive performance and fairness of models in critical health prediction tasks.

2.4.3 Evaluation, results and discussion

Before oversampling

In the realm of evaluating classification models, various metrics serve as tools to dissect different dimensions of a model’s predictive abilities.

This evaluation can be conducted using the **accuracy_score**, **precision_score**, **recall_score**, and **f1_score** functions, respectively.

Table 2.3 Table 2.4 Table 2.5 Table 2.6 displays the results obtained from both datasets before oversampling:

Accuracy	Precision	Recall	F1-score
74%	70%	75%	72%

Table 2.3: Obtained results before bias mitigation techniques of Heart Disease Dataset using GBM

Accuracy	Precision	Recall	F1-score

68%	**60%**	**62%**	**60%**

Table 2.4: Obtained results before bias mitigation techniques of Heart Disease Dataset using KNN

Accuracy	Precision	Recall	F1-score
99%	**100%**	**98%**	**99%**

Table 2.5: Obtained results before bias mitigation techniques of Diabetes Risk Dataset using GBM

Accuracy	Precision	Recall	F1-score
86%	**93%**	**84%**	**88%**

Table 2.6: Obtained results before bias mitigation techniques of Diabetes Risk Dataset using GBM

The obtained results can be expressed using the bar plot in Figure 2.6, Figure 2.7, Figure 2.8 and Figure 2.9

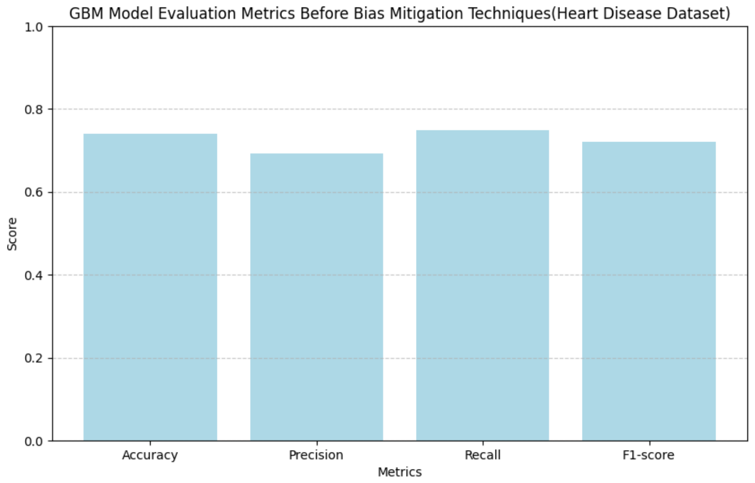


Figure 2.6: Obtained results before bias mitigation techniques of Heart Disease Dataset using GBM

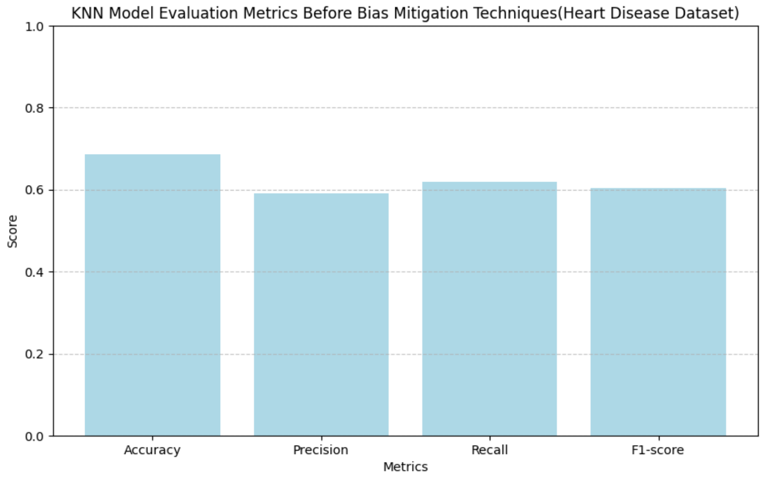


Figure 2.7: Obtained results before bias mitigation techniques of Heart Disease Dataset using KNN

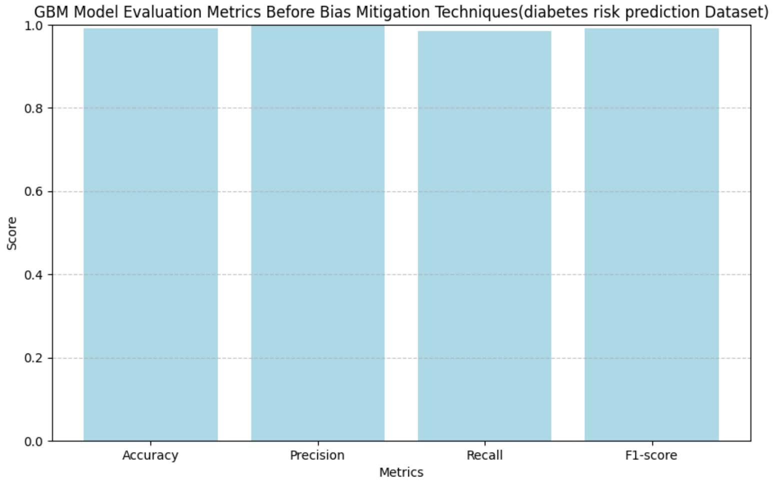


Figure 2.8: Obtained results before bias mitigation techniques of Diabetes Risk Dataset using GBM

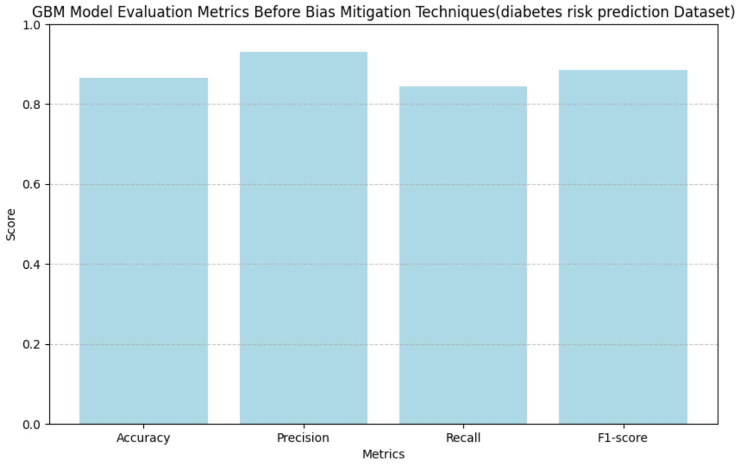


Figure 2.9: Obtained results before bias mitigation techniques of Diabetes Risk Dataset using KNN

- **Confusion Matrix**

After predicting the labels for x_test and obtaining the predicted class labels (y_pred), we computed the confusion matrix (as shown in Figure 2.10, Figure 2.11, Figure 2.12 and Figure 2.13). The confusion matrix is a crucial tool in evaluating the performance of a classification model. It provides a tabular summary that categorizes predictions into four key outcomes: true positives, true negatives, false positives, and false negatives. This detailed breakdown allows for a comprehensive analysis of the model's accuracy and performance.

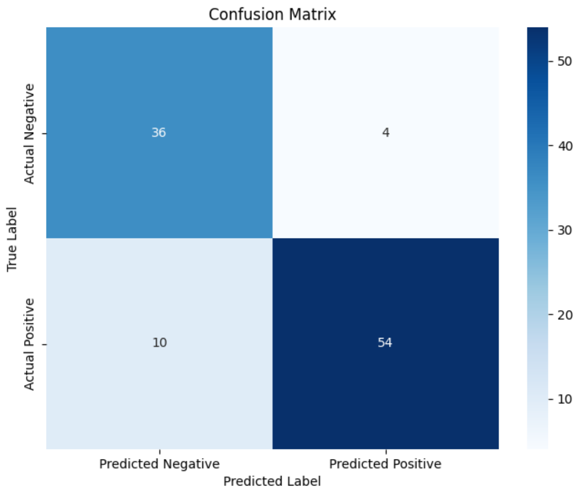


Figure 2.10: Confusion matrix of Diabetes Risk Prediction Dataset generated using KNN before Oversampling.

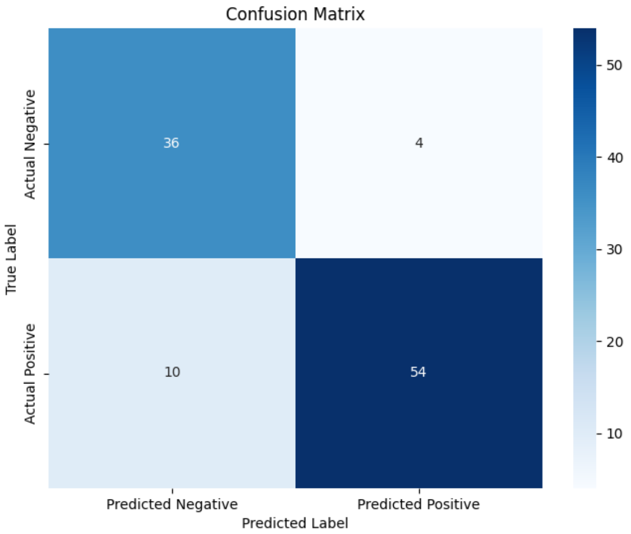


Figure 2.11: Confusion matrix of Diabetes Risk Prediction Dataset generated using GBM before Oversampling.

