Independent University **Bangladesh (IUB)**

IUB Academic Repository

Computer Science and Engineering | Undergraduate Thesis

2026-04

Empirical Evidence on Data Leakage from Improper Resampling Practices in Diabetes Care

Dip, Solayman Alam

IUB

https://ar.iub.edu.bd/handle/11348/1201

Downloaded from IUB Academic Repository



Empirical Evidence on Data Leakage from Improper Resampling Practices in Diabetes Care

April 2026

Prepared by:

Solayman Alam Dip
ID: 2221430

Md. Tahmid Hossain
ID: 2220557

Fatin Fawjia Tanha
ID: 2110414

Department of Computer Science and Engineering
Independent University, Bangladesh

Supervised by

Ashraful Islam, PhD
Assistant Professor
Department of Computer Science and Engineering
Independent University, Bangladesh

Attestation

We are aware of the fact that plagiarism is strictly prohibited and it conflicts with our university's rules and regulations. We guarantee the authenticity of our work. We have used some copyrighted material and models of others in our project, which have been properly cited following the international standards and proper guidelines.

Author Name: .....

Signature: .....

Author Name: .....

Signature: .....

Author Name: .....

Signature: .....

Evaluation Committee

Supervision Panel

.....	.....
Dr. Ashraful Islam	Dr. M Ashraful Amin
Supervisor	Co-supervisor

Examiners

.....	.....
Dr. internal (Internal)	Mr. internal 2 (Internal)

.....
Mr. external (External Examiner)

Office Use

.....	
Dr. AKM Mahbubur Rahaman	.....
Program Coordinator	Head of the Department

.....
Prof. Dr. M. Arshad Momen
Director, Office of the Graduate Studies, Research and Industry Relations

.....
Library

Acknowledgement

We would like to express our sincere gratitude to our respected supervisor, Dr. Ashraful Islam, and co-supervisor, Prof. Dr. M Ashraful Amin from Independent University, Bangladesh. Their dedicated guidance, insightful suggestions, remarkable patience, and the generous investment of his valuable time played a pivotal role in the successful completion of our project. Our Professor's expertise and mentorship greatly enhanced the quality and depth of our work, and we are truly appreciative of his unwavering support throughout our academic journey.

Additionally, our heartfelt thanks go to the Center for Computational & Data Sciences (CCDS) for their extraordinary help and consistent support. The collaborative and resourceful environment provided by CCDS significantly contributed to the development and execution of our project. Moreover, the encouragement and assistance have been instrumental in shaping our academic and research fields. We are grateful for the opportunities and resources extended to us during our research, which undoubtedly enriched our learning experience.

We acknowledge and appreciate the contributions of all individuals who supported us, both directly and indirectly, during this endeavor.

Abstract

Machine learning (ML) is a popular tool for diabetes prediction. Medical datasets, however, contain a majority of healthy patients and a minority of diabetic patients, creating a severe class imbalance. Researchers use synthetic resampling to fix this imbalance. A common error is applying this resampling before separating the test data. This pre-split approach causes data leakage and creates a false illusion of high performance. Furthermore, studies depend on standard accuracy, a metric that often hides the failure of a model to detect actual diabetic cases.

This study examined data leakage, metric reliability, and model explainability across three public datasets: PIMA indians diabetes dataset, Behavioral Risk Factor Surveillance System 2015 (BRFSS-2015), and Diahealth. The research compared pre-split and post-split resampling pipelines using eXtreme Gradient Boosting (XGBoost), Light Gradient Boosting Machine (LightGBM), and Naive Bayes models. SHAP (SHapley Additive exPlanations) was then applied to evaluate how data leakage alters the clinical features the models use to identify diabetic patients.

The experiments prove that pre-split resampling inflates performance scores by 10 to 20%. The post-split pipelines generated lower numbers but provided an honest assessment of real-world capability. The results also demonstrate that standard accuracy is a misleading metric for imbalanced data. The Matthews Correlation Coefficient (MCC) and Recall offer a safer and more realistic evaluation. At the end, the SHAP analysis revealed that data leakage distorts the internal logic of the model. Pre-split pipelines prioritized noisy or indirect variables, while the corrected post-split pipelines isolated expected clinical markers like Glucose, Body Mass Index (BMI), and Age.

Resampling must occur after the train-test split to ensure a valid evaluation. To guarantee patient safety, future diagnostic research needs to prioritize MCC over standard accuracy and incorporate explainability tools to verify clinical reasoning.

Contents

1 Introduction **14**
- 1.1 Background and Motivation . . . 14
  - 1.1.1 Diabetes Mellitus and Its Global Impact . . . 14
  - 1.1.2 Advancements and Trends in Machine Learning for Diabetes Prediction . . . 14
  - 1.1.3 Role of Different Machine Learning Models . . . 15
  - 1.1.4 Challenges in Diabetes Prediction Using Machine Learning . . . 16
  - 1.1.5 Motivation of This Study . . . 16
- 1.2 Problem Statement . . . 17
  - 1.2.1 Issues in Data Quality and Model Performance in Diabetes Prediction Using ML . . . 17
  - 1.2.2 Research Gap . . . 18
- 1.3 Research Questions (RQ) . . . 19
- 1.4 Research Objectives . . . 19
- 1.5 Contribution of the Study . . . 19
- 1.6 Report Outline . . . 20

2 Literature Review **22**
- 2.1 Diabetes and Clinical Predictors . . . 22
- 2.2 Machine Learning in Diabetes Prediction: An Overview . . . 23
- 2.3 Common Datasets Used . . . 25
- 2.4 Resampling Techniques in Imbalanced Learning . . . 26
- 2.5 Evaluation Metrics in imbalanced Medical Datasets . . . 27
- 2.6 Explainability in Medical ML (XAI) . . . 29

3 Materials and Methodology **30**
- 3.1 Research Design Overview . . . 30
- 3.2 Dataset description . . . 32
  - 3.2.1 PIMA Indians Diabetes Dataset . . . 32
  - 3.2.2 BRFSS 2015 . . . 33
  - 3.2.3 DiaHealth . . . 36
- 3.3 Pre-Processing . . . 39
  - 3.3.1 PIMA Indian Diabetes Dataset . . . 39
  - 3.3.2 BRFSS-2015 . . . 41
  - 3.3.3 Diahealth . . . 41
- 3.4 Feature selection . . . 42
  - 3.4.1 PIMA Indians Diabetes Dataset . . . 42
  - 3.4.2 BRFSS-2015 Dataset . . . 44
  - 3.4.3 DiaHealth Dataset . . . 45
- 3.5 Resampling techniques . . . 47

3.5.1 Synthetic Minority Over-sampling Technique (SMOTE) 47
3.5.2 Adaptive Synthetic Sampling (ADASYN) 47
3.5.3 Hybrid Resampling: SMOTETomek and SMOTEENN 47
3.6 Machine Learning Algorithms 48
3.6.1 Gaussian Naïve Bayes 48
3.6.2 XGBoost (eXtreme Gradient Boosting) 48
3.6.2.1 LightGBM (Light Gradient Boosting Machine) 49
3.7 Evaluation metrics 49
3.8 Explainability with SHAP 51
3.8.1 Theoretical Foundation: Shapley Values and Additive Attribution . 51
3.8.2 Algorithm-Specific Explanations: TreeSHAP and KernelSHAP . . . 51
3.8.3 Application in this Research 52
3.9 Experimental setups & Implementation details 53
3.9.1 Hardware and Software Environment 53
3.9.2 Hyperparameter Configuration 53
3.9.3 Execution Protocol 54

4 Results and Analysis **55**
4.1 Pre-split Prediction Result 55
4.1.1 PIMA Indian Diabetes Dataset (PIDD) Prediction Result 55
4.1.2 BRFSS-2015 Prediction Result 68
4.1.3 DiaHealth Prediction Result 75
4.2 Post-split Prediction Result 83
4.2.1 PIMA Indian Diabetes Dataset (PIDD) Prediction Result 84
4.2.2 BRFSS-2015 Prediction Result 101
4.2.3 DiaHealth Prediction Result 110
4.3 Comparative analysis: Post-split VS Pre-split 119
4.4 Metric Comparison and Clinical Utility 122
4.4.1 Accuracy vs. Clinical Metrics 122
4.4.2 Recommended Metric Protocol 122
4.5 Impact of Resampling on Data Distribution 123
4.6 SHAP-Based Feature Importance Analysis 131
4.6.1 PIMA indian dataset SHAP Analysis 131
4.6.2 Diahealth SHAP Analysis 134
4.6.3 BRFSS-2015 SHAP Analysis 138

5 Discussion of Key Findings **143**
5.1 Interpreting the Impact of Pre-Split Resampling (RQ1) 143
5.2 Metric Selection and Clinical Trust (RQ2) 143
5.3 What SHAP Reveals About the Corrected Model (RQ3) 144

6 Threats to Validity **147**
6.1 Internal Validity 147
6.1.1 Randomness in Train-Test Splits 147
6.1.2 Pre-processing Choices and Their Influence on Results 147
6.1.3 Hyperparameter Configuration 147
6.2 External Validity 148
6.2.1 Dataset-Specific Demographics and Population Scope 148
6.2.2 Generalizability to Other Algorithms 148
6.2.3 Generalizability to Other Medical Prediction Tasks 148
6.3 Construct Validity 149

6.3.1 Measuring Data Leakage Through Metric Inflation 149
6.3.2 Evaluating Clinical Utility Through Standard Metrics 149
6.3.3 SHAP as a Measure of Feature Importance 149
6.4 Conclusion Validity 150
6.4.1 Absence of Formal Statistical Significance Testing 150
6.4.2 Limited Cross-Validation in the Main Pipeline 150
6.4.3 Scope of the Resampling Techniques Evaluated 150
6.5 Summary 151

7 Limitations **152**
7.1 Dataset Constraints and Generalizability 152
7.2 Explainability Limitations (SHAP) 152
7.3 Lack of Prospective Clinical Validation 152

8 Ethical Considerations **153**
8.1 Scientific Integrity in Performance Reporting 153
8.2 Clinical Consequences of Misclassification 153
8.3 Algorithmic Fairness and Demographic Bias 153

9 Conclusion and Future Work **155**
9.1 Summary of Findings 155
9.2 Research Contributions 156
9.3 Future Work 156

Bibliography **156**

List of Figures

3.1 Methodology diagram for Post-split resampling pipeline 31
3.2 Methodology diagram for Pre-split resampling pipeline 31
3.3 Feature selection graphs of the PIMA indian dataset: correlation-based and mutual information-based selection results. 43
3.4 Correlation-based feature selection graph of the BRFSS dataset 44
3.5 Mutual information-based feature selection graph of the BRFSS dataset . . 45
3.6 Feature selection graphs of the DiaHealth dataset: correlation-based and mutual information-based selection results. 46

4.1 Data Distribution for Imputation & Outlier removed without feature selection 55
4.2 ROC curves demonstrating model performance on the PIMA indian dataset for pre-split data resampling Imputation and Outlier removed without feature selection. 56
4.3 Data Distribution for Elimination & Outlier removed without feature selection 57
4.4 ROC curves demonstrating model performance on the PIMA indian dataset for pre-split data resampling Elimination and Outlier removed without feature selection. 58
4.5 Data Distribution for Imputation & Outlier removed with feature selection 58
4.6 ROC curves demonstrating model performance on the PIMA indian dataset for pre-split data resampling Imputation and Outlier removed with feature selection. 59
4.7 Data Distribution for Elimination & Outlier removed with feature selection 60
4.8 ROC curves demonstrating model performance on the PIMA indian dataset for pre-split data resampling Elimination and Outlier removed with feature selection. 61
4.9 Data Distribution for Imputation With outlier without feature selection . . 61
4.10 ROC curves demonstrating model performance on the PIMA indian dataset for pre-split data resampling Imputation With outlier without feature selection. 62
4.11 Data Distribution for Elimination With outlier without feature selection . 63
4.12 ROC curves demonstrating model performance on the PIMA indian dataset for pre-split data resampling Elimination With outlier without feature selection. 64
4.13 Data Distribution for Imputation With outlier with feature selection 64
4.14 ROC curves demonstrating model performance on the PIMA indian dataset for pre-split data resampling Imputation With outlier with feature selection. 65
4.15 Data Distribution for Elimination With outlier with feature selection . . . 66
4.16 ROC curves demonstrating model performance on the PIMA indian dataset for pre-split data resampling Elimination With outlier with feature selection. 67