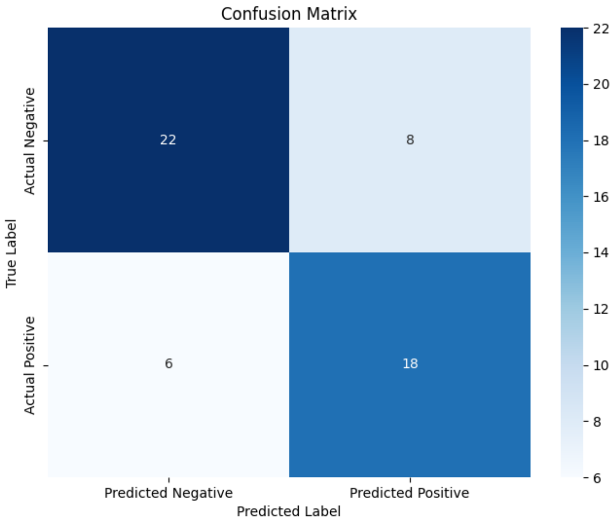


Figure 2.12: Confusion matrix of Heart Disease Prediction generated using GBM before oversampling.

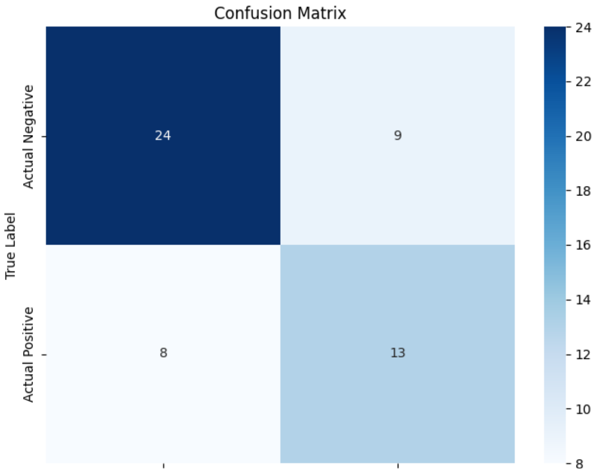
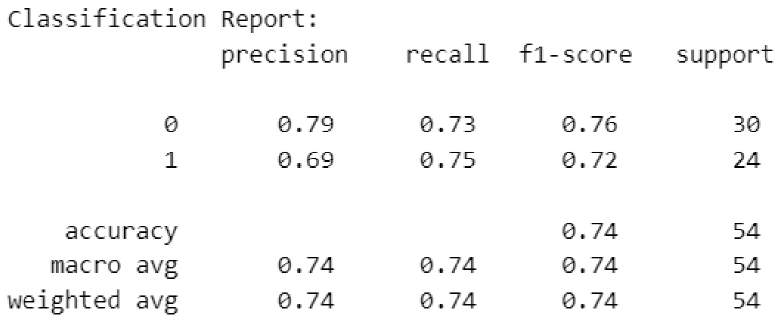


Figure 2.13: Confusion matrix of Heart Disease Prediction generated using KNN before oversampling.

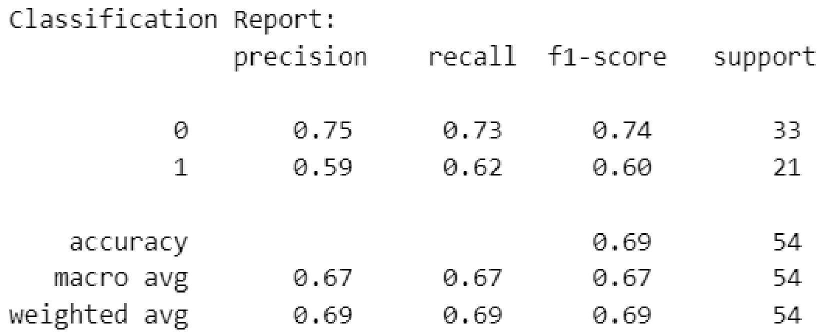
Classification Report

Figure 2.14, Figure 2.15, Figure 2.16 and Figure 2.17 bellow illustrates Classification Report before bias mitigation techniques for the both datasets



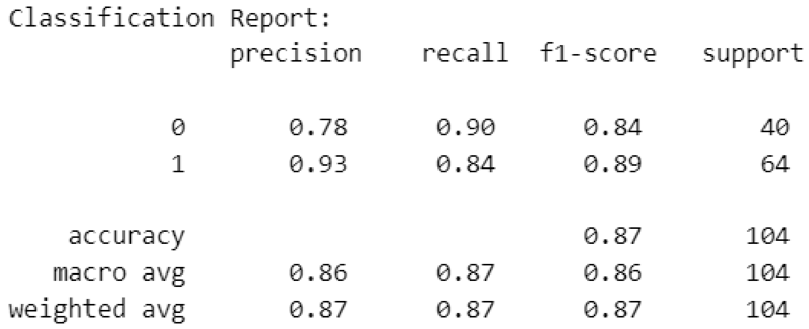
```
Classification Report:
              precision    recall  f1-score   support

           0       0.79      0.73      0.76        30
           1       0.69      0.75      0.72        24

    accuracy                           0.74        54
   macro avg       0.74      0.74      0.74        54
weighted avg       0.74      0.74      0.74        54
```

Figure 2.14 Classification Report before bias mitigation techniques Heart Disease Dataset using GBM



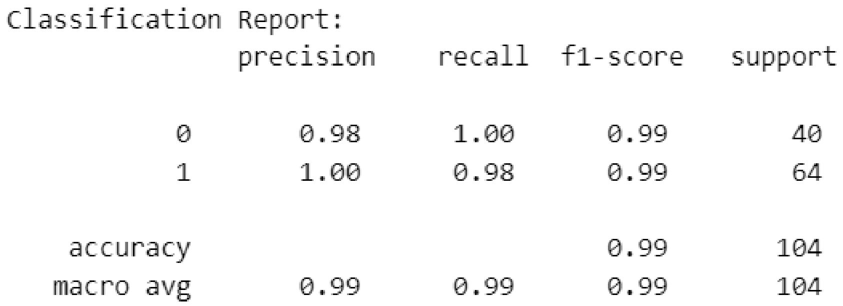
```
Classification Report:
              precision    recall  f1-score   support

           0       0.75      0.73      0.74        33
           1       0.59      0.62      0.60        21

    accuracy                           0.69        54
   macro avg       0.67      0.67      0.67        54
weighted avg       0.69      0.69      0.69        54
```

Figure 2.15: Classification Report before bias mitigation techniques Heart Disease Dataset using KNN



```
Classification Report:
              precision    recall  f1-score   support

           0       0.78      0.90      0.84        40
           1       0.93      0.84      0.89        64

    accuracy                           0.87       104
   macro avg       0.86      0.87      0.86       104
weighted avg       0.87      0.87      0.87       104
```

Figure 2.16: Classification Report before bias mitigation techniques of Diabetes Risk Dataset



```
Classification Report:
              precision    recall  f1-score   support

           0       0.98      1.00      0.99        40
           1       1.00      0.98      0.99        64

    accuracy                           0.99       104
   macro avg       0.99      0.99      0.99       104
```

Figure 2.17: Classification Report before bias mitigation techniques of Diabetes Risk Dataset using GBM

Evaluation of fairness metrics:

Table 2.8, Table 2.9, Table 2.10 and Table 2.11 bellow illustrates evaluation of fairness metrics before bias mitigation techniques for the both datasets using GBM and KNN.

Positive Predictive Value (PPV)	Disparate Impact	Equal Opportunity Difference (EOD)	Impact ratio
0.69	1.60	0.06	1.25

Table 2.7: Fairness metrics before bias mitigation techniques of Heart Disease Dataset using GBM

Positive Predictive Value (PPV)	Disparate Impact	Equal Opportunity Difference (EOD)	Impact ratio
0.59	0.69	0.16	1.25

Table 2.8 Fairness metrics before bias mitigation techniques of Heart Disease Dataset using KNN

Positive Predictive Value (PPV)	Disparate Impact	Equal Opportunity Difference (EOD)	Impact ratio
1.0	0.43	0.03	1.60

Table 2.10: Fairness metrics before bias mitigation techniques Diabetes Risk Dataset using GBM

Positive Predictive Value (PPV)	Disparate Impact	Equal Opportunity Difference (EOD)	Impact ratio
0.93	0.47	0.08	1.60

Table 2.11: Fairness metrics before bias mitigation techniques Diabetes Risk Dataset using KNN

After oversampling

Table 2.12 presents a comparison of the Gradient Boosting Machine (GBM) model's performance in predicting heart disease, both before and after applying SVM-SMOTE and BorderlineSMOTE. The table evaluates key metrics such as Accuracy, Precision, Recall, F1-Score, PPV, EOD, DI, and IR. This analysis provides insights into the effectiveness of GBM when enhanced by these oversampling techniques, illustrating improvements in handling imbalanced data and promoting fairness across different demographic groups.

Ensemble Learning: GBM "Heart Disease Prediction"			
Metric	**Before Oversampling**	**After Oversampling (SVM-SMOTE)**	**After Oversampling (BorderlineSMOTE)**
Accuracy	0.74	0.85	0.85
Precision	0.69	0.79	0.77
Recall	0.75	0.88	0.92
F1 Score	0.72	0.84	0.84
PPV (Positive Predictive Value)	0.69	0.80	0.77
EOD (Equal Opportunity Difference)	0.06	0.03	0.10
DI (Disparate Impact)	1.59	0.95	0.79
IR (Imbalance Ratio)	1.25	1.00	1.00

Table 2.12: Results before and after SVM-SMOTE and BorderlineSMOTE on Heart Disease Prediction Dataset Using GBM

Table 2.13 presents a comparison of the K-Nearest Neighbors (KNN) model's performance in predicting heart disease, both before and after applying SVM-SMOTE and BorderlineSMOTE.

Machine Learning: KNN "Heart Disease Prediction"			
Metric	**Before Oversampling**	**After Oversampling (SVM-SMOTE)**	**After Oversampling (BorderlineSMOTE)**
Accuracy	0.68	0.68	0.71
Precision	0.59	0.68	0.70

Recall	0.62	0.65	0.72
F1 Score	0.60	0.66	0.71
PPV (Positive Predictive Value)	0.59	0.68	0.7
EOD (Equal Opportunity Difference)	0.16	0.17	0.07
DI (Disparate Impact)	0.68	0.66	0.75
IR (Impact Ratio)	1.25	1.00	1.00

Table 2.13: Results before and after SVM-SMOTE and BorderlineSMOTE on Heart Disease Prediction Dataset Using KNN

Table 2.14 presents a comparison of the Gradient Boosting Machine (GBM) model's performance in Diabetes Risk Prediction Dataset after applying SVM-SMOTE and BorderlineSMOTE.

Ensemble Learning: GBM “Diabetes Risk Prediction”			
Metric	**Before Oversampling**	**After Oversampling (SVM-SMOTE)**	**After Oversampling (BorderlineSMOTE)**
Accuracy	0.99	0.96	0.96
Precision	1.00	0.96	0.98
Recall	0.98	0.98	0.94
F1 Score	0.99	0.96	0.96
PPV (Positive Predictive Value)	1.00	0.98	0.98
EOD (Equal Opportunity Difference)	0.03	0.01	0.01
DI (Disparate Impact)	0.43	1.11	1.11

IR (Impact Ratio)	1.60	1.00	1.00

Table 2.14: Results before and after SVM-SMOTE and BorderlineSMOTE on Diabetes Risk Prediction Dataset Using GBM

Table 2.15 presents a comparison of KNN model's performance in Diabetes Risk Prediction Dataset after applying SVM-SMOTE and BorderlineSMOTE.

Machine Learning: KNN on "Diabetes Risk Prediction"			
Metric	**Before Oversampling**	**After Oversampling (SVM-SMOTE)**	**After Oversampling (BorderlineSMOTE)**
Accuracy	0.99	0.91	0.91
Precision	1.00	1.00	1.0
Recall	0.98	0.81	0.81
F1 Score	0.99	0.90	0.90
PPV (Positive Predictive Value)	1.00	1.00	1.00
EOD (Equal Opportunity Difference)	0.03	0.27	0.27
DI (Disparate Impact)	0.43	0.82	0.82
IR (Impact Ratio)	1.60	1.00	1.00

Figure 2.15: Results before and after SVM-SMOTE and BorderlineSMOTE on Diabetes Risk Prediction Dataset Using KNN

2.4.7 Discussion

In the realm of healthcare machine learning, achieving both accuracy and fairness in predictive models is paramount to mitigate biases and ensure equitable healthcare outcomes. Our approach addresses this challenge through the implementation of advanced oversampling techniques, specifically SVM-SMOTE and BorderlineSMOTE. These methods are designed to tackle class imbalances and enhance both the performance metrics—such as accuracy, precision, recall, and F1 Score—and fairness metrics—like Equal Opportunity Difference (EOD), Disparate Impact (DI), and Imbalance Ratio (IR)—of predictive models.

We applied SVM-SMOTE and BorderlineSMOTE to two crucial healthcare datasets: heart disease prediction and diabetes risk prediction. Employing Gradient Boosting Machine (GBM) and K-Nearest Neighbors (KNN) models, we evaluated the impact of these techniques comprehensively across various metrics.

In the Heart Disease Prediction Dataset, our findings revealed substantial improvements in recall metrics, particularly notable with GBM models. For instance, GBM models on the heart disease dataset saw recall increase significantly from 0.75 to 0.88 with SVM-SMOTE, further improving to 0.92 with BorderlineSMOTE. These enhancements underscore the effectiveness of oversampling in enhancing the model's sensitivity to correctly identify positive cases. Conversely, KNN models exhibited more modest improvements or slight declines in performance due to the introduction of synthetic samples.

In terms of fairness metrics, both SVM-SMOTE and BorderlineSMOTE effectively mitigated biases within the heart disease prediction dataset. Disparate Impact (DI) metrics approached the ideal value of 1, indicating improved equity in model predictions across demographic groups. Specifically, for GBM models, DI improved significantly from 1.59 to 0.95 with SVM-SMOTE and further to 0.79 with BorderlineSMOTE. Equal Opportunity Difference (EOD) metrics generally showed improvement with SVM-SMOTE, although results varied with BorderlineSMOTE. For instance, KNN models on the heart disease dataset exhibited a slight increase in EOD from 0.16 to 0.17 with SVM-SMOTE, but a notable decrease to 0.07 with BorderlineSMOTE. This variability highlights the nuanced impact of oversampling techniques on fairness outcomes.

In evaluating trade-offs, SVM-SMOTE typically achieved balanced improvements in both performance and fairness metrics. This makes it suitable for applications where optimizing overall equity in predictive outcomes is essential. In contrast, BorderlineSMOTE prioritized significant enhancements in recall, potentially at the expense of increased disparities measured by metrics like EOD. This approach proves beneficial in scenarios where maximizing sensitivity (recall) is critical, such as in medical diagnostics.

In summary, the choice of oversampling technique should align with specific model requirements and fairness objectives within healthcare machine learning applications. By integrating advanced techniques like SVM-SMOTE and BorderlineSMOTE, we can effectively enhance predictive accuracy while promoting fairness and equity in healthcare decision-making. This approach contributes to mitigating biases and ensuring more reliable and inclusive healthcare outcomes across diverse demographic groups.

2.5 Conclusion

In conclusion, SVM-SMOTE and BorderlineSMOTE proved effective in improving both the performance and fairness of healthcare machine learning models, particularly in heart disease prediction. GBM models showed significant increases in recall with SVM-SMOTE (0.75 to 0.88) and BorderlineSMOTE (0.92), highlighting their ability to accurately identify positive cases. While KNN models exhibited mixed results, both techniques successfully reduced biases, as reflected in improved Disparate Impact metrics. SVM-SMOTE achieved balanced improvements across metrics, whereas BorderlineSMOTE prioritized recall enhancement, potentially impacting fairness measures like Equal Opportunity Difference. Overall, these approaches enhance predictive accuracy and equity in healthcare applications.

Chapter 3

Implementation

3.1 Introduction

Our approach to mitigating bias in healthcare machine learning leverages advanced oversampling techniques such as SVM-SMOTE and BorderlineSMOTE to effectively address class imbalance in datasets. By doing so, we aim to enhance the accuracy and fairness of our predictive models, ensuring that underrepresented classes receive adequate consideration.

In this chapter, we have detailed the comprehensive environment utilized for our implementation, encompassing both hardware and software components. Additionally, we have outlined the libraries employed in our work and provided illustrative screenshots of the source code and user interface. These elements collectively demonstrate the practical steps taken to operationalize our bias mitigation strategies, highlighting the technical foundation and user-friendly design of our solution.

3.2 Environment

3.2.1 Hardware

The study utilized a hardware environment centered around an Intel(R) Core(TM) i5-10210U CPU running at 1.60GHz, complemented by 8 GB of RAM and an SSD for storage. This configuration leveraged the CPU's processing power and the SSD's fast data access to effectively manage large datasets and complex algorithms. With 8 GB of RAM, the system accommodated concurrent execution of multiple model instances and algorithms, ensuring reliable performance and smooth operation throughout the research. This setup proved pivotal in achieving accurate results and maintaining computational efficiency, particularly in the context of healthcare-focused machine learning investigations.

3.2.2 Software

- ***Python***

Python, a high-level programming language, has become vital in scientific computing because of its quick development cycle and ease of program maintenance. Compared to traditional low-level compiled languages, Python is often more efficient for prototyping new concepts. Additionally, Python boasts a variety of high-quality numerical libraries, solidifying its importance in the field of scientific computing [W5].

- ***Jupyter***

Jupyter is an open-source project that provides a web-based interactive computing environment for creating and sharing documents containing live code, equations, visualizations, and narrative text. It supports various programming languages, including Python, R, and Julia. Jupyter notebooks allow users to write and execute code in cells, making it a popular tool for data analysis, machine learning, and scientific research [W6].

- ***Used Libraries***
  - ***NumPy***

NumPy is a Python library that provides flexible and effective support for huge, multidimensional arrays and matrices, as well as numerical computing through a vast array of mathematical functions. It is an essential part of using Python for scientific computing, and it has many uses in physics, data science, engineering, and other disciplines. NumPy is a powerful tool for numerical computations, offering array objects that surpass Python's built-in data structures, enabling vectorized array operations and reducing calculation time. It also offers features like random number generation, Fourier analysis, linear algebra, and integration with other libraries and languages [31].

  - ***Sklearn***

Scikit-learn is a well-liked Python machine-learning library that offers simple implemetations of several well-known algorithms. Its easy distribution is ensured by its smooth integration with Python and reliance solely on NumPy and SciPy. Because Scikit-learn uses compiled code and incorporates C++ libraries for assistance, it is efficient. It is extensively used across many industries, including academics, and is accessible on a number of platforms for both free and a fee. The BSD license under which the library is provided makes it freely available and embraced. [32]

  - ***Pandas***

Pandas is a Python package made to improve data analysis skills by filling the gap between database languages and specialist statistical platforms and Python's general-purpose computing capabilities. Strong capabilities like hierarchical indexing and automated data alignment are available, and functionalities are more strongly linked than in other computer environments. [33]

  - ***Matplotlib***

Matplotlib is a Python library designed for generating 2D plots, spanning from static to animated and interactive visualizations. Widely utilized in scientific computing, it facilitates data exploration and visualization tasks. Offering a rich assortment of 2D plotting capabilities and extensive customization options, Matplotlib empowers users to craft intricate and advanced plots effortlessly. Its compatibility with diverse Python libraries and frameworks further augments its adaptability, allowing for versatile data visualization across various domains [47].