4.17 Data Distribution for Outlier removed without feature selection 68
4.18 ROC curves demonstrating model performance on the BRFSS dataset for pre-split data resampling Outlier removed without feature selection. 69
4.19 Data Distribution for Outlier removed with feature selection 70
4.20 ROC curves demonstrating model performance on the BRFSS dataset for pre-split data resampling Outlier removed with feature selection. 71
4.21 Data Distribution for With outlier without feature selection 72
4.22 ROC curves demonstrating model performance on the BRFSS dataset for pre-split data resampling With outlier without feature selection. 73
4.23 Data Distribution for With outlier with feature selection 74
4.24 ROC curves demonstrating model performance on the BRFSS dataset for pre-split data resampling With outlier with feature selection. 75
4.25 Data Distribution for Outlier removed without feature selection 76
4.26 ROC curves demonstrating model performance on the DiaHealth dataset for pre-split data resampling Outlier removed without feature selection. . . 77
4.27 Data Distribution for Outlier removed with feature selection 78
4.28 ROC curves demonstrating model performance on the DiaHealth dataset for pre-split data resampling Outlier removed with feature selection. 79
4.29 Data Distribution for With outlier without feature selection 80
4.30 ROC curves demonstrating model performance on the DiaHealth dataset for pre-split data resampling With outlier without feature selection. 81
4.31 Data Distribution for With outlier with feature selection 82
4.32 ROC curves demonstrating model performance on the DiaHealth dataset for pre-split data resampling With outlier with feature selection. 83
4.33 Data Distribution for Imputation & Outlier removed without feature selection 84
4.34 ROC curves demonstrating model performance on the PIMA indian dataset for post-split data resampling Imputation and Outlier removed without feature selection. 86
4.35 Data Distribution for Elimination & Outlier removed without feature selection 87
4.36 ROC curves demonstrating model performance on the PIMA indian dataset for post-split data resampling Elimination and Outlier removed without feature selection. 88
4.37 Data Distribution for Imputation & Outlier removed with feature selection 89
4.38 ROC curves demonstrating model performance on the PIMA indian dataset for post-split data resampling Imputation and Outlier removed with feature selection. 90
4.39 Data Distribution for Elimination & Outlier removed with feature selection 91
4.40 ROC curves demonstrating model performance on the PIMA indian dataset for post-split data resampling Elimination and Outlier removed with feature selection. 92
4.41 Data Distribution for Imputation With outlier without feature selection . . 93
4.42 ROC curves demonstrating model performance on the PIMA indian dataset for post-split data resampling Imputation With outlier without feature selection. 94
4.43 Data Distribution for Elimination With outlier without feature selection . 95
4.44 ROC curves demonstrating model performance on the PIMA indian dataset for post-split data resampling Elimination With outlier without feature selection. 96
4.45 Data Distribution for Imputation With outlier with feature selection 97

4.46 ROC curves demonstrating model performance on the PIMA indian dataset for post-split data resampling Imputation With outlier with feature selection. 98
4.47 Data Distribution for Elimination With outlier with feature selection . . . 99
4.48 ROC curves demonstrating model performance on the PIMA indian dataset for post-split data resampling Elimination With outlier with feature selection. 100
4.49 Data Distribution for Outlier removed without feature selection 101
4.50 ROC curves demonstrating model performance on the BRFSS dataset for post-split data resampling Outlier removed without feature selection. . . . 103
4.51 Data Distribution for Outlier removed with feature selection 104
4.52 ROC curves demonstrating model performance on the BRFSS dataset for post-split data resampling Outlier removed with feature selection. 105
4.53 Data Distribution for With outlier without feature selection 106
4.54 ROC curves demonstrating model performance on the BRFSS dataset for post-split data resampling With outlier without feature selection. 107
4.55 Data Distribution for With outlier with feature selection 108
4.56 ROC curves demonstrating model performance on the BRFSS dataset for post-split data resampling With outlier with feature selection. 109
4.57 Data Distribution for Outlier removed without feature selection 110
4.58 ROC curves demonstrating model performance on the DiaHealth dataset for post-split data resampling Outlier removed without feature selection. . 112
4.59 Data Distribution for Outlier removed with feature selection 113
4.60 ROC curves demonstrating model performance on the DiaHealth dataset for post-split data resampling Outlier removed with feature selection. . . . 114
4.61 Data Distribution for With outlier without feature selection 115
4.62 ROC curves demonstrating model performance on the DiaHealth dataset for post-split data resampling With outlier without feature selection. . . . 116
4.63 Data Distribution for With outlier with feature selection 117
4.64 ROC curves demonstrating model performance on the DiaHealth dataset for post-split data resampling With outlier with feature selection. 118
4.65 Histograms demonstrating the feature distributions of the PIMA indian dataset (Part 1): Raw distribution and after outlier removal. 124
4.66 Histograms demonstrating the feature distributions of the PIMA indian dataset (Part 2): Pre-imputation data after applying SMOTEENN for training and testing. 125
4.67 Histograms demonstrating the feature distributions of the PIMA indian dataset (Part 3): Post-imputation modeling with SMOTEENN for training and testing data. 126
4.68 Histograms demonstrating the feature distributions of the DiaHealth dataset (Part 1): Raw distribution and after outlier removal. 128
4.69 Histograms demonstrating the feature distributions of the DiaHealth dataset (Part 2): Pre-split data after applying SMOTEENN for training and testing. 129
4.70 Histograms demonstrating the feature distributions of the DiaHealth dataset (Part 3): Post-split modeling with SMOTEENN for training data and preserved testing data. 130
4.71 Representative SHAP summary plots illustrating feature importance across the entire population of the PIMA indian dataset: (a) pre-split beeswarm plot, (b) pre-split bar plot, (c) post-split beeswarm plot, and (d) post-split bar plot. 132

4.72 Representative SHAP summary plots illustrating feature importance specifically for the diabetic minority class in the PIMA indian dataset: (a) pre-split beeswarm plot, (b) pre-split bar plot, (c) post-split beeswarm plot, and (d) post-split bar plot. 133
4.73 Representative SHAP summary plots illustrating feature importance across the entire population of the Diahealth dataset: (a) pre-split beeswarm plot, (b) pre-split bar plot, (c) post-split beeswarm plot, and (d) post-split bar plot. 135
4.74 Representative SHAP summary plots illustrating feature importance specifically for the diabetic minority class in the Diahealth dataset: (a) pre-split beeswarm plot, (b) pre-split bar plot, (c) post-split beeswarm plot, and (d) post-split bar plot. 137
4.75 Representative SHAP summary plots illustrating feature importance across the entire population of the BRFSS-2015 dataset: (a) pre-split beeswarm plot, (b) pre-split bar plot, (c) post-split beeswarm plot, and (d) post-split bar plot. 139
4.76 Representative SHAP summary plots illustrating feature importance specifically for the diabetic minority class in the BRFSS-2015 dataset: (a) pre-split beeswarm plot, (b) pre-split bar plot, (c) post-split beeswarm plot, and (d) post-split bar plot. 141

List of Tables

2.1 Summary of recent machine learning studies for diabetes prediction and their reported performance metrics 25

3.1 Summary of PIMA Indian Dataset Attributes and Class Distribution . . . 32
3.2 PIMA Indian Dataset Feature Names, Descriptions, and Measurement Units 33
3.3 Attribute-wise Statistical Summary of PIMA indian dataset Features (N = 768). (Without imputation) 33
3.4 Summary of BRFSS 2015 Dataset Attributes and Class Distribution 34
3.5 BRFSS 2015 Feature Names, Descriptions, Types, and Value Ranges . . . 35
3.6 Attribute-wise Statistical Summary of BRFSS 2015 Features (after prediabetic exclusion, N = 249,049) 36
3.7 Summary of DiaHealth Dataset Attributes and Class Distribution 37
3.8 DiaHealth Feature Names, Descriptions, Types, and Values/Units 38
3.9 Attribute-wise Statistical Summary of DiaHealth Continuous Features (N = 5,437) 39
3.10 Missing Values in PIMA Indian Diabetes Dataset 40
3.11 Summary of experimental components and their configurations 54

4.1 Performance comparison of models after imputation, outlier removal, and without feature selection in the PIMA indian dataset 56
4.2 Performance comparison of models after elimination, outlier removal, and without feature selection in the PIMA indian dataset 57
4.3 Performance comparison of models after imputation, outlier removal, and with feature selection in the PIMA indian dataset 59
4.4 Performance comparison of models after elimination, outlier removal, and with feature selection in the PIMA indian dataset 60
4.5 Performance comparison of models after imputation, with outliers, and without feature selection in the PIMA indian dataset 62
4.6 Performance comparison of models after elimination, with outliers, and without feature selection in the PIMA indian dataset 63
4.7 Performance comparison of models after imputation, with outliers, and with feature selection in the PIMA indian dataset 65
4.8 Performance comparison of models after elimination, with outliers, and with feature selection in the PIMA indian dataset 66
4.9 Performance comparison of models after outlier removal and without feature selection in the BRFSS dataset 68
4.10 Performance comparison of models after outlier removal and with feature selection in the BRFSS dataset 70
4.11 Performance comparison of models with outliers and without feature selection in the BRFSS dataset 72

4.12 Performance comparison of models with outliers and with feature selection in the BRFSS dataset . . . 74
4.13 Performance comparison of models after outlier removal and without feature selection in the DiaHealth dataset . . . 76
4.14 Performance comparison of models after outlier removal and with feature selection in the DiaHealth dataset . . . 78
4.15 Performance comparison of models with outliers and without feature selection in the DiaHealth dataset . . . 80
4.16 Performance comparison of models with outliers and with feature selection in the DiaHealth dataset . . . 82
4.17 Performance comparison of models with imputation, outliers removed, and without feature selection in the PIMA indian dataset . . . 85
4.18 Performance comparison of models with elimination, outliers removed, and without feature selection in the PIMA indian dataset . . . 87
4.19 Performance comparison of models with imputation, outliers removed, and with feature selection in the PIMA indian dataset . . . 89
4.20 Performance comparison of models with elimination, outliers removed, and with feature selection in the PIMA indian dataset . . . 91
4.21 Performance comparison of models with imputation, with outliers, and without feature selection in the PIMA indian dataset . . . 93
4.22 Performance comparison of models with elimination, with outliers, and without feature selection in the PIMA indian dataset . . . 95
4.23 Performance comparison of models with imputation, with outliers, and with feature selection in the PIMA indian dataset . . . 97
4.24 Performance comparison of models with elimination, with outliers, and with feature selection in the PIMA indian dataset . . . 99
4.25 Performance comparison of models after outlier removal and without feature selection in the BRFSS dataset . . . 102
4.26 Performance comparison of models after outlier removal and with feature selection in the BRFSS dataset . . . 104
4.27 Performance comparison of models with outliers and without feature selection in the BRFSS dataset . . . 106
4.28 Performance comparison of models with outliers and with feature selection in the BRFSS dataset . . . 108
4.29 Performance comparison of models with outliers removed and without feature selection in the Diahealth dataset . . . 111
4.30 Performance comparison of models with outliers removed and with feature selection in the Diahealth dataset . . . 113
4.31 Performance comparison of models with outliers and without feature selection in the Diahealth dataset . . . 115
4.32 Performance comparison of models with outliers and with feature selection in the Diahealth dataset . . . 117
4.33 Pima Indian Diabetes Dataset comparison: Pre-split vs. Post-split . . . 120
4.34 BRFSS-2015 comparison: Pre-split vs. Post-split . . . 121
4.35 Diahealth comparison: Pre-split vs. Post-split . . . 121
4.36 Dominant feature comparison across the eight matched pre/post pairs (PIDD) 132
4.37 Dominant feature comparison across the four matched pre/post pairs (Diahealth) . . . 136
4.38 Dominant feature comparison across the four matched pre/post pairs (BRFSS-2015) . . . 140

1 Introduction

1.1 Background and Motivation

1.1.1 Diabetes Mellitus and Its Global Impact

Diabetes Mellitus (DM) is a group of metabolic disorders, caused by prolonged hyperglycaemia due to insufficient production of insulin, namely Type 1 Diabetes, or as a result of insulin resistance combined with impaired insulin secretion, known as Type 2 Diabetes, globally [1]. This enduring increase of glucose level in blood can lead to both microvascular, where diabetic retinopathy, nephropathy, and neuropathy are caused due to damaged small blood vessels, and macrovascular complications, where large blood vessels are affected resulting in coronary artery disease and stroke, alongside other disorders like depression contributing to morbidity and reducing life quality [1][2][3]. It is among the most rapidly spreading major public health concerns throughout the world, with the International Diabetes Federation estimating 536.6 million adults with diabetes in 2021, expected to increase over 783.2 million by 2045 [4]. Thus, early diagnosis can reduce its clinical severity and contributing risk factors, enhancing the importance of predictive systems in healthcare [3][5].

Recent studies demonstrated the importance of diabetes prediction because of its rising frequency and treatment complexities in advanced stages [3][6]. For large-scale early detection, conventional approaches utilizing clinical tests and physician evaluation for diagnosis are often costly and time-consuming [5][7].

1.1.2 Advancements and Trends in Machine Learning for Diabetes Prediction

The patterns associated with diabetes can be effectively predicted using the developed Machine Learning (ML) paradigms which utilizes the clinical and demographic data of the subjects [8][9][10]. Early risk detections using ML models are enabling timely preventive measures adoption now, due to their capability of analyzing larger datasets and optimizing the predictive performance in comparison to the traditional diagnostic methods [11][12].

Diabetes has been predicted by implementing various ML algorithms including Naive Bayes, Random Forest (RF), Support Vector Machine (SVM), and boosting-based methods such as eXtreme Gradient Boosting (XGBoost) and Light Gradient Boosting Machine (LightGBM) so far [13][14][15][16]. To date, the potential of ensemble approaches, especially the boosting frameworks, have been widely investigated for their better classification performance and complex relationships representation ability in datasets [16][17][18][19].

XGBoost and Logistic Regression (LR) was recorded as the highest performing predictive models, on PIMA Indian Diabetes Dataset , with 0.8203 accuracy and 0.8824 precision for both through a comparative research evaluating multiple ML models [20].

Likewise, the benefits of integrating multiple models are demonstrated through ensem-

ble paradigm studies, where Decision Tree, RF, XGBoost, and LightGBM are combined to achieve an accuracy of 0.735 and 0.832 AUC [21][22][23].

Yet, the predictive performance of different ML models varies for a range of tasks. For instance, diabetes progression has been evaluated by applying Gaussian Naive Bayes, SVM, and RF where a competitive performance has been demonstrated by RF in experimental studies [9][24].

In recent studies, boosting models like XGBoost and LightGBM have gained considerable attention for diabetes prediction. A study observed that to achieve better accuracy and computational performance appropriate feature selection techniques need to be employed with LightGBM [13][15]. Another study illustrated a strong predictive capability of the ensemble models, integrating XGBoost and KNN, through a high accuracy of 0.9482 and a strong 0.989 ROC-AUC performance utilizing the Behavioral Risk Factor Surveillance System-2015 (BRFSS-2015) dataset [25].

Furthermore, in certain diabetes prediction tasks comparative analysis studies have shown that LightGBM can outperform traditional ML models, while XGBoost consistently provides competitive performance [15].

1.1.3 Role of Different Machine Learning Models

In diabetes prediction, ML models play a vital role of learning and establishing multifaceted relationships between clinical and demographic variables. Recent studies consistently show that different algorithms make unique contributions to interpretability, robustness, and predictive performance in several clinical classification tasks.

Due to the simplicity and computational efficiency of Naive Bayes, it is frequently used as a baseline probabilistic classifier in studies. Research on the PIMA Indian Diabetes dataset illustrates that although Naive Bayes is competitive in early-stage prediction tasks performance, when there are strong correlations among clinical variables like glucose, Body Mass Index (BMI), and insulin levels, its performance may be compromised due to its assumption of feature independence [26][27].

Tree-based models such as Decision Trees and RFs are widely employed in medical data because of their ability to identify non-linear relationships, with RF showing better stability and generalization than single decision trees. Comparative studies in diabetes prediction report that RF consistently outperforms simpler models by achieving strong accuracy and recall under identical experimental conditions [9][24].

Some of the recent studies reveals that more advanced ensemble techniques, including boosting-based algorithms such as XGBoost and LightGBM, exhibit superior performance by optimizing errors iteratively from previous learners and are highly effective in handling imbalanced datasets and complex feature interactions. For instance, a strong classification ability, with a high Area Under the Receiver Operating Characteristic Curve (AUC-ROC) and accuracy value, was exhibited by the XGBoost-based frameworks while distinguishing between diabetic and non-diabetic cases [10][13][15][28].

Recent studies reveal that hybrid and ensemble learning approaches, combining multiple classifiers like XGBoost, RF, and others are integrated through stacked and weighted ensembles to achieve a higher ROC-AUC and F1-scores, demonstrate strong robustness in predictions for clinical decision support systems [25]. Additionally, the state-of-the-art performance was demonstrated through modern hybrid architectures, where the shift toward ensemble-driven medical prediction systems by combining multiple learning paradigms is shown [29].

Overall, recent literature shows that traditional classifiers are continuously outperformed by ensemble and boosting-based models in both clinical reliability and predictive

accuracy, justifying the research inclination towards these models. However, model interpretability and bias control remain essential considerations in healthcare applications for practical deployment.

1.1.4 Challenges in Diabetes Prediction Using Machine Learning

Despite the effective ML approaches, diabetes prediction still encounters several major challenges, like imbalanced datasets, where the number of non-diabetic cases is more than diabetic instances, affecting model predictive performance and reliability [30][31][32].

Furthermore, healthcare datasets often require preprocessing, such as imputation and feature replacement, to improve model accuracy by handling missing values and removing outliers [33][34] as such data quality problems can impact the ML models effectiveness notably.

Another challenge emphasized in recent literature is the difficulty in balancing precision and recall. Some studies indicate that models with high precision may still have relatively low recall, showing that a portion of diabetic cases may not be correctly identified, known as the false negative cases [30][31]. These cases illustrate the limitation of ML models in clinical applications where timely and accurate identification of positive cases is critical.

1.1.5 Motivation of This Study

Existing studies show that ML models, such as XGBoost, LightGBM, and Naive Bayes, have been widely applied for diabetes prediction to achieving promising results. Despite having the potential to support faster, more cost-effective diagnosis and assist clinicians through cross-checking mechanisms, ultimately facilitating early detection rather than replacing medical expertise, these models are not yet consistently integrated into hospital systems or routine clinical practice. A key contributing factor for this is that model performance varies depending on the dataset and experimental conditions [10][15][24]. Factors such as class imbalance, missing data, and feature selection have a major impact on predictive performance [15][30][34].

Moreover, despite appropriate preprocessing techniques and model optimization, challenges like inflated performance due to data leakage, inconsistent evaluation metrics, and limited interpretability remain, indicating limitations in current approaches [10][24]. Indicating that both dataset attributes and model behavior play crucial roles in determining prediction performance. Therefore, this study aims to systematically evaluate the performance of multiple ML models-specifically XGBoost, LightGBM, and Naive Bayes-across different diabetes datasets, including BRFSS-2015, DiaHealth and PIMA Indian Diabetes, under two distinct preprocessing pipelines (pre-split and post-split). This research focuses on three key aspects: quantifying the effect of pre-split resampling on performance metrics, assessing which evaluation measures (accuracy, precision, recall, AUC-ROC, F1-score, and Matthews Correlation Coefficient (MCC)) best reflect clinical utility in imbalanced datasets as primary measures, and identifying the features most crucial for minority class (Diabetes Positive) detection through SHapley Additive exPlanations (SHAP) based model interpretability.

1.2 Problem Statement

1.2.1 Issues in Data Quality and Model Performance in Diabetes Prediction Using ML

Despite the widespread implementation of ML models for diabetes prediction, prior research has exhibited that following standard methodological procedures, such as dividing datasets into training and testing sets before resampling the class imbalance, often results in inconsistent and suboptimal evaluation metrics [32][35]. A comparative study on the PIMA Indian Diabetes dataset found that XGBoost achieved the highest accuracy of 0.7467 while Naive Bayes without feature selection achieved 0.7077 accuracy and 0.6481 recall among LR, SVM, RF, Naive Bayes, and XGBoost [36].

Class imbalance is a common issue in clinical datasets, where the number of non-diabetic cases is much higher than diabetic ones. This imbalance often leads to biased predictions which weakens the ML models ability to generalize properly [37][38]. Various resampling techniques, such as Synthetic Minority Over-sampling Technique (SMOTE), SMOTE + Edited Nearest Neighbor (SMOTEENN), SMOTETomek, and Adaptive Synthetic Sampling (ADASYN), have been shown to improve evaluation metrics like AUC-ROC and F1-score. However, these methods still need to be applied carefully, especially before the evaluation stage, to properly handle the imbalance. In addition to this, the overall quality of clinical data also creates challenges. Problems like noisy data, missing values, and uneven data distributions can make it harder for ML models to learn accurate decision boundaries. As a result, these issues can reduce the overall robustness and reliability of the models [30][38].

However, existing research shows that traditional ML models, such as RF and gradient boosting methods like XGBoost and LightGBM, can achieve fairly high accuracy. Even then, they often struggle to keep a good balance across different evaluation metrics, particularly recall and F1-score, which are especially important in clinical settings [15][30][31]. For example, models like RF and XGBoost have shown strong recall and F1-score when resampling techniques such as SMOTE are used. However, additional preprocessing steps are still necessary to properly handle class imbalance and maintain the models ability to generalize well, especially when working with datasets like PIMA.

A key gap highlighted in recent studies is that many works focus on evaluating diabetes prediction using only a single dataset, without systematically examining how different dataset characteristics, such as the number of features, sample size, presence of missing values, or categorical variables, can influence the results when using advanced models like XGBoost, LightGBM, and Naive Bayes [15][30].Although some studies have reported changes in evaluation metrics like AUC-ROC and F1-score after applying resampling techniques, there has been limited effort to understand how and why these variations occur across datasets with different levels of quality and class imbalance. Moreover, while explainable ML approaches such as SHAP have been used to interpret feature importance, comparative analyses across multiple datasets are still relatively rare.

The DiaHealth dataset consists of 5,437 instances and 14 features, which covers clinical information, sociodemographic details, and medical history [39]. This makes it important to evaluate model performance on data that is broader and closer to real clinical scenarios. From the literature reviewed, it is clear that most studies do not systematically measure how performance inflation caused by factors like data leakage, choice of evaluation metrics, or feature contribution to detecting the minority (diabetic) class varies across different benchmark and real-world datasets when using an unbiased pipeline.

For this reason, the present study focuses on evaluating ML models, specifically XG-

Boost, LightGBM, and Naive Bayes, across multiple diabetes-related datasets, including PIMA Indian Diabetes, BRFSS-2015, and DiaHealth. The goal is to determine whether issues related to dataset quality or the characteristics of the models themselves are the main reasons behind lower predictive performance, even when standard pipelines are applied.

More specifically, this study aims to:

- Measure the impact of applying resampling before the train-test split on evaluation results.
- Examine which performance metrics (such as accuracy, precision, recall, F1-score, MCC or AUC-ROC) better reflect clinical usefulness in imbalanced diabetes datasets.
- Identify the most important features for detecting the minority (diabetes-positive) class using SHAP-based interpretability methods.

The results of this study will help clarify whether the limitations in model performance are mainly due to issues in the data such as class imbalance, poor feature quality, or missing values or if they arise from the inherent characteristics of the models themselves. This understanding can guide the development of more reliable and robust ML approaches for clinical diabetes prediction [10][24][31][38].

1.2.2 Research Gap

A research gap is an area of knowledge or aspect of a topic that is not thoroughly explored in previous studies yet, it opens up possibilities for fresh and further investigations that can progress the area, complete knowledge gaps, and tackle open-ended problems [40]. To identify the research gap, a comprehensive review was conducted of existing research papers, and examining the current studies allowed us to pinpoint the unaddressed aspects and led us to formulate the research focus of our study.

In the field of ML-based diabetes prediction, several research gaps have been identified in the literature. A systematic review of ML applications in diabetes prediction reported limitations including data collection and preprocessing, handling missing values, feature importance assessment, standardization, and addressing class imbalance issues as key challenges that are not yet fully resolved in existing ML studies [24]. Another recent review of ML and artificial intelligence (AI) applications for diabetes prediction noted that although many ML models have been developed, there remains limited external validation, a lack of standardized preprocessing frameworks, and insufficient explainability in model outputs to support clinical adoption and practical applicability [24].

In this context, the major research gaps that were found are:

- Impact of data leakage and resampling strategies: Insufficient understanding of how improper resampling before train-test splitting inflates evaluation metrics in diabetes datasets, affecting the reliability of reported model performance [41][42].
- Evaluation metric relevance for clinical applicability: Conflicting perspectives on which evaluation metrics (accuracy, precision, recall, AUC-ROC, F1-score, or MCC) best reflect clinical utility in imbalanced diabetes datasets, and limited comparative studies examining how metric choice influences model interpretation and prediction outcomes [43][44][45].
- Feature contribution to minority class detection: Few studies have systematically investigated which features most significantly influence correct identification of diabetes-positive cases, especially under unbiased pipelines, and how SHAP based explanations can provide insights into model decision-making [38][46][47].

- Standardized preprocessing and validation frameworks: There is a need for structured methods for preprocessing, resampling, and validating ML models that can reliably generalize across diverse populations and datasets [24][30][31]. These gaps highlight the necessity of a structured investigation into the combined effects of dataset quality, evaluation methodology, and model choice on predictive performance guiding us towards the core focus of this study.