  - ***Seaborn***

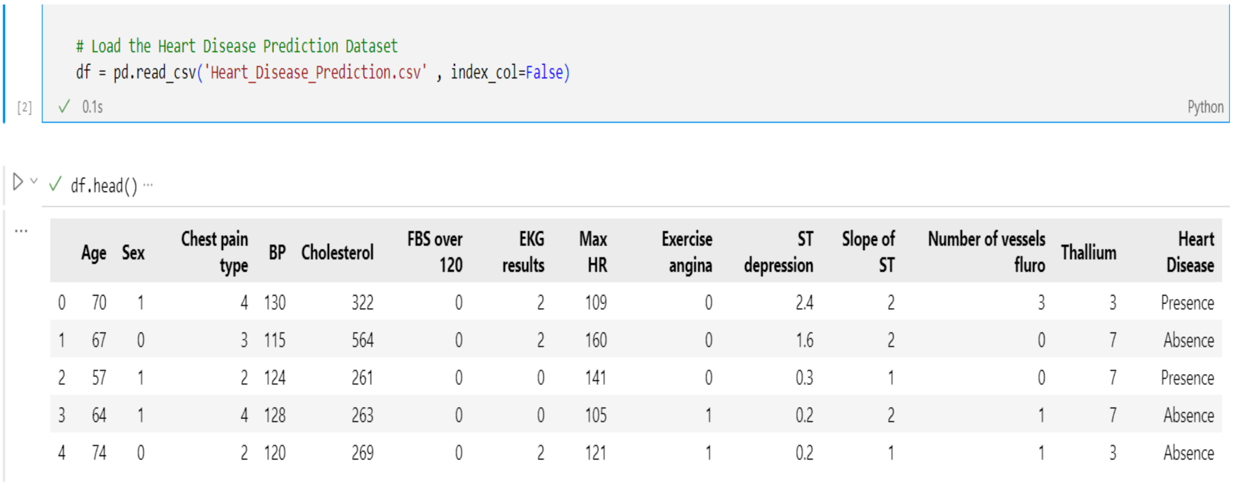
Seaborn is a Python library designed for creating statistical graphics and visualizations. It extends the functionality of matplotlib and integrates seamlessly with pandas data structures.

Seaborn simplifies the process of exploring and comprehending datasets by providing high-level plotting functions that operate directly on dataframes or arrays containing entire datasets. Internally, it handles semantic mapping and statistical aggregation, allowing users to generate informative plots without delving into the intricacies of plotting mechanics. Its dataset-oriented and declarative API enables users to focus on interpreting the meaning of plot elements rather than the technical details of their creation. Thus, Seaborn facilitates effective data exploration and visualization in Python, particularly in statistical and data analysis contexts [W6].

3.3 Implementation

- ***Loading*** Dataset

Figure 3.1 represents how we can load the dataset from a CSV file named ’Heart_Disease_Prediction.csv’ into a Pandas DataFrame called ’df’ and then display the first five rows of the DataFrame:



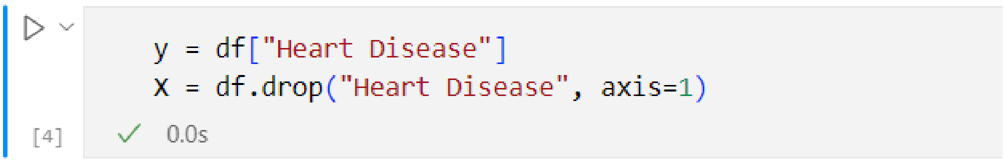
```
# Load the Heart Disease Prediction Dataset
df = pd.read_csv('Heart_Disease_Prediction.csv' , index_col=False)
[2] ✓ 0.1s                                                    Python
```

```
✓ df.head() ···
```

	Age	Sex	Chest pain type	BP	Cholesterol	FBS over 120	EKG results	Max HR	Exercise angina	ST depression	Slope of ST	Number of vessels fluro	Thallium	Heart Disease
0	70	1	4	130	322	0	2	109	0	2.4	2	3	3	Presence
1	67	0	3	115	564	0	2	160	0	1.6	2	0	7	Absence
2	57	1	2	124	261	0	0	141	0	0.3	1	0	7	Presence
3	64	1	4	128	263	0	0	105	1	0.2	2	1	7	Absence
4	74	0	2	120	269	0	2	121	1	0.2	1	1	3	Absence

Figure 3.1: Loading the dataset.

The code in figure 3.2 separates the original DataFrame ’df’ into two parts: X: which contains the features, and y: which contains the corresponding labels Presence (1) and Absence (0).

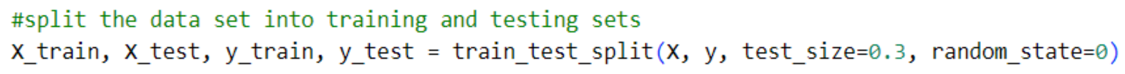


```
y = df["Heart Disease"]
X = df.drop("Heart Disease", axis=1)
[4] ✓ 0.0s
```

Figure 3.2: Dataset separation.

- ***Data Splitting***

We split the data set into training and testing sets using a 70% -30% ratio, as shown in figure 3.3. This ensures that we have a portion of the data reserved for evaluation purposes.

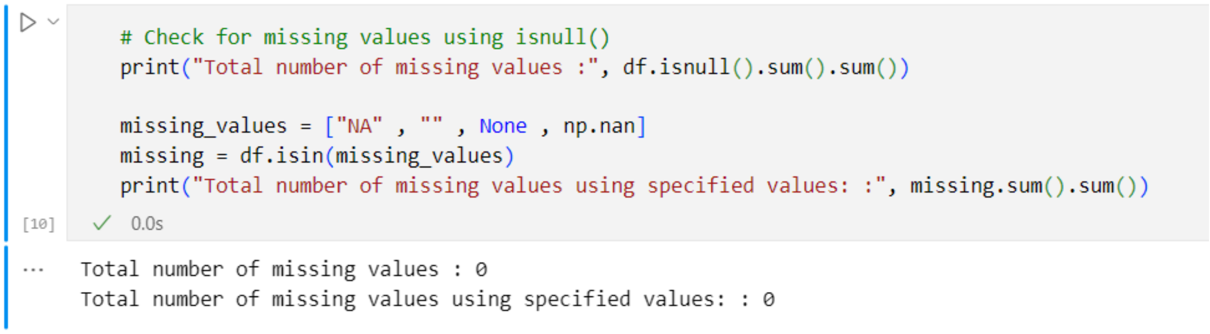


```
#split the data set into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=0)
```

Figure 3.3: Data splitting.

- Checking ***missing*** values

Figure 3.4, we employ the isnull() method to calculate and aggregate all instances of missing values throughout the entire DataFrame. Additionally, we define a custom list of potential missing value indicators ('NA', '', None, np.nan), employing the isin() method to detect and tally these specific missing values within the DataFrame.



```
# Check for missing values using isnull()
print("Total number of missing values :", df.isnull().sum().sum())

missing_values = ["NA" , "" , None , np.nan]
missing = df.isin(missing_values)
print("Total number of missing values using specified values: :", missing.sum().sum())
```

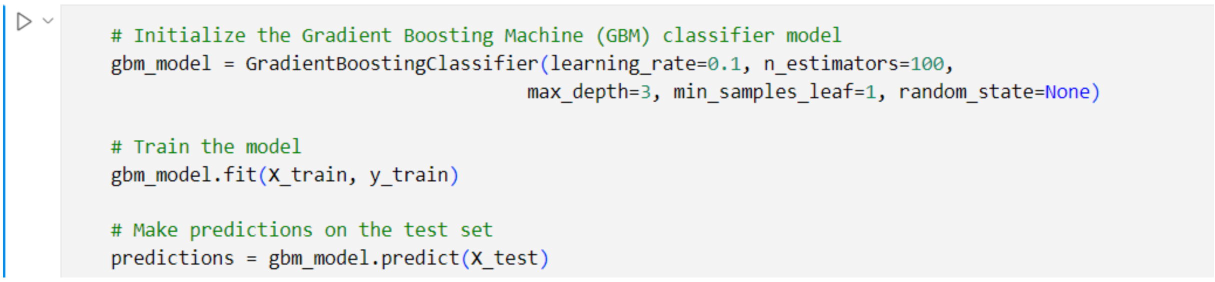
[10] ✓ 0.0s

```
Total number of missing values : 0
Total number of missing values using specified values: : 0
```

Figure 3.4: Checking Missing values implementation.

- ***Gradient*** Boosting Machine (GBM) classifier implementation

The figure 3.5 illustrates a Gradient Boosting Machine (GBM) classifier model with default parameters. It shows the model being trained on the training data (X_train, y_train) and then used to make predictions on the test set (X_test).

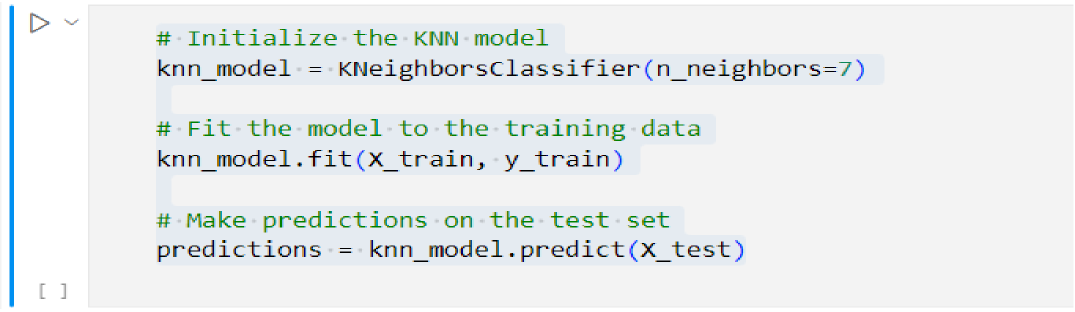


```
# Initialize the Gradient Boosting Machine (GBM) classifier model
gbm_model = GradientBoostingClassifier(learning_rate=0.1, n_estimators=100,
                                       max_depth=3, min_samples_leaf=1, random_state=None)

# Train the model
gbm_model.fit(X_train, y_train)

# Make predictions on the test set
predictions = gbm_model.predict(X_test)
```

Figure 3.5: GBM implementation.

- K-Nearest Neighbors (KNN) classifier implementation

Figure 3.6 illustrates initializing a K-Nearest Neighbors (KNN) classifier model with n_neighbors=7, fitting it to the training data (X_train, y_train), and then using the trained model to make predictions on the test set (X_test).

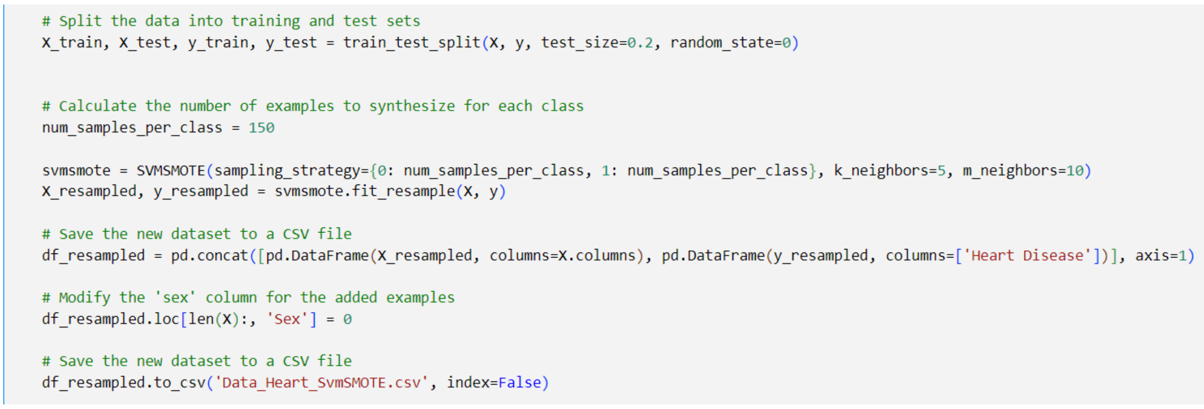


```
# Initialize the KNN model
knn_model = KNeighborsClassifier(n_neighbors=7)

# Fit the model to the training data
knn_model.fit(X_train, y_train)

# Make predictions on the test set
predictions = knn_model.predict(X_test)
```

Figure 3.6: KNN implementation

- SvmSMOTE implementation

In Figure 3.7, the depicted code segment first splits the original dataset into training and test sets using an 80:20 ratio, ensuring a stratified division to maintain class distribution integrity. Subsequently, SVMSMOTE (Support Vector Machine Synthetic Minority Over-sampling Technique) is employed to balance the dataset by synthesizing additional samples for the minority class (heart disease cases) to match the number of samples in the majority class (non-heart disease cases). The oversampling is configured with parameters k_neighbors=5 and m_neighbors=10 to guide the selection and synthesis of new instances based on nearest neighbors. The resulting resampled data is consolidated into a pandas DataFrame, where adjustments are made to the 'Sex' column for the newly generated samples. Finally, the augmented dataset is saved as 'Data_Heart_SvmSMOTE.csv'.

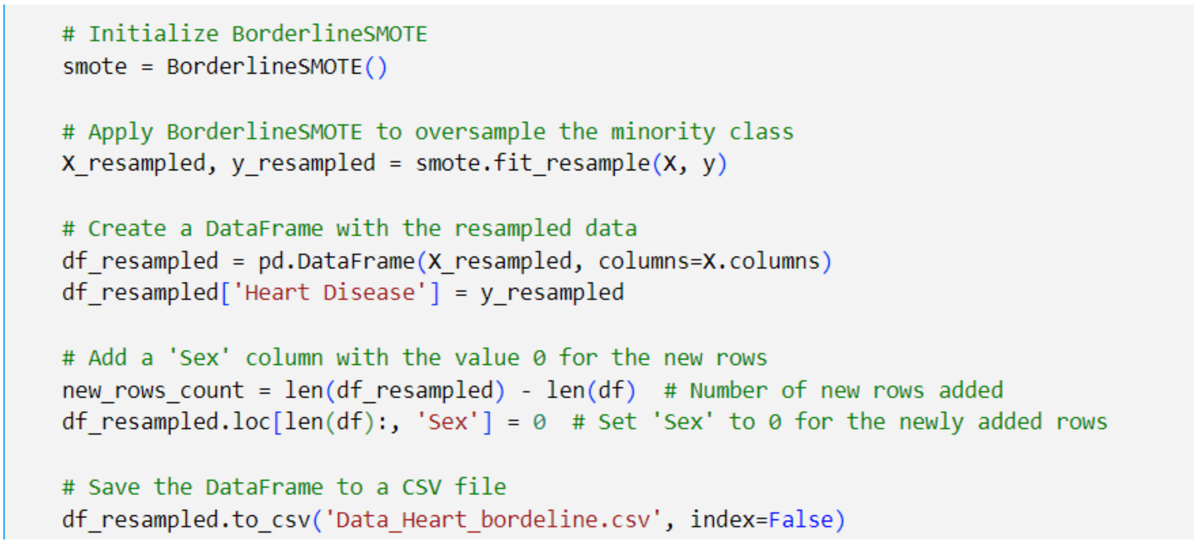


```
# Split the data into training and test sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=0)


# Calculate the number of examples to synthesize for each class
num_samples_per_class = 150

svmsmote = SVMSMOTE(sampling_strategy={0: num_samples_per_class, 1: num_samples_per_class}, k_neighbors=5, m_neighbors=10)
X_resampled, y_resampled = svmsmote.fit_resample(X, y)

# Save the new dataset to a CSV file
df_resampled = pd.concat([pd.DataFrame(X_resampled, columns=X.columns), pd.DataFrame(y_resampled, columns=['Heart Disease'])], axis=1)

# Modify the 'sex' column for the added examples
df_resampled.loc[len(X):, 'Sex'] = 0

# Save the new dataset to a CSV file
df_resampled.to_csv('Data_Heart_SvmSMOTE.csv', index=False)
```

Figure 3.7: SvmSMOTE implementation

- BorderlineSMOTE implementation

In Figure 3.8, the code snippet demonstrates the application of BorderlineSMOTE for oversampling the minority class in a heart disease prediction dataset. Initially, BorderlineSMOTE is initialized and applied to generate synthetic samples, ensuring a balanced representation of both classes. The resulting resampled data is structured into a pandas DataFrame, where additional rows are appended with a 'Sex' column initialized to 0 for newly synthesized instances. Finally, the augmented dataset is saved as 'Data_Heart_bordeline.csv'.



```
# Initialize BorderlineSMOTE
smote = BorderlineSMOTE()

# Apply BorderlineSMOTE to oversample the minority class
X_resampled, y_resampled = smote.fit_resample(X, y)

# Create a DataFrame with the resampled data
df_resampled = pd.DataFrame(X_resampled, columns=X.columns)
df_resampled['Heart Disease'] = y_resampled

# Add a 'Sex' column with the value 0 for the new rows
new_rows_count = len(df_resampled) - len(df)  # Number of new rows added
df_resampled.loc[len(df):, 'Sex'] = 0  # Set 'Sex' to 0 for the newly added rows

# Save the DataFrame to a CSV file
df_resampled.to_csv('Data_Heart_bordeline.csv', index=False)
```