1.3 Research Questions (RQ)

- **RQ1:** How significantly does data leakage from pre split resampling inflate model performance metrics in existing diabetes detection studies?
- **RQ2:** Which evaluation metrics most accurately reflect clinical utility in an imbalanced Diabetes Dataset and how do results differ when accuracy is replaced with precision, recall (sensitivity), AUC-ROC, F1-score and MCC as primary measures?
- **RQ3:** Which feature most significantly contributes to minority class (Diabetes Positive) detection under a correct unbiased pipeline and what do SHAP based explanations reveal about model decision making in diabetes prediction?

1.4 Research Objectives

The primary objectives of this research are:

- To evaluate the impact of data leakage from pre-split resampling and post-split resampling, using SMOTE, SMOTEENN, SMOTETomek and ADASYN, on diabetes classification models predictive performance. This involves systematically assessing XGBoost, LightGBM, and Naive Bayes on BRFSS-2015, DiaHealth and PIMA Indian Diabetes datasets to quantify how improper preprocessing, particularly pre-split resampling, inflates model evaluation metrics such as accuracy, F1-score, and recall (sensitivity).
- To identify evaluation metrics that most accurately reflect clinical utility in imbalanced datasets. By comparing accuracy, precision, recall, F1-score, MCC and AUC-ROC to analyze the changes in model performance interpretation, this study aims to determine which measures provide more reliable insights for real-world diabetes detection for minority class (diabetes positive) cases when accuracy is replaced.
- To investigate the contribution of individual features to minority class prediction and interpret model decisions using SHAP-based explainability methods. This objective investigates which features most strongly influence correct diabetes-positive predictions and how model decision-making varies across XGBoost, LightGBM, and Naive Bayes under unbiased pipelines.

1.5 Contribution of the Study

- This study began by reviewing scientific papers to improve the reliability and clinical relevance of ML models analyzing publicly available datasets like BRFSS-2015, DiaHealth and PIMA Indian Diabetes for diabetes prediction, and identifying limitations such as data leakage, improper evaluation metrics, and imbalanced datasets

in previous research. The review identifies crucial limitations like pursuing unstructured experimental pipeline including presplit resampling, feature preprocessing, and model tuning, to achieve higher performance results affecting its reliability and clinical relevance.

- To address these issues, we implemented and evaluated multiple ML models via standardized and leakage-free experimental pipeline, specifically XGBoost, LightGBM, and Naive Bayes, across various datasets to determine the impact of data characteristics on model performance. Comparison of the evaluation results revealed the extent to which presplit resampling artificially inflates performance metrics, and demonstrated which evaluation measures among AUC-ROC, F1-score, MCC and Recall (Sensitivity) more accurately reflect clinical utility in imbalanced diabetes datasets.

- Furthermore, SHAP-based feature importance analysis was conducted to clarify which features contribute the most to detect minority class instances (Diabetes Positive), providing interpretable insights into model decision-making. These contributions collectively improve the understanding of how diabetes prediction frameworks are impacted by data preprocessing, evaluation strategy, and model interpretability, guiding the development of more robust, unbiased, and clinically useful ML models.

1.6 Report Outline

Chapter 1 (Introduction): The introduction chapter serves a thorough description and overview of diabetes mellitus and its worldwide impact, alongside the increasing significance of ML in early disease prediction. It discusses the motive behind using ML techniques in healthcare, namely for diabetes prediction, and addresses the crucial challenges like class imbalance, problems with data quality, and limitations in evaluation metrics. Additionally, the research gaps are identified- from which the research questions (**RQ1**, **RQ2**, and **RQ3**) are formulated besides highlighting the objectives and key contributions of this study. In general, it provides the foundation and context of this thesis along with the entire thesis outline.

Chapter 2 (Literature Review): A comprehensive review of existing ML based diabetes prediction studies is analyzed in this chapter where it explores diabetic clinical factors, commonly used datasets, and various ML models utilized in prior research. It also evaluated various resampling techniques for handling imbalanced datasets, clinically relevant evaluation metrics, and explainable AI methods like SHAP to analyze the decision-making process. Chapter 2 concludes with highlighting the limitations of previous research besides reinforcing the research gaps addressed (**RQ1**,**RQ2**, and **RQ3**) in this study.

Chapter 3 (Materials and Methodology): The overall experimental design and framework of this study is outlined in this chapter besides providing detailed descriptions of the used datasets named as PIMA Indian Diabetes Dataset, BRFSS-2015, and DiaHealth. A detailed explanation of the preprocessing steps including handling missing value, outlier reduction, and feature selection is given here, along with the implementation discussion of resampling methods like SMOTE, SMOTEENN, SMOTETomek, and ADASYN. This chapter also described the ML models (Gaussian Naive Bayes. XGBoost, and LightGBM) used, evaluation metrics considered, and SHAp-based explainabil-

ity paradigm. Lastly, it mentions about the hardware, software, and execution protocols of the experimental setup.

Chapter 4 (Results and Analysis): The pre-split and post-split resampling pipelines results obtained across the three datasets are represented in this chapter. It answers **RQ2** via thorough performance comparison of the three models under each of the resampling approaches implemented using various evaluation metrics like accuracy, precision, recall, F1-score, AUC-ROC, and MCC. **RQ1** is addressed by the comparative analysis, to measure the effects of data leakage, between the pre-split and post-split results. It also discusses the SHAP-based feature importance analysis, conducted to understand the models behaviors beside identifying the key predictors for diabetes predictions in response to **RQ3**.

Chapter 5 (Discussion of Key Findings): This chapter interprets the major experimental outcomes and analyzes them in light of the three research questions. It investigates how pre-split resampling impacts performance inflation (**RQ1**), examines which evaluation metrics reflect the best clinical utility (**RQ2**), and describes the contribution of SHAP to reveal the most significant features during minority class detection (**RQ3**), such as in diabetes prediction when diabetes is the positive and minority class. The results are then compared with the existing literature to emphasize the importance and implications of this research.

Chapter 6 (Threats to Validity): The potential limitations which may affect the studys validity are reviewed in this chapter. The transparency and reliability of the findings related to the research questions are ensured through internal validity (such as randomness in data splitting and decisions for preprocessing), external validity (like generalizability spanning datasets and populations), construct validity (for instance.evaluation results and explainability), and conclusion validity (including lack of statistical significance testing and limited use of cross-validation) here.

Chapter 7 (Limitations): The practical limitations of the research like dataset constraints, limited generalizability, and challenges linked to SHAP-based interpretability are outlined in this chapter, beside mentioning the restriction to employ in practical application due to lack of prospective clinical validation.

Chapter 8 (Ethical Considerations): The ethical consideration part addresses the studys ethical implications, including algorithmic bias and fairness, potential clinical consequences of misclassification, and scientific integrity in presenting results alongside highlighting how crucial responsible implementation of ML in healthcare is.

Chapter 9 (Conclusion and Future Work): The chapter underscores the key conclusions and contributions to summarize the entire thesis. Here, the improvements brought by the research in knowledge and understanding of data leakage through **RQ1**, evaluation metrics via **RQ2**, and the importance of features in diabetes prediction using **RQ3** are thoroughly explained. It also recommends future study directions such as generalizability improvement, use of additional datasets, and development of more robust and clinically reliable ML frameworks.

2 Literature Review

2.1 Diabetes and Clinical Predictors

Diabetes Mellitus is a chronic, multi-systemic metabolic disorder characterized by continuously high levels of blood glucose caused by problems in insulin secretion, insulin function, or both [48]. There are three main categories of diabetes, Type 1 diabetes, Type 2 diabetes and Gestational Diabetes Mellitus (GDM) [49]. Type-1 diabetes is an autoimmune condition where the body attacks the insulin-producing beta cells in the pancreas, resulting in an absolute deficiency of insulin [49]. Type-2 diabetes is predominantly driven by progressive insulin resistance and is heavily linked to lifestyle and metabolic factors [49]. GDM occurs specifically during the time of pregnancy in women without a previous history of diabetes [49]. Most of the time GDM is often cured after childbirth, it suggests the presence of a serious metabolic disorder and puts the mother at a much higher-than-average risk of developing diabetes type 2 later on in life [49]. While there are several types of classifications, Type 2 diabetes mellitus (T2DM) accounts for over 90% of all cases worldwide [48]. The condition now exists as an urgent worldwide healthcare crisis.

The chronic hyperglycemia associated with T2DM does not occur independently; rather, it initiates a series of systemic processes, including inflammation, oxidative stress, and endothelial dysfunction, which collectively damage both the microvascular and macrovascular systems across the body [50].

Diagnosing and predicting the onset of diabetes relies on evaluating a combination of physiological, genetic, and lifestyle factors. Before translating patient data into computational models, it is critical to understand the biological mechanisms that make certain clinical features strong predictors of the disease.

- **Blood Glucose Indicators:** Blood glucose concentration is the most important concern for diabetes. In clinical practice, fasting plasma glucose (FPG), oral glucose tolerance tests (OGTT), and Hemoglobin A1c (HbA1c) levels can indicate how well the body controls blood glucose over short and long periods. When these levels are higher than normal, it usually means the pancreas is not producing insulin properly, or the bodys cells are not responding to insulin effectively [51].

- **Body Mass Index (BMI) and Obesity:** Obesity is the main modifiable risk factor for T2DM The body fat that exceeds normal levels, especially the visceral fat distribution produces pro-inflammatory cytokines together with non-esterified fatty acids which both disrupt the body's insulin signaling mechanisms [52]. T2DM can be predicted through elevated BMI because it serves as a highly sensitive clinical indicator.

- **Blood Pressure (Hypertension):** Hypertension and diabetes are closely interlinked through various metabolic pathways that both conditions share according

to their common mechanisms which include endothelial dysfunction and oxidative stress [53]. The clinical observation shows that higher systolic and diastolic blood pressure levels directly connect to the development of insulin resistance which makes these blood pressure measurements vital for assessing patients [54].

- **Family History and Genetic Factors:** Genetic factors play an important role in the development of both Type 1 and Type 2 diabetes. A family history of diabetes suggests a higher inherited risk of problems with insulin production or insulin function [55].
- **Pregnancy and Gestational History:** Because of the biological strain pregnancy places on insulin signaling, a history of Gestational Diabetes Mellitus is a critical predictive feature in clinical datasets. Hormonal changes during pregnancy can induce temporary insulin resistance; if the pancreas cannot compensate, GDM occurs. Consequently, medical professionals view a previous GDM diagnosis as a strong early-warning indicator, as it radically increases the statistical probability of the patient developing chronic Type 2 diabetes in the future [56].
- **Lifestyle and Metabolic Factors:** Medical professionals also consider other important health indicators such as physical inactivity, eating habits, and lipid profiles (for example, cholesterol levels). The absence of consistent exercise leads to decreased muscle capacity for glucose processing, which ultimately results in deteriorating metabolic health [57].

2.2 Machine Learning in Diabetes Prediction: An Overview

Diabetes has become a global health crisis, because of this, early and accurate detection is crucial to prevent severe health complications [58][59].However, traditional clinical methods of screening, such as regular blood tests or advanced cardiovascular imaging, are too costly, time-consuming, and resource intensive to implement for a large population [59][60]. Researchers turned to ML in order to overcome these barriers, as it is a cost-effective alternative to standard diagnostic testing [58][59][60]. By maximizing the power of these algorithms to process complex clinical data very fast, healthcare professionals hope to achieve more efficient, and accurate predictions that can empower early intervention and ultimately transform patient care [59][60][61].

Modak and Jha [62] worked with the "Diabetic2" dataset, consisting of 5,000 instances and a total of 9 attributes. Models used in the study are LR, SVM, Naïve Bayes, RF, XGBoost, LightGBM, CatBoost, AdaBoost, and Bagging. CatBoost performed the best; it scored 0.9558 accuracy during 10-fold cross-validation, and on the baseline evaluation, it scored 0.954 accuracy, 0.926 precision, 0.909 recall, and a 0.917 F1 score [62].

Ullah et al. [63] worked with the BRFSS dataset consisting of 253,680 instances and in total 22 attributes. Models used in the study are K-Nearest Neighbor (KNN), RF, XGBoost, Bagging, and AdaBoost. KNN performed the best, scoring 0.9836 accuracy, 0.98 precision, 0.98 recall, and a 0.98 F1 score [63].

Rahman et al. [64] in their studies used the Pima Indian Diabetes Dataset (768 records) and a Bangladeshi dataset called DiaHealth (5,437 records) for diabetes prediction. In pre-processing stage, the researchers used a detailed cleaning process, which included fixing missing values, removing outliers, and balancing the data. They tested two tree-based models, LightGBM and XGBoost, and combined with different ML tools

to pick the most important health features. The LightGBM model using Boruta-selected features performed the best for both datasets; on the PIDD, it scored 0.8516 accuracy, 0.8404 precision, 0.8681 recall, and a 0.8541 F1 score , while on the DiaHealth dataset, it scored 0.9939 accuracy, recall 1, and a 0.9939 F1 score. This method significantly outperformed the TIPNet deep learning model and used SHAP analysis to ensure the results were easy for medical professionals to interpret [64].