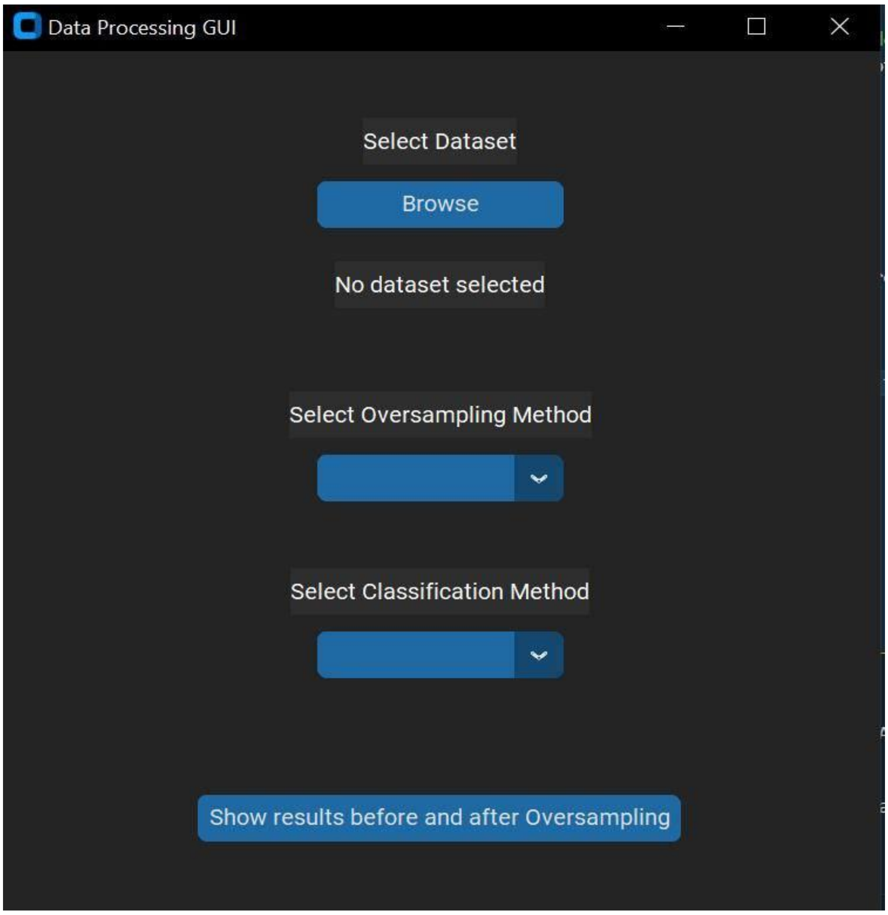
Figure 3.8: BorderlineSMOTE implementation

3.4 User interface

In the main interface, we have the following components:

- **Select Dataset:** Users can choose a dataset by clicking the “Browse” button,The text area below it indicates whether a dataset has been selected.
- **Oversampling Method:** A dropdown menu allows users to pick an oversampling method.
- **Classification Method:** Another dropdown menu lets users select a classification method.
- **Results Display:** At the bottom, there’s a button labeled “Show results before and after Oversampling.”

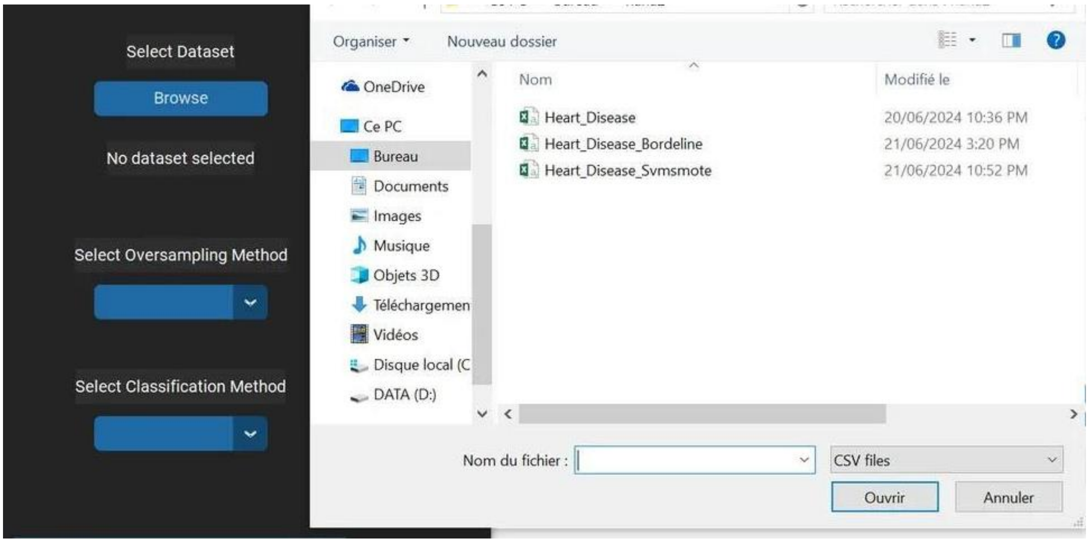
The figure 3.9 illustrate the main interface


Data Processing GUI
Select Dataset
Browse
Select Oversampling Method
Select Classification Method
Show results before and after Oversampling

No dataset selected

Figure 3.9: The main interface.

The figure 3.10 illustrates how tot "Select the Dataset", the "Browse" button for selecting files from the user's computer.

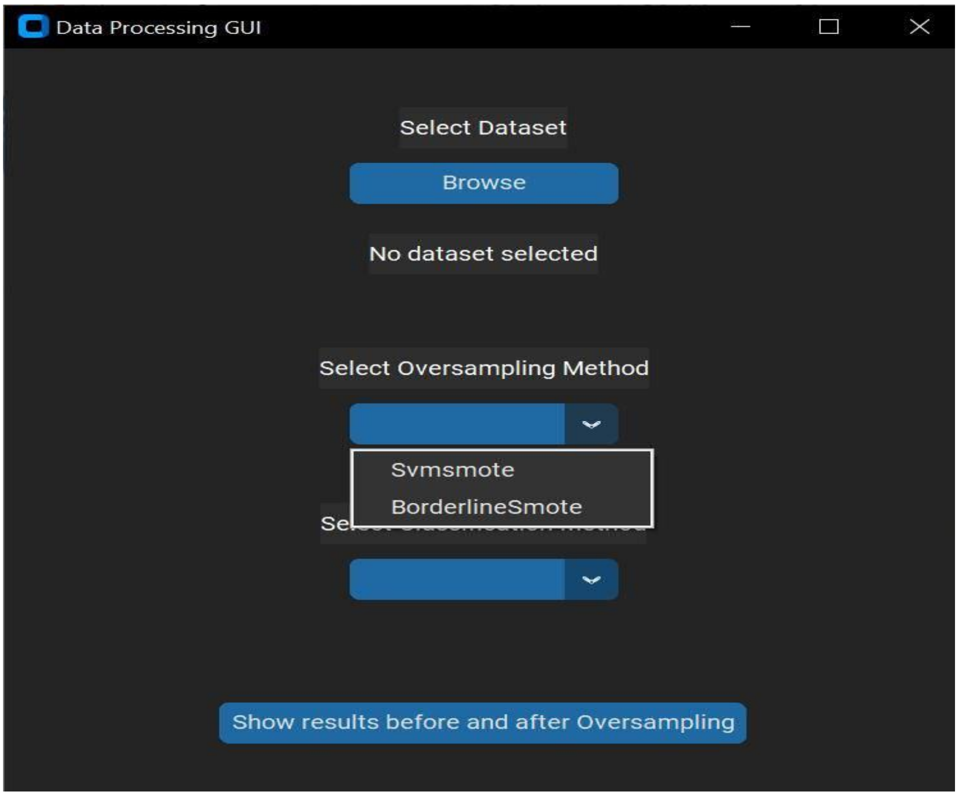


No dataset selected

Nom	Modifié le
Heart_Disease	20/06/2024 10:36 PM
Heart_Disease_Bordeline	21/06/2024 3:20 PM
Heart_Disease_Svmsmote	21/06/2024 10:52 PM

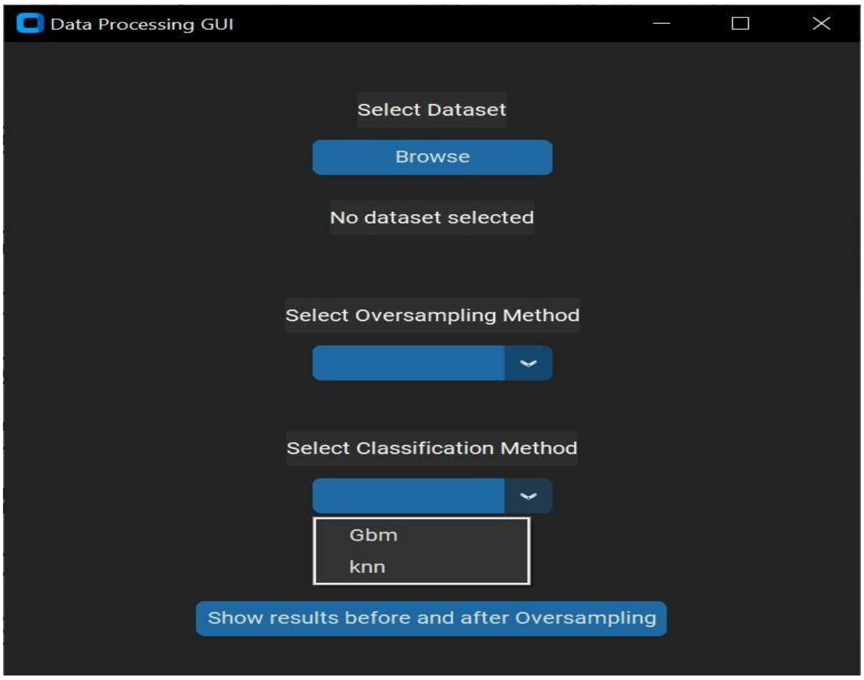
Figure 3.10: Select Dataset.