Naz and Ahuja [65] worked with the PIMA Indian dataset, consisting of 768 instances and a total of 8 attributes. Models used in the study are Deep Learning (DL), Artificial Neural Network (ANN), Decision Tree (DT), and Naïve Bayes. Deep Learning performed the best; it scored 0.98 accuracy, 0.95 precision, 0.98 recall, 0.97 F1 score, and 0.99 specificity [65].

El-Sofany et al. [66] worked with the Pima Indians dataset consisting of 768 instances and a private dataset consisting of 300 instances, using a total of 9 attributes. Models used in the study are LR, RF, KNN, Decision Tree, Bagging, AdaBoost, XGBoost, Voting, SVM, and Naïve Bayes. XGBoost paired with the SMOTE technique performed the best on both datasets; on the private dataset, it scored 0.974 accuracy, 0.96 precision, 0.97 recall, and a 0.95 F1 score, while on the combined datasets, it scored 0.831 accuracy, 0.70 precision, 0.84 recall, and a 0.76 F1 score [66].

Chowdhury et al. [67] worked with the 2021 BRFSS dataset, consisting of 262,958 instances and a total of 21 attributes. The researchers used multiple data balancing techniques, which included oversampling through SMOTE-N and undersampling through Edited Nearest Neighbor (ENN), as well as hybrid methods that combined SMOTE-ENN with SMOTE-Tomek to address the extreme data imbalance problem. Models used in the study are LR, RF, Gradient Boosting, AdaBoost, and a Voting Classifier. Before balancing the data, the models achieved high overall accuracies ranging from 0.817 to 0.830, but suffered from low recall rates between 0.574 and 0.580, meaning they missed many positive diabetes cases. When used together with ENN's data balance technique, Gradient Boosting and Voting Classifier displayed peak performance. Gradient Boosting scored 0.703 accuracy, 0.635 precision, 0.717 recall, and a 0.791 AUC score, while the Voting Classifier scored 0.706 accuracy, 0.635 precision, 0.714 recall, and a 0.789 AUC score [67].

Ahmed et al. [68] worked with the PIMA Indian diabetes dataset, consisting of 768 instances and a total of 8 attributes. Models used in the study are RF (RF), Decision Tree (DT), Naïve Bayes, and LR. RF performed the best; it scored 0.80 accuracy, 0.82 precision, 0.88 recall, 0.84 F1 score and AUC-ROC of 0.83, alongside the lowest error rate (20%) [68].

Chowdhury [69] worked with the BRFSS 2015 survey dataset, containing demographic, behavioural, and lifestyle-related features. Models used in the study are Dummy Classifier, LR, and RF (RF). RF performed the best; it scored 0.83 accuracy, 0.74 precision, 0.74 recall, and 0.73 macro-averaged F1-score on the held-out test set [69].

Andrade-Arenas and Yactayo-Arias [70] worked with the Pima Indian dataset consisting of 768 instances and 9 attributes. Models used in the study are Decision Tree, Naïve Bayes, KNN, LR, RF, XGBoost, Artificial Neural Networks (ANN), SVM, and AdaBoost. RF and AdaBoost performed the best; during cross-validation, RF scored 0.816 accuracy, 0.804 precision, 0.866 recall, and a 0.828 F1 score, while on the test data, AdaBoost scored 0.766 accuracy, 0.645 precision, 0.741 recall, and a 0.69 F1 score [70].

Table 2.1: Summary of recent machine learning studies for diabetes prediction and their reported performance metrics

Study	Dataset	Models Used	Best Model	Best Reported Performance
[62]	Diabetic2	LR, SVM, NB, RF, XGBoost, LightGBM, CatBoost, AdaBoost, Bagging	CatBoost	Accuracy: 0.9558 (10-fold CV (Cross Validation) Precision: 0.926 Recall: 0.909 F1-score: 0.917
[63]	BRFSS-2015	KNN, RF, XGBoost, Bagging, AdaBoost	KNN	Accuracy: 0.9836 Precision: 0.980 Recall: 0.980 F1-score: 0.980
[64]	PIDD and DiaHealth	LightGBM, XGBoost with feature selection	LightGBM with Boruta features	PIDD: Accuracy: 0.8516 Precision: 0.8404 Recall: 0.8681 F1-score: 0.8541 DiaHealth: Accuracy: 0.9939 Recall: 1.000 F1-score: 0.9939
[65]	PIMA	DL, ANN, DT, NB	Deep Learning	Accuracy: 0.980 Precision: 0.950 Recall: 0.980 F1-score: 0.970 Specificity: 0.990
[66]	Pima and private dataset	LR, RF, KNN, DT, Bagging, AdaBoost, XGBoost, Voting, SVM, NB	XGBoost with SMOTE	Private dataset: Accuracy: 0.974 Precision: 0.960 Recall: 0.970 F1-score: 0.950 Combined dataset: Accuracy: 0.831 Precision: 0.700 Recall: 0.840 F1-score: 0.760
[67]	BRFSS-2021	LR, RF, Gradient Boosting, AdaBoost, Voting	Gradient Boosting and Voting with ENN balancing	Gradient Boosting: Accuracy: 0.703 Precision: 0.635 Recall: 0.717 AUC: 0.791 Voting: Accuracy: 0.706 Precision: 0.635 Recall: 0.714
[68]	PIMA	RF, DT, NB, LR	RF	Accuracy: 0.800 Precision: 0.820 Recall: 0.880 F1-score: 0.840
[69]	BRFSS-2015	Dummy, LR, RF	RF	Accuracy: 0.830 Precision: 0.740 Recall: 0.740 F1-score: 0.730
[70]	Pima Indian dataset	DT, NB, KNN, LR, RF, XGBoost, ANN, SVM, AdaBoost	RF (CV) and AdaBoost (test)	RF (CV): Accuracy: 0.816 Precision: 0.804 Recall: 0.866 F1-score: 0.828

2.3 Common Datasets Used

For diabetes prediction, there are only a few datasets which are available publicly. The most common and established dataset on diabetes prediction is the Pima Indian Diabetes Dataset. Although historical, it remains a primary resource in recent methodologies too.

For example, Ahmed et al. [68] and Modak and Jha [62] utilized this dataset in diabetes prediction using various ML algorithms. On the other hand, in order to expand the study on this dataset, researchers like Edeh et al. [61] have proposed classification algorithm-based hybrid prediction models to enhance accuracy, e.g. combining supervised ML algorithms with k-means clustering. Rahman et al. [64] have also used the Pima Indian dataset so that they could achieve a better prediction outcome by implementing feature selection algorithms beside boosting-based classifiers. Though the Pima Indian dataset is a historical dataset, this is not the only one which is used in diabetes prediction in recent years. Studies have also shifted toward broader epidemiological data. Chowdhury et al. [67] investigated ML algorithms and data augmentation techniques using the class-imbalanced BRFSS-2021 dataset to improve diagnostic accuracy . Similarly advancing modern applications, El-Sofany et al. [66] leveraged a combination of the Pima Indians dataset and a private hospital dataset so that they can propose a prediction technique which is integrated into a mobile application. There is another common dataset named BRFSS-2015 available publicly which is from the Centers for Disease Control and Prevention (CDC), publicly available data, but preprocessed and available in Kaggle. In a very recent study, Ullah et al. [63] used the BRFSS-2015 dataset to detect high-risk factors for diabetes and build an automated decision support system to assist medical practitioners in early diagnosis. The National Health and Nutrition Examination Survey (NHANES) datasets provide rich clinical features but the datasets are complex and require too many pre-processing steps. But Still several researcher like Qin et al. [71] have processed these data and used the data to contribute to diabetes care. Addressing the geographical gap in these standard datasets, DiaHealth was established as a crucial South Asian dataset published by Bangladeshi university in 2024 [72]. Rahman et al. [64] utilized this dataset in their study to contribute in diabetes studies. Beyond these, the Kaggle "Diabetic2" dataset appears in studies like Modak and Jha [62], though it is no longer accessible in the public repository, highlighting the volatility of unofficial data sources. Consequently, researchers also turn to private or hospital-specific datasets to secure reliable, population-specific data for their single-label prediction studies [66]. The transition from localized, historical datasets like PIDD to massive, national-scale surveys like BRFSS-2015 and regionally focused sets like DiaHealth shows how ML diabetes research requires data that represents different population groups.

2.4 Resampling Techniques in Imbalanced Learning

In healthcare ML, class imbalance is a major challenge during diabetes prediction because healthy people who do not have the disease outnumber actual diabetic patients [73]. The design of standard ML algorithms focuses on maximizing their total accuracy. Because of this, they naturally favor the larger group (the majority class) and learn to simply predict healthy most of the time to keep their overall error rate low [74][75]. While this makes the model look highly accurate on paper, it greatly lowers the model's sensitivity, meaning the algorithm will frequently miss the smaller group of actual diabetic patients [75][73]. In the medical field, the cost of a false negative (failing to diagnose a sick patient) is much worse than a false positive. Therefore, this built-in bias makes the model practically useless in real clinical settings [75][76]. To fix this issue, researchers use resampling techniques to balance the training data. This removes the model's initial bias toward the majority class and helps it become sensitive enough to actually detect the disease [74][76][73].

Researchers created multiple data-level solutions that address the problem of unequal class distribution because they need to achieve mathematical balance between their train-

ing space dimensions. The foundation technique is the SMOTE, introduced by Chawla et al. [74], which synthesizes new minority class instances by interpolating between existing neighboring data points in the feature space.

He et al. [77] proposed the Adaptive Synthetic (ADASYN) sampling technique, which uses a weighted density distribution to dynamically shift the generation of synthetic data toward minority instances that are harder for the algorithm to learn.

Pure over-sampling methods successfully boost minority representation, but they create problems because they extend minority clusters too far into majority territory, which produces overlapping noise that disrupts class boundaries and causes overfitting problems [74][75]. To mitigate this, hybrid techniques such as SMOTE + ENN (combining SMOTE with Wilson's Edited Nearest Neighbor Rule) and SMOTE + Tomek (combining SMOTE with Tomek links) were introduced [75]. These methods first over-sample the minority class and then use data cleaning procedures to remove noisy or borderline samples from both classes, which helps to create more distinct decision boundaries that the classifier can use [75].

Resampling techniques are effective for addressing class imbalance; their validity strictly depends on their placement within the ML pipeline. A fundamental requirement of predictive modeling is that the test set must remain completely independent to ensure a valid evaluation. Consequently, resampling must only be applied after the dataset has been partitioned into training and testing sets (post-split resampling). Using techniques like SMOTE on the whole dataset before splitting it into training and testing sets is a common mistake that can result in biased models and overly optimistic error estimates [76].

The mechanism of data leakage occurs because generating synthetic data prior to the train-test split (pre-split resampling) allows information from the test set to bleed into the training set. As the literature explains, using oversampling before splitting the data into training and test sets leads to an imperfect separation between the training and test sets because data generated using oversampling from the training set will also be present in the test set [78]. When the entire dataset is oversampled first, copies of the same patterns may appear in both the training and test sets [76]. This flawed implementation ultimately results in data leakage and inflated performance metrics because the model is essentially evaluated on variations of the exact data it has already seen during the training phase [73]. To mathematically guarantee the integrity of the evaluation, a strict post-split protocol must be followed. Research confirms that to prevent data leakage and ensure valid performance evaluation, a strict rule must be implemented where the dataset is split into training (80%) and test (20%) sets before any SMOTE application, ensuring that SMOTE was applied only to the training set [73]. By applying oversampling only to the training partitions, the patterns included in the test set are never oversampled or seen by the model in the training stage, thus allowing a proper evaluation of the models capability to generalize from the training data [76].

2.5 Evaluation Metrics in imbalanced Medical Datasets

When it comes about prediction, we think about accuracy. Based on accuracy, we can understand how accurate the prediction is. The question is, Is accuracy enough to justify how well a model is performing?

The research by He et al. [79] shows that accuracy does not provide enough information to evaluate a model's complete performance when testing medical datasets. A dataset exists which contains 95% healthy patients and only 5% patients with a particular disease.

A model which is designed poorly can predict everyone is healthy and still the model can achieve that 0.95 accuracy score. The number looks quite impressive in the paper, but the model is not even able to predict one single diseased patient. In the medical field this is a very concerning issue and this can lead to a very dangerous consequence. The accuracy metric can easily hide these kinds of critical, real-world failures. It does not provide enough information, which means we must rely on other metrics to ensure the model is giving attention to the rare cases [79].

As, simple accuracy alone is not enough to ensure a model's performance, it is necessary to use metrics that are designed to capture a model's true real-world performance. The metrics like Precision, Recall, F-Measure, and G-mean assess model performance through the evaluation of correct predictions beyond their basic accuracy. Ultimately, these metrics evaluate the real world scenario. which shows how many true positive predictions were made versus the number of false case predictions [79].

Besides, the Matthews Correlation Coefficient (MCC) came up with a more reliable metric for imbalanced data. While many recent studies rely heavily on combinations of Precision, Recall, and the F-measure to evaluate their models [60][66][63], these standard metrics have a blind spot which is, the F-measure completely ignores true negative results. MCC solves this blind spot by calculating a single correlation score that mathematically integrates all four categories of the confusion matrix: true positives, true negatives, false positives, and false negatives. Because a high MCC score is only mathematically possible if the model successfully predicts both the majority (healthy) and minority (diseased) classes, it offers a much more realistic assessment of performance [80].

The Receiver Operating Characteristic (ROC) curve shows how successful classification models handle various data distributions because it provides a visual representation of their strengths and weaknesses in different situations according to research [81]. The construction of an ROC curve requires artists to plot their True Positive Rate value on the Y-axis while they place their False Positive Rate value on the X-axis according to the mathematical definition of this graph [81]. This two-dimensional graph helps researchers clearly observe how improving one aspect of the model may affect another. The Area Under the Curve (AUC) provides a measurement so that we can compare the overall performance of different predictive models [81].

However, when predicting rare conditions such as diabetes in highly imbalanced datasets, ROC curves may sometimes give an overly optimistic impression of a models true performance. This happens because the formula used to calculate the false positive rate includes true negatives in its denominator. When the dataset contains a very large number of healthy patients, even a noticeable increase in false positives may not significantly change the overall rate [79][82].

Lastly, as every evaluation metric highlights a different characteristic of a models predictive behavior, relying on a single measure is not sufficient for a comprehensive assessment. As demonstrated by the varying strengths and limitations of Accuracy, F-Measure and MCC each metric interprets model performance from a different mathematical perspective. Therefore, adding multiple metrics to assess the models performance would be a good practice. By keeping more than one evaluation metric, researchers can cross-validate their findings, figure out the hidden biases in imbalanced medical datasets, and ensure a transparent, robust, and reliable interpretation of the model's true real-world capabilities [83].

2.6 Explainability in Medical ML (XAI)

Even though ML algorithms can predict outcomes with high accuracy, their use in real-world clinics is often limited because they act as black boxes [84][85]. In healthcare, high accuracy scores are simply not enough. No matter how accurate a model is, healthcare professionals will simply not use a tool they cannot trust [86]. When doctors cannot see how an algorithm makes its decisions, it goes against the core values of patient care, because they cannot explain to their patients how a specific diagnosis was reached. Relying on these unexplainable tools threatens a doctor's ethical duties, especially the need to do no harm and respect patient choices. So, using predictive models in healthcare without providing clear explanations directly undermines fundamental medical ethics. Therefore, making AI transparent is absolutely a necessary thing to do to protect patient trust and public health [84].

To solve this black box problem, researchers have created Explainable AI (XAI) tools that can look inside complex algorithms. Local Interpretable Model-agnostic Explanations (LIME) and SHAP are currently two of the most popular solutions [87]. LIME works by creating a simple, readable backup model that mimics how the main algorithm behaves for one specific patient [86]. It makes small changes to the patient's data to see how the prediction shifts, which helps identify exactly which clinical features caused that specific outcome [87]. On the other hand, SHAP is built on the mathematical rules of cooperative game theory [88]. It treats each clinical feature (like age or glucose) as a player in a game. It then calculates an exact feature importance score for every variable, and after that, it shows exactly how much that feature turned the prediction away from the model's average baseline [87][88]. Both of these methods cut through the confusing math to provide clear rankings of which features matter most, bridging the gap between complex computer logic and human medical understanding [87][88].

In modern medical research, scientists are actively using XAI to look inside complex models that predict diabetes. Nowadays, researchers are combining powerful ensemble algorithms, like Extreme Gradient Boosting (XGBoost) and RF, with XAI tools like SHAP so that they can produce clear medical insights [87][89]. In a recent study, after analyzing the Pima Indian Diabetes Dataset, researchers found that glucose levels, BMI, and age were the most important factors for predicting the disease using SHAP [87].

Another recent study focusing on older adults in South Korea used an XGBoost algorithm with SHAP analysis. They found that high blood pressure, older age, hyperlipidemia, high heart rate, and severe stress were the biggest causes of diabetes risk [89]. By clearly showing how these health markers impact the model's decision, XAI allows doctors to actually understand how the algorithm works, helping them design better, personalized treatment plans for their patients [87][89].

3 Materials and Methodology

3.1 Research Design Overview

To answer our RQ-1, and RQ-3, we have conducted experiments on three publicly available datasets. Datasets used in this study are the PIMA Indian Diabetes Dataset, BRFSS-15, and the Diahealth dataset. And our RQ-2 has been answered through the observation of the evaluation results of our experiments where our literature review also helping us strongly. To find out how significantly data leakage from pre-split resampling inflates the model performance metrics, we have done both pre-split resampling and post-split resampling so that we can compare and conclude about how much inflated the results can be. Also, to find out which features most significantly contribute to minority class detection (Diabetes Positive) under a correct and unbiased pipeline, we have used the SHAP-based explainable artificial intelligence technique. Finally, we have performed comparative experiments under multiple preprocessing configurations, including different imputation strategies, datasets with and without outlier removal, and pipelines with and without feature selection. This allowed us to evaluate the impact of each preprocessing component on models' performance. For our experiments, we selected XGBoost and LightGBM as our primary classifiers, as both are gradient boosting tree-based models that have consistently demonstrated strong performance on tabular medical data, particularly in the presence of class imbalance. Their ensemble nature makes them less prone to overfitting and more robust when minority class samples are limited. We also included Naive Bayes as a probabilistic baseline, since its simplicity and different underlying assumptions allow us to observe whether the patterns found by more complex models hold true across different learning paradigms. Also, our experiments has been evaluated using some robust evaluation metrics like MCC, F1-Score, AUC-ROC.

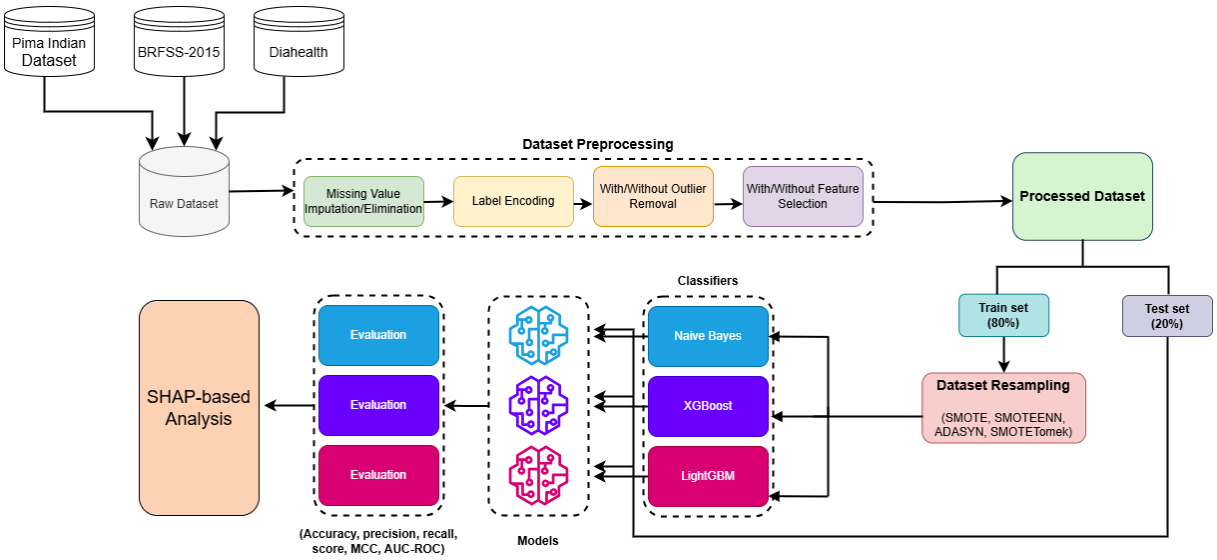


Figure 3.1: Methodology diagram for Post-split resampling pipeline

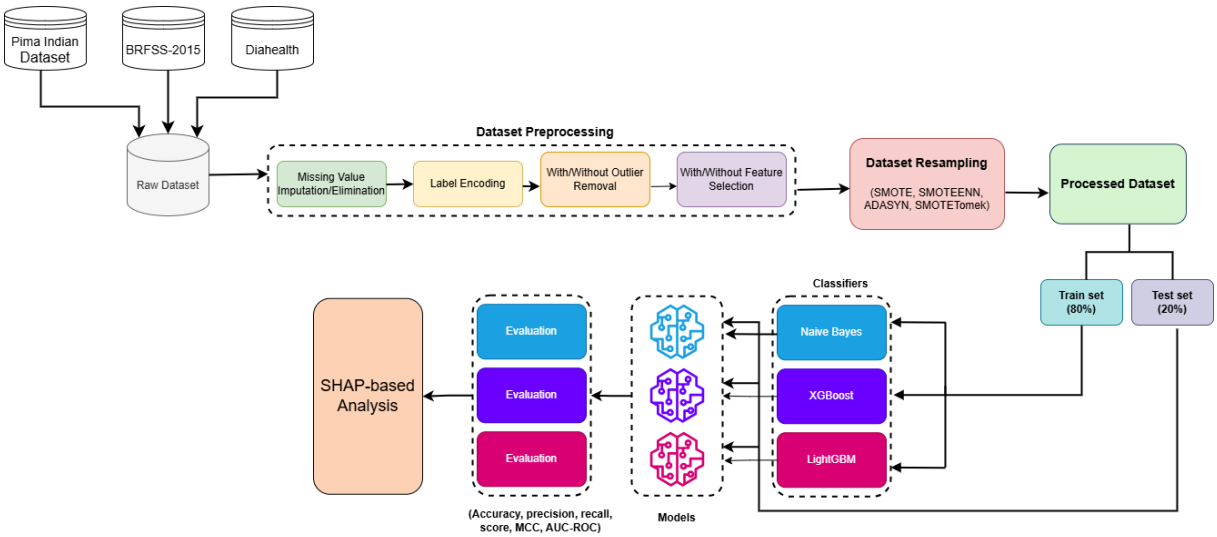


Figure 3.2: Methodology diagram for Pre-split resampling pipeline

The given Figure 3.1 and Figure 3.2 shows our complete methodological pipeline for post-split resampling and pre-split resampling approach. As mentioned in the diagrams, we have done several experiments based on some different combinations. The PIMA Indian Diabetes Dataset presented a specific preprocessing challenge known as the "zero value problem. In the original clinical study, unknown measurements were simply left blank [90]. However, when the data was moved to digital repositories, those blanks were replaced with zeros. As a result, the dataset now includes physically impossible readings, such as a blood pressure, glucose level, or BMI of zero [91]. To prevent the model from learning these distorted and impossible patterns, we have to identify these zeros and treat them as missing data before any resampling occurs [91]. So, due to the zero value problem of the PIMA indian diabetes dataset, we have added another layer at the pre-processing stage for the experiments of this dataset, which is Imputation/Elimination. But the BRFSS-2015 and Diahealth dataset did not have any missing values so this step is not included in the experiments pipeline of BRFSS-2015 and Diahealth dataset. Our research includes an extensive comparison of the outcomes of those different combinations and mainly emphasised on the outcome and findings of the two different pipelines. And after

discussion we will give a verdict for which pipeline would be methodologically correct and why. Also, throughout the discussions we have answered our RQs with reasoning.

3.2 Dataset description

This study uses three public datasets, the PIMA Indians Diabetes Dataset, the BRFSS 2015, and DiaHealth dataset. We chose these three because they are all very different. They cover different groups of people, vary in size, and contain different types of information. Because they are so varied, they help us prove that our methods work well in many different real-world situations, not just one. We picked PIMA indian dataset as our main test because many other diabetes studies use it. This makes it easy to compare our results with past research. It is a small dataset (768 patient records with 8 features of health info each). It also has some well-known data errors, like missing information being recorded as a "0" [91]. This makes it a perfect tool to compare between results and through the comparative analysis we might be able to find out the answers of our RQs. The BRFSS-2015 dataset is much larger and looks at a general population. It helps us test our tools on a dataset where there are many more healthy people than sick people. The dataset will be helpful to address the class imbalance problem. The dataset includes real medical records from South Asian patients. This helps us include a different demographic group and lets us test our tools on a dataset where the healthy-to-sick ratio is even more extreme.