The figure 3.11 illustrates the “select oversampling method” and the figure 3.12 “select classification method”


Data Processing GUI
Select Dataset
Browse
Select Oversampling Method
Svmsmote
BorderlineSmote
Se
Show results before and after Oversampling

No dataset selected

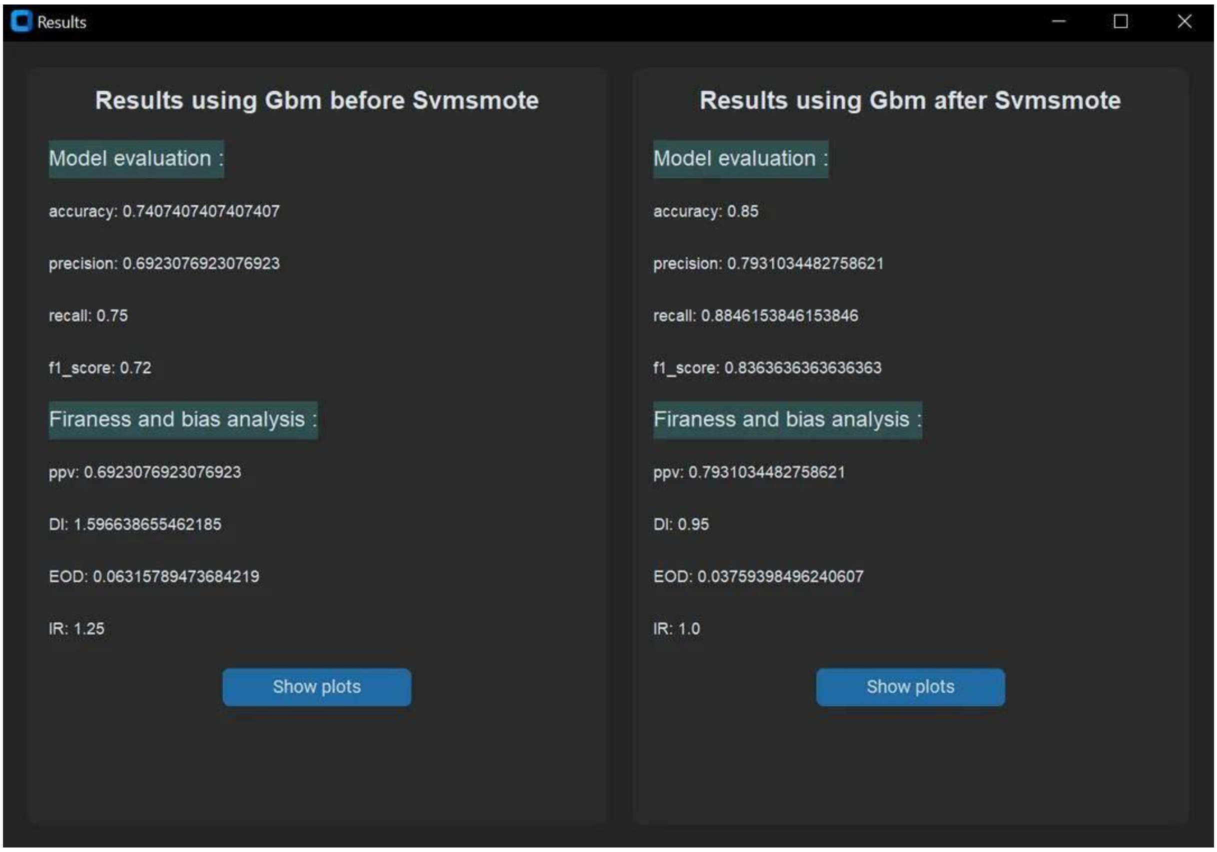
Figure 3.11: select oversampling method


Data Processing GUI
Select Dataset
Browse
Select Oversampling Method
Select Classification Method
Gbm
knn
Show results before and after Oversampling

No dataset selected

Figure 3.12: select classification method.

Figure 3.13 illustrate the results before and after oversampling.



Results using Gbm before Svmsmote

Model evaluation :

accuracy: 0.7407407407407407

precision: 0.6923076923076923

recall: 0.75

f1_score: 0.72

Firaness and bias analysis :

ppv: 0.6923076923076923

DI: 1.596638655462185

EOD: 0.06315789473684219

IR: 1.25

Results using Gbm after Svmsmote

Model evaluation :

accuracy: 0.85

precision: 0.7931034482758621

recall: 0.8846153846153846

f1_score: 0.8363636363636363

Firaness and bias analysis :

ppv: 0.7931034482758621

DI: 0.95

EOD: 0.03759398496240607

IR: 1.0

Figure 3.13: The results before and after oversampling

3.4 Conclusion

In conclusion, the implementation of oversampling techniques like SVM-SMOTE and BorderlineSMOTE represents a critical advancement in addressing class imbalance within healthcare machine learning. In this chapter, we presented the environment, both hardware and software, the libraries used in the implementation, and provide some screenshots of the source code and user interface.

General conclusion

In conclusion, the strategic application of machine learning techniques, particularly through the use of oversampling methods like SVM-SMOTE and BorderlineSMOTE, represents a critical advancement in addressing the challenge of class imbalance in biomedical datasets for heart disease and diabetes risk prediction. Our study reveals significant improvements across various metrics, underscoring the efficacy of these methods in enhancing both predictive accuracy and fairness in model outcomes.

For heart disease prediction using Gradient Boosting Machine (GBM), both SVM-SMOTE and BorderlineSMOTE demonstrated substantial enhancements in key metrics. After oversampling with BorderlineSMOTE, accuracy remained high at 0.85, while recall improved notably to 0.92, indicating better identification of positive cases. Precision and F1 score also showed robust improvements, highlighting the models' increased capability to correctly classify instances across all classes. Importantly, fairness metrics such as Equal Opportunity Difference (EOD) and Disparate Impact (DI) improved significantly with BorderlineSMOTE, contributing to more equitable predictions across demographic groups.

Similarly, in the context of diabetes risk prediction with K-Nearest Neighbors (KNN), oversampling with BorderlineSMOTE led to improvements in accuracy (0.71), precision (0.70), recall (0.72), and F1 score (0.71). While the improvements were more modest compared to GBM, the introduction of synthetic samples notably reduced EOD and DI values, indicating reduced bias in model predictions.

Future work should involve validating the proposed oversampling techniques on larger and more diverse biomedical datasets. This would help to generalize the findings and confirm their applicability across various medical conditions and population demographics.

We also propose to explore the development and evaluation of hybrid oversampling methods that combine multiple techniques (e.g., combining SMOTE variants with adversarial training) to further improve the balance and fairness of datasets, potentially leading to even more robust predictive models.

Bibliography

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

[2] Fletcher, R. R., Nakeshimana, A. et Olubeko, O. (2021).Addressing fairness, bias, and appropriate use of artificial intelligence and machine learning in global health. Frontiers in artificial intelligence, 3:6

[3] Pagano, T. P., Loureiro, R. B., Lisboa, F. V. N., Cruz, G. O. R., Peixoto, R. M., Guimar˜aes, G. A. d. S., Santos, L. L. d., Araujo, M. M., Cruz, M., de Oliveira, E. L. S. et al. (2022). Bias and unfairness in machine learning models: a systematic literature review. arXiv preprint arXiv:2202.08176.

[4] Iwasi´nski, L. (2020). Social implications of algorithmic bias. [Jiang et al., 2023] Jiang, S., Han, R., Chakrabarty, K., Page, D., Stead, W. W. et Zhang, A. R. (2023). Timeline registration for electronic health records. AMIA Summits on Translational Science Proceedings, 2023:29

[5] Drukker, K., Chen, W., Gichoya, J., Gruszauskas, N., Kalpathy-Cramer, J., Koyejo, S., Myers, K., S´a, R. C., Sahiner, B., Whitney, H. et al. (2023). Toward fairness in artificial intelligence for medical image analysis: identification and mitigation of potential biases in the roadmap from data collection to model deployment. Journal of Medical Imaging, 10(6):061104–061104.

[6] Pagano, T. P., Loureiro, R. B., Lisboa, F. V., Peixoto, R. M., Guimar˜aes, G. A., Cruz, G. O., Araujo, M. M., Santos, L. L., Cruz, M. A., Oliveira, E. L. et al. (2023). Bias and unfairness in machine learning models: a systematic review on datasets, tools, fairness metrics, and identification and mitigation methods. Big data and cognitive computing, 7(1):15.

[7] Habehh, H. et Gohel, S. (2021). Machine learning in healthcare. Current genomics, 22(4):291

[8] Althubaiti, A. (2016). Information bias in health research: definition, pitfalls, and adjustment methods. Journal of multidisciplinary healthcare, pages 211–217.

[9] Panch, T., Mattie, H. et Atun, R. (2019). Artificial intelligence and algorithmic bias: implications for health systems. Journal of global health, 9(2).

[10] Pagano, T. P., Loureiro, R. B., Lisboa, F. V., Peixoto, R. M., Guimar˜aes, G. A., Cruz, G. O., Araujo, M. M., Santos, L. L., Cruz, M. A., Oliveira, E. L. et al. (2023). Bias and unfairness in machine learning models: a systematic review on datasets, tools, fairness metrics, and identification and mitigation methods. Big data and cognitive computing, 7(1):15.