3.2.1 PIMA Indians Diabetes Dataset

The National Institute of Diabetes and Digestive and Kidney Diseases originally collected the PIMA indian dataset. Anyone can access it for free through the UCI ML Repository [1]. Researchers created this dataset to help computers learn how to predict if someone has diabetes based on their health and body measurements [90]. The dataset contains health records from 768 women of PIMA Indian heritage. All the women were 21 years or older and lived near Phoenix, Arizona. The details are shown in Table 3.1.

Table 3.1: Summary of PIMA Indian Dataset Attributes and Class Distribution

Attribute	**Description / Value**
Total Instances	768
Participant Demographics	Female, PIMA Indian heritage, Age $\geq$ 21 years
Location	Near Phoenix, Arizona
Number of Input Features	8
Target Variable	Binary (0: Non-diabetic, 1: Diabetic)
Class Distribution (Negative)	500 (65.1%)
Class Distribution (Positive)	268 (34.9%)
Imbalance Ratio	$\sim$1.87:1
Missing Values	None (encoded as zeros in 5 features)

The dataset comprises eight input features, each capturing a clinically relevant aspect of patient health, alongside a binary outcome label. A full description of each feature is provided in Table 3.2.

Table 3.2: PIMA Indian Dataset Feature Names, Descriptions, and Measurement Units

Feature Name	**Description**	**Measurement / Unit**
Pregnancies	Total number of times pregnant	Count
Glucose	Plasma glucose concentration (2-hour oral glucose tolerance test)	mg/dL
BloodPressure	Diastolic blood pressure	mm Hg
SkinThickness	Triceps skinfold thickness	mm
Insulin	2-Hour serum insulin	μU/ml
BMI	Body mass index	kg / m^2
DiabetesPedigreeFunction	Genetic likelihood of diabetes based on family history	Continuous Score
Age	Age of the patient	Years
Outcome (Label)	Diabetes diagnosis (1: Positive, 0: Negative)	Binary (0 or 1)

To provide an overview of data quality, Table 3.3 presents the attribute-wise statistical summary of the PIMA indian dataset features, including count, minimum, maximum, mean, standard deviation, and median for each feature.

Table 3.3: Attribute-wise Statistical Summary of PIMA indian dataset Features (N = 768). (Without imputation)

No.	**Feature Name**	**Count**	**Min**	**Max**	**Mean**	**Std Dev**	**Median**	**Feature Type**
0	Pregnancies	768	0.0	17.0	3.84	3.36	3.0	Continuous
1	Glucose	768	0.0	199.0	120.89	31.97	117.0	Continuous
2	BloodPressure	768	0.0	122.0	69.10	19.35	72.0	Continuous
3	SkinThickness	768	0.0	99.0	20.53	15.95	23.0	Continuous
4	Insulin	768	0.0	846.0	79.79	115.24	30.5	Continuous
5	BMI	768	0.078	67.1	31.99	7.88	32.0	Continuous
6	DiabetesPedigreeFunction	768	0.078	2.42	0.47	0.33	0.37	Continuous
7	Age	768	0.0	81.0	33.24	11.76	29.0	Continuous

Looking at Table 3.3, five features named, Glucose, Blood Pressure, Skin Thickness, Insulin, and BMI, showing a lowest value of zero. Since a living person cannot have a zero for these, it means the makers of the dataset used a zero whenever they were missing real information. As mentioned previously, other researchers have also noticed this problem in the past. We also saw that the insulin numbers varied a lot from person to person. Another measurement, which looks at family history to guess the risk of diabetes, also showed a very wide range of scores. Because of these messy details, we have to carefully clean and adjust the data before we can use it to teach our computer model. We explain exactly how we do this in Section 3.3.

3.2.2 BRFSS 2015

The CDC collected the Behavioral Risk Factor Surveillance System (BRFSS) 2015 [92] dataset by doing phone interviews with people across all 50 US states. The very first version was huge, including answers from over 440,000 people to over 300 questions about their health, daily habits, and illnesses. Later, a researcher named Teboul shared a cleaner, simpler version on a website called Kaggle to help people build computer learning models. This newer version uses just 21 key pieces of information about a person's lifestyle, living situation, and general health to predict if they have diabetes [93]. This cleaner version includes answers from over 250,000 adults. It originally sorted people into three groups: those without diabetes, those with prediabetes, and those with diabetes. Since our study only wants to compare people who definitely have diabetes against those who definitely do not, we removed 4,631 instances, people who had prediabetes. This left us with a final group of about 249,000 people. In this group, almost 86 % of the people do not have

diabetes, while about 14 % do. This means there are roughly six healthy people for every one person with diabetes in this data. The details are shown in Table 3.4.

Table 3.4: Summary of BRFSS 2015 Dataset Attributes and Class Distribution

Attribute	**Description / Value**
Total Instances (raw)	253,680
Prediabetic Instances Excluded	4,631 (Class 1 removed for binary classification)
Instances Used (binary)	249,049
Participant Demographics	Adults aged 18 and older, across all 50 U.S. states
Data Collection Method	Telephone-based health survey (landline + cell phone)
Source	CDC BRFSS 2015 Survey (cleaned version via Kaggle)
Number of Input Features	21
Target Variable	Binary (0: Non-diabetic, 1: Diabetic)
Class Distribution (Negative)	213,703 (85.8%)
Class Distribution (Positive)	35,346 (14.2%)
Class Imbalance Ratio	∼6.04:1
Missing Values	None (pre-cleaned dataset)

The 21 input features fall into four broad categories: clinical and physiological indicators (HighBP, HighChol, CholCheck, BMI, Stroke, HeartDiseaseorAttack), lifestyle and behavioural factors (Smoker, PhysActivity, Fruits, Veggies, HvyAlcoholConsump), healthcare access indicators (AnyHealthcare, NoDocbcCost), and general health and demographic attributes (GenHlth, MentHlth, PhysHlth, DiffWalk, Sex, Age, Education, Income) [92][93]. The majority of features are binary (0/1). General health, mental health, physical health, age, education, and income are encoded as ordinal scales, while BMI is the only continuous numeric variable [93]. A full description of each feature is provided in Table 3.5.

Table 3.5: BRFSS 2015 Feature Names, Descriptions, Types, and Value Ranges

Feature Name	**Description**	**Type**	**Values / Range**
HighBP	Ever told by a doctor they have high blood pressure	Binary	0 = No 1 = Yes
HighChol	Ever told by a doctor they have high cholesterol	Binary	0 = No 1 = Yes
CholCheck	Cholesterol check within the past 5 years	Binary	0 = No 1 = Yes
BMI	Body Mass Index	Continuous	Numeric (integer)
Smoker	Smoked at least 100 cigarettes in their lifetime	Binary	0 = No 1 = Yes
Stroke	Ever told they had a stroke	Binary	0 = No 1 = Yes
HeartDisease-orAttack	Coronary heart disease or myocardial infarction history	Binary	0 = No 1 = Yes
PhysActivity	Physical activity in the past 30 days (outside regular job)	Binary	0 = No 1 = Yes
Fruits	Consumes fruit one or more times per day	Binary	0 = No 1 = Yes
Veggies	Consumes vegetables one or more times per day	Binary	0 = No 1 = Yes
HvyAlcohol-Consump	Heavy drinker (men >14 drinks/week, women >7 drinks/week)	Binary	0 = No 1 = Yes
AnyHealthcare	Has any form of health care coverage	Binary	0 = No 1 = Yes
NoDocbcCost	Could not see a doctor in past 12 months due to cost	Binary	0 = No 1 = Yes
GenHlth	General health self-rating	Ordinal	1 = Excellent $\ldots$ 5 = Poor
MentHlth	Days in past 30 with poor mental health	Ordinal	030 days
PhysHlth	Days in past 30 with poor physical health	Ordinal	030 days
DiffWalk	Serious difficulty walking or climbing stairs	Binary	0 = No 1 = Yes
Sex	Biological sex of the respondent	Binary	0 = Female 1 = Male
Age	Age category of the respondent	Ordinal	1 (1824) $\ldots$ 13 ($\geq$80 years)
Education	Highest level of education completed	Ordinal	1 (No schooling) $\ldots$ 6 (College graduate)
Income	Annual household income category	Ordinal	1 (<\$10,000) $\ldots$ 8 ($\geq$ \$75,000)
Diabetes_012 (Label)	Diabetes diagnosis class (prediabetic class excluded)	Ordinal $\rightarrow$ Binary	0 = No diabetes 2 = Diabetes

Table 3.6 presents the attribute-wise statistical summary of the BRFSS 2015 features after prediabetic exclusion. As the dataset contains no missing values in the cleaned version [93], the focus of preprocessing is on outlier removal and addressing the class imbalance, both of which are discussed in Section 3.3. Notable observations from Table 3.6 include the high cholesterol check adherence (CholCheck mean = 0.96, median = 1.0), indicating that most respondents had received a cholesterol check within the past five years. Physical activity (PhysActivity mean = 0.76) and vegetable consumption (Veggies mean = 0.81) are also reported at relatively high rates. Mental and physical health

poor-day counts (MentHlth, PhysHlth) both have a median of zero, indicating that the majority of respondents reported no poor-health days in the past month, though the high standard deviations (7.38 and 8.68, respectively) reflect substantial variation within the population.

Table 3.6: Attribute-wise Statistical Summary of BRFSS 2015 Features (after prediabetic exclusion, N = 249,049)

Feature Name	**Count**	**Min**	**Max**	**Mean**	**Std Dev**	**Median**	**Feature Type**
HighBP	249,049	0	1	0.43	0.49	0.0	Binary
HighChol	249,049	0	1	0.42	0.49	0.0	Binary
CholCheck	249,049	0	1	0.96	0.19	1.0	Binary
BMI	249,049	12	98	28.34	6.59	27.0	Continuous
Smoker	249,049	0	1	0.44	0.50	0.0	Binary
Stroke	249,049	0	1	0.04	0.20	0.0	Binary
HeartDiseaseorAttack	249,049	0	1	0.09	0.29	0.0	Binary
PhysActivity	249,049	0	1	0.76	0.43	1.0	Binary
Fruits	249,049	0	1	0.63	0.48	1.0	Binary
Veggies	249,049	0	1	0.81	0.39	1.0	Binary
HvyAlcoholConsump	249,049	0	1	0.06	0.23	0.0	Binary
AnyHealthcare	249,049	0	1	0.95	0.22	1.0	Binary
NoDocbcCost	249,049	0	1	0.08	0.28	0.0	Binary
GenHlth	249,049	1	5	2.50	1.07	2.0	Ordinal
MentHlth	249,049	0	30	3.16	7.38	0.0	Ordinal
PhysHlth	249,049	0	30	4.20	8.68	0.0	Ordinal
DiffWalk	249,049	0	1	0.17	0.37	0.0	Binary
Sex	249,049	0	1	0.44	0.50	0.0	Binary
Age	249,049	1	13	8.01	3.06	8.0	Ordinal
Education	249,049	1	6	5.06	0.98	5.0	Ordinal
Income	249,049	1	8	6.07	2.07	7.0	Ordinal

The class imbalance ratio of 6.04:1 in BRFSS 2015. This imbalance is a well-known characteristic of population-level health survey data and requires targeted resampling or cost-sensitive strategies prior to model training, as discussed in Section 3.3.

3.2.3 DiaHealth

Researchers from United International University and Southeast University in Bangladesh created the DiaHealth dataset [72]. The dataset contains details like a patient's family medical history, daily habits, body measurements, and vital signs, all to help predict whether they have Type 2 diabetes [72][64]. This collection contains real medical records from 5,437 patients in Bangladesh, tracking 14 different pieces of health information for each person [72]. A huge majority of the people in this group do not have the disease. Specifically, 5,093 of them are diabetes-free, while only 344 actually have it. This means there are almost 15 healthy people for every single person with diabetes, making it the most unevenly balanced dataset out of the three we are using. You can find a summary of these details in Table 3.7.

Table 3.7: Summary of DiaHealth Dataset Attributes and Class Distribution

Attribute	**Description / Value**
Total Instances	5,437
Participant Demographics	Real-world Bangladeshi clinical cohort (mixed age and gender)
Data Collection Source	United International University & Southeast University, Bangladesh
Repository	Mendeley Data, released September 2024
Number of Input Features	14
Target Variable	Binary (0: Non-diabetic, 1: Diabetic)
Class Distribution (Negative)	5,093 (93.7%)
Class Distribution (Positive)	344 (6.3%)
Class Imbalance Ratio	∼14.81:1 (Severe)
Missing Values	None

The 14 input features span four categories: demographic attributes (Age, Gender), anthropometric and physiological measurements (Height, Weight, BMI, Pulse Rate, Systolic BP, Diastolic BP, Glucose), lifestyle factors (Smoking History, Physical Activity, Dietary Habits), and comorbidity and family history indicators (Hypertension, Cardiovascular Disease, Family History of Diabetes) [72]. Six features are continuous numeric measurements and eight are binary categorical indicators. Unlike PIMA indian dataset, which is limited to a single ethnic group with no lifestyle or comorbidity features, DiaHealth provides a richer, more representative clinical picture for South Asian populations. A complete description of each feature is provided in Table 3.8.