[11] Park, Y., Hu, J., Singh, M., Sylla, I., Dankwa-Mullan, I., Koski, E. et Das, A. K. (2021). Comparison of methods to reduce bias from clinical prediction models of postpartum depression. JAMA network open, 4(4):e213909–e213909

[12] Juhn, Y. J., Ryu, E., Wi, C.-I., King, K. S., Malik, M., Romero-Brufau, S., Weng, C., Sohn, S., Sharp, R. R. et Halamka, J. D. (2022). Assessing socioeconomic bias in machine learning algorithms in health care: a case study of the

houses index. Journal of the American Medical Informatics Association, 29(7):1142–1151

[13] R¨o¨osli, E., Bozkurt, S. et Hernandez-Boussard, T. (2022). Peeking into a black box, the fairness and generalizability of a mimic-iii benchmarking model. Scientific Data, 9(1):24

[14] Wang, H., Li, Y., Naidech, A. et Luo, Y. (2022). Comparison between machine learning methods for mortality prediction for sepsis patients with different social determinants. BMC medical informatics and decision making, 22(Suppl2):156

[15] Meng, C., Trinh, L., Xu, N., Enouen, J. et Liu, Y. (2022). Interpretability and fairness evaluation of deep learning models on mimic-iv dataset. Scientific Reports, 12(1):7166

[16] Raza, S. et Bashir, S. R. (2023). Auditing icu readmission rates in an clinical database: An analysis of risk factors and clinical outcomes. In 2023 IEEE 11th International Conference on Healthcare Informatics (ICHI), pages 722–726. IEEE

[17] Allen, A., Mataraso, S., Siefkas, A., Burdick, H., Braden, G., Dellinger, R. P., McCoy, A., Pellegrini, E., Hoffman, J., Green-Saxena, A. et al. (2020). A racially unbiased, machine learning approach to prediction of mortality: algorithm development study. JMIR public health and surveillance, 6(4):e22400.

[18] Karlsson, I. et Bostr¨om, H. (2014). Handling sparsity with random forests when predicting adverse drug events from electronic health records. In 2014 ieee international conference on healthcare informatics, pages 17–22.IEEE

[19] Lee, T., Wollstein, G., Madu, C. T., Wronka, A., Zheng, L., Zambrano, R., Schuman, J. S. et Hu, J. (2023). Reducing ophthalmic health disparities through transfer learning: A novel application to overcome data inequality.Translational Vision Science & Technology, 12(12):2–2.

[20] Li, C., Jiang, X. et Zhang, K. (2024). A transformer-based deep learning approach for fairly predicting post-liver transplant risk factors. Journal of Biomedical Informatics, 149:104545.

[21] Zhu, X., Hurtado, J. et Tao, H. (2017). Localized sampling for hospital re-admission prediction with imbalanced sample distributions. In 2017 International Joint Conference on Neural Networks (IJCNN), pages 4571–4578. IEEE

[22] Huda, N. M. et al. (2019). Design of istitaah classification system based on machine learning using imbalanced dataset. In 2019 7th International Conference on Cyber and IT Service Management (CITSM), volume 7, pages 1–6. IEEE

[23] Hee, K. (2017). Is data quality enough for a clinical decision?: Apply machine learning and avoid bias. In 2017 IEEE International Conference on Big Data (Big Data), pages 2612–2619. IEEE.

[24] Getz, K., Hubbard, R. A. et Linn, K. A. (2023). Performance of multiple imputation using modern machine learning methods in electronic health records data. Epidemiology, 34(2):206–215.

[25] Yin, K., Qian, D. et Cheung, W. K. (2023). Patnet: Propensity-adjusted temporal network for joint imputation and prediction using binary ehrs with observation bias. IEEE Transactions on Knowledge and Data Engineering.

[26] Cui, S., Pan, W., Zhang, C. et Wang, F. (2023). Bipartite ranking fairness through a model agnostic ordering adjustment. IEEE Transactions on Pattern Analysis and Machine Intelligence

[27] Jiang, S., Han, R., Chakrabarty, K., Page, D., Stead, W. W. et Zhang, A. R. (2023). Timeline registration for electronic health records. AMIA Summits on Translational Science Proceedings, 2023:291.

[28] Jiang, S., Han, R., Chakrabarty, K., Page, D., Stead, W. W. et Zhang, A. R. (2023). Timeline registration for electronic health records. AMIA Summits on Translational Science Proceedings, 2023:291.

[29] Wolk, D. M., Lanyado, A., Tice, A. M., Shermohammed, M., Kinar, Y., Goren, A., Chabris, C. F., Meyer, M. N., Shoshan, A. et Abedi, V. (2022). Prediction of influenza complications: development and validation of a machine learning prediction model to improve and expand the identification of vaccine-hesitant patients at risk of severe influenza complications. Journal of Clinical Medicine, 11(15):4342.

[30] Abay, A., Zhou, Y., Baracaldo, N., Rajamoni, S., Chuba, E. et Ludwig, H. (2020). Mitigating bias in federated learning. arXiv preprint arXiv:2012.02447.

[31] Walt et al., 2011] Van Der Walt, S., Colbert, S. C. et Varoquaux, G. (2011). The numpy array: a structure for efficient numerical computation. Computing in science & engineering, 13(2):22–30

[32] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V. et al. (2011). Scikit-learn: Machine learning in python. the Journal of machine Learning research, 12:2825–2830.

[33] McKinney, W. et al. (2011). pandas: a foundational python library for data analysis and statistics. Python for high performance and scientific computing, 14(9):1–9.

[34] Mitl ̈ohner, J., Neumaier, S., Umbrich, J. et Polleres, A. (2016). Characteristics of open data csv files. In 2016 2nd International Conference on Open and Big Data (OBD), pages 72–79. IEEE

[35] Li, F., Wu, P., Ong, H. H., Peterson, J. F., Wei, W.-Q. et Zhao, J. (2023). Evaluating and mitigating bias in machine learning models for cardiovascular disease prediction. Journal of Biomedical Informatics, 138:104294.

[36] Saeed, S., Alireza, B., Mohamed, E., & Ahmed, N. (2015). Evidence based emergency medicine part 2: positive and negative predictive values of diagnostic tests.

[37] Mokoatle, M., Coleman, T., & Mokilane, P. (2023, November). A comparative study of over-sampling techniques as applied to seismic events. In *Southern African Conference for Artificial Intelligence Research* (pp. 331-345). Cham: Springer Nature Switzerland.

[38] Nemade, B., Bharadi, V., Alegavi, S. S., & Marakarkandy, B. (2023). A Comprehensive Review: SMOTE-Based Oversampling Methods for Imbalanced Classification Techniques, Evaluation, and Result Comparisons. *International Journal of Intelligent Systems and Applications in Engineering*, *11*(9s), 790-803.

[39] Ayyadevara, V. Kishore, and V. Kishore Ayyadevara. "Gradient boosting machine." *Pro machine learning algorithms: A hands-on approach to implementing algorithms in python and R* (2018): 117-134.

[40] Lu, Haihao, and Rahul Mazumder. "Randomized gradient boosting machine." *SIAM Journal on Optimization* 30.4 (2020): 2780-2808.

[41] Ray, Susmita. "A quick review of machine learning algorithms." *2019 International conference on machine learning, big data, cloud and parallel computing (COMITCon)*. IEEE, 2019.

[42] Egwom, Onyinyechi Jessica, et al. "An LDA–SVM machine learning model for breast cancer classification." *BioMedInformatics* 2.3 (2022): 345-358.

[43] Vishwanathan, S. V. M., and M. Narasimha Murty. "SSVM: a simple SVM algorithm." *Proceedings of the 2002 International Joint Conference on Neural Networks. IJCNN'02 (Cat. No. 02CH37290)*. Vol. 3. IEEE, 2002.

[44] Singh, Suryabhan Pratap, and Umesh Chandra Jaiswal. "RETRACTED ARTICLE: Classification of audio signals using SVM-WOA in Hadoop map-reduce framework." *SN Applied Sciences* 2.12 (2020): 2044.

[45] Moldagulova, A., & Sulaiman, R. B. (2017, May). Using KNN algorithm for classification of textual documents. In *2017 8th international conference on information technology (ICIT)* (pp. 665-671). IEEE.

[46] Xing, W., & Bei, Y. (2019). Medical health big data classification based on KNN classification algorithm. *Ieee Access*, *8*, 28808-28819.

[47] Zhu, R., Guo, Y., & Xue, J. H. (2020). Adjusting the imbalance ratio by the dimensionality of imbalanced data. *Pattern Recognition Letters*, *133*, 217-223.

[48] Li, J., Zhu, Q., Wu, Q., & Fan, Z. (2021). A novel oversampling technique for class-imbalanced learning based on SMOTE and natural neighbors. *Information Sciences*, *565*, 438-455.

[49] Öztornaci, R. O., Syed, H., Morris, A. P., & Taşdelen, B. (2023). The use of class imbalanced learning methods on ULSAM data to predict the case–control status in genome-wide association studies. *Journal of Big Data*, *10*(1), 174.

[50] Mohammed, R., Rawashdeh, J., & Abdullah, M. (2020, April). Machine learning with oversampling and undersampling techniques: overview study and experimental results. In *2020 11th international conference on information and communication systems (ICICS)* (pp. 243-248). IEEE.

[51] Al Majzoub, H., Elgedawy, I., Akaydın, Ö., & Köse Ulukök, M. (2020). HCAB-SMOTE: A hybrid clustered affinitive borderline SMOTE approach for imbalanced data binary classification. *Arabian Journal for Science and Engineering*, *45*(4), 3205-3222.

Webography

[W1] https://datascience.cancer.gov/news-events/blog/trusting-data-look-data-bias.Last access to the site: April 2, 2024.

[W2] https://www.kaggle.com/datasets/rishidamarla/heart-disease-prediction. Last access to the site: Juin 05, 2024.

[W3] https://www.kaggle.com/datasets/ishandutta/early-stage-diabetes-risk-prediction-dataset. Last access to the site: Juin 05, 2024.

[W4] https://docs.python.org/3/library/csv.html. Last access to the site: Juin 05, 2024.

[W5] https://www.python.org/about/ . Last access to the site: Juin 08, 2024.

[W6] Project Jupyter | About Us. . Last access to the site: Juin 08, 2024.

[W7] https://seaborn.pydata.org/tutorial/introduction Last access to the site: Juin 21, 2024.