Table 3.8: DiaHealth Feature Names, Descriptions, Types, and Values/Units

Feature Name	**Description**	**Type**	**Values / Unit**
Age	Age of the patient	Continuous	Years
Gender	Biological sex of the patient	Binary	0 = Female 1 = Male
Pulse Rate	Resting heart rate	Continuous	bpm
Systolic BP	Systolic (peak) blood pressure	Continuous	mm Hg
Diastolic BP	Diastolic (resting) blood pressure	Continuous	mm Hg
Glucose	Blood glucose concentration	Continuous	mg/dL (or mmol/L)
Height	Height of the patient	Continuous	Meters
Weight	Weight of the patient	Continuous	kg
BMI	Body Mass Index (derived from height and weight)	Continuous	kg/m^2
Smoking History	Whether the patient has a history of smoking	Binary	0 = No 1 = Yes
Physical Activity	Whether the patient engages in regular physical activity	Binary	0 = No 1 = Yes
Dietary Habits	Whether the patient follows a healthy or regulated diet	Binary	0 = No 1 = Yes
Hypertension	Whether the patient has a history of hypertension	Binary	0 = No 1 = Yes
Cardiovascular Disease	Whether the patient has a history of cardiovascular disease	Binary	0 = No 1 = Yes
Family History of Diabetes	Whether any first-degree relative has been diagnosed with diabetes	Binary	0 = No 1 = Yes
Diabetic (Label)	Binary class label for Type 2 Diabetes status	Binary	0 = Non-diabetic 1 = Diabetic

Table 3.9 presents the attribute-wise statistical summary for the nine continuous fea-

tures in DiaHealth. The binary features (Gender, Smoking History, Physical Activity, Dietary Habits, Hypertension, Cardiovascular Disease, Family History of Diabetes) are excluded from Table 3.9, as their distributions are more meaningfully captured through class frequency counts.

Table 3.9: Attribute-wise Statistical Summary of DiaHealth Continuous Features (N = 5,437)

Feature Name	**Count**	**Min**	**Max**	**Mean**	**Std Dev**	**Median**	**Feature Type**
Age	5,437	8.00	112.00	45.53	14.32	45.00	Continuous
Pulse Rate	5,437	5.00	133.00	76.77	12.29	76.00	Continuous
Systolic BP	5,437	62.00	231.00	133.86	22.29	130.00	Continuous
Diastolic BP	5,437	45.00	119.00	82.06	12.49	81.00	Continuous
Glucose	5,437	0.00	33.46	7.54	2.92	6.92	Continuous
Height	5,437	0.36	1.96	1.55	0.08	1.55	Continuous
Weight	5,437	3.00	100.70	53.63	10.09	53.00	Continuous
BMI	5,437	1.22	574.13	22.47	8.78	21.87	Continuous

Looking at Table 3.9, a few things stand out. First, the lowest blood sugar level is zero. Just like in the first dataset, this is impossible for a living person and means someone likely typed a zero when they were missing the real number. Second, the BMI numbers cover a surprisingly wide range, going from 1.22 all the way up to 574.13. This shows there are some extreme errors in the data that we need to fix. Third, the ages range from 8 to 112 years old. This is a much wider age gap than we normally see for Type 2 diabetes patients, meaning the highest and lowest ages might just be typing mistakes. On average, the top blood pressure number is around 134, which makes sense since many patients in this group have high blood pressure. Even though the original creators call this a high-quality dataset with no missing information [72], we still need to carefully clean up those zero blood sugar readings and extreme BMI numbers before we can actually use the data. We explain how we do this in Section 3.3. Out of our three datasets, DiaHealth has the most uneven mix of healthy and sick patients. For every person with diabetes, there are almost 15 people without it. This gap is more than twice as big as the one in the BRFSS 2015 data and about eight times bigger than the one in the PIMA indian dataset data. This uneven mix matches what we actually see in the real world for this specific population. However, the presence of missing values and outliers in the continuous features must be verified during preprocessing, and any class imbalance identified must be addressed through appropriate resampling strategies prior to model training, though the dataset is new but as discussed in Section 3.3. Rahman et al. [64] confirmed the utility of this dataset by successfully applying a LightGBM-based framework with Boruta feature selection, achieving an accuracy of 0.8516 and F1-score of 0.8541 on DiaHealth.

3.3 Pre-Processing

3.3.1 PIMA Indian Diabetes Dataset

The PIMA indian dataset dataset has 768 instances. Weve previously mentioned that the PIMA indian dataset encodes missing clinical measurements (such as blood pressure and BMI) with a biological impossibility, a value of "0". Before any models could be trained, these zeroes were converted to standard NaN (Not a Number) values so that we could quantify the true information about missing data. The distribution of these missing values is detailed in Table 3.10

Table 3.10: Missing Values in PIMA Indian Diabetes Dataset

Clinical Feature	**Number of Missing Values**	**Percentage of Missing Data**
Glucose	5	0.65%
Blood Pressure	35	4.56%
Skin Thickness	227	29.56%
Insulin	374	48.70%
BMI	11	1.43%

In order to observe how missing data handling impacts a model's performance, two parallel experimental tracks were conducted. One is elimination of missing values, and another one is imputation of missing values. In the elimination part, any row containing a missing value was entirely dropped from the dataset. This ensures 100% authentic data but it drastically reduces the dataset size.

So, to prevent this severe data loss, the imputation technique was applied using the Multiple Imputation by Chained Equations (MICE) technique. MICE is not like simple mean or median imputation, which can severely distort the variance of a dataset, MICE is an advanced, robust algorithm which models each missing feature as a function of all other features in the dataset. It runs multiple regression iterations to predict and fill in the missing values while accounting for statistical uncertainty. According to Azur et al. [94] MICE is highly preferred in medical datasets because it keeps the complex, non-linear relationships between clinical variables without having too much bias in the dataset. In order to keep the models aware between categorical values and numerical values, weve converted the datatypes for the numerical values to float64, while the categorical values were kept as int64. The dataset had outlier values that fall far outside the normal physiological range. To identify these anomalies, data visualization was conducted using the Seaborn and Matplotlib libraries. Box plots revealed significant extreme values, particularly within the Insulin, SkinThickness, BloodPressure, BMI, and DiabetesPedigreeFunction features. To find out if outlier is an issue or not, we have applied two configurations. The first one is, predicting keeping the outliers as it is, and the second one is, removing outliers through the Interquartile Range (IQR) method. The IQR is a highly robust statistical measure of statistical dispersion, pioneered by John Tukey (1977) [95], that calculates the spread of the middle 50% of the data.

The formula of IQR is defined as:

$$\text{IQR} = Q_3 - Q_1 \tag{3.1}$$

Where Q_3 is the 75th percentile and Q_1 is the 25th percentile of the data. Any data point falling below the lower bound or above the upper bound is flagged as an outlier and subsequently removed:

$$\textbf{Lower Bound} = Q_1 - (1.5 \times \text{IQR}) \tag{3.2}$$

$$\textbf{Upper Bound} = Q_3 + (1.5 \times \text{IQR}) \tag{3.3}$$

Because extreme physiological values in medical data can sometimes represent severe, but genuine, stages of a disease rather than recording errors, we maintained a dual-track experimental setup. One variation of the dataset was processed with IQR outlier removal

to observe how models perform on normalized, strictly bounded data, while a parallel dataset was tested without outlier removal to evaluate the models' natural robustness to extreme clinical variance.

3.3.2 BRFSS-2015

The BRFSS-2015 dataset is relatively clean than the PIMA indian dataset and does not contain any missing or zero-encoded values. As there were no missing values in the dataset, no data imputation or row elimination steps were applied to this dataset in the pre-processing step. However, to align this dataset with our research objective, we needed to perform label encoding on the target variable. The BRFSS dataset contains 253680 instances. The name of the output label of the dataset was 'Diabetes012', this column contained three values, '0', '1', '2'. Where 0 referred to non-diabetic, 1 referred to pre-diabetic and 2 referred to diabetic. So, to maintain a consistent single-label binary output for diabetes across all our experiments, we replaced the value of 2 with 1 in the target column. This converts the target variable into a standard binary format (0 for negative, 1 for positive), to ensure that the models focus entirely on a clear positive or negative diagnosis. After Label encoding the distribution was,(0=213703 instances , 1=35346 instances). Just like the previous dataset, in order to keep the models aware of the distinction between categorical and numerical values, weve converted the datatypes for the numerical features to float64, while the categorical features and the newly formatted binary labels were kept as int64. The only numerical feature in this dataset was BMI, all the other features were categorical. To identify if there is any anomaly in the continuous variables, data visualization was conducted using the Seaborn and Matplotlib libraries. Box plots draw the actual picture of extreme values falling far outside the normal physiological range for certain health indicators. To handle the anomalies, we applied the exact same dual-track configuration used for the PIMA indian dataset (as detailed in Section 3.3.1). One variation of the BRFSS-2015 dataset was processed using the IQR method to remove outliers and observe how the models perform on normalized data. Simultaneously, a parallel dataset was tested without outlier removal to evaluate the models' natural robustness to extreme, yet potentially genuine, clinical variance.

3.3.3 Diahealth

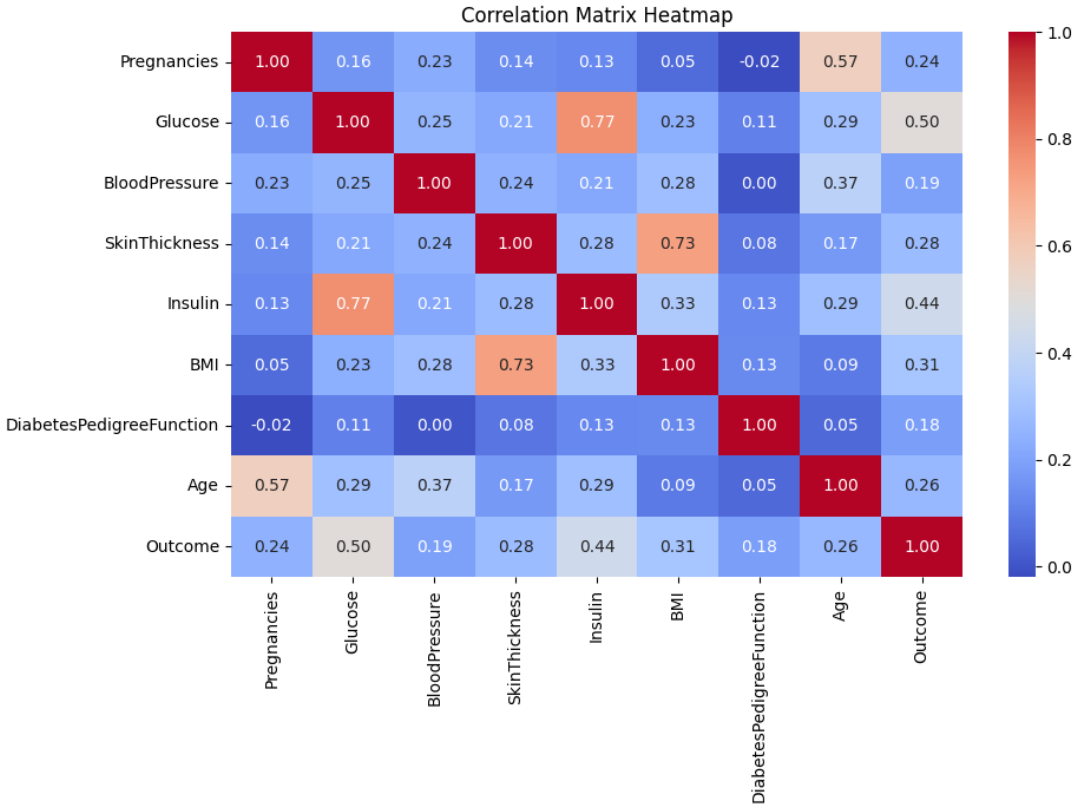
Similar to the BRFSS-2015 dataset, the Diahealth dataset was already well-structured and did not suffer from missing values. Therefore, we did not perform any data imputation or row elimination steps for this dataset. However, the Diahealth dataset had text data in gender and diabetic columns. Both were categorical columns. To resolve this, we have used a Label Encoder to iterate through the dataset, and automatically identify all text-based columns and transform their categorical text into distinct numerical values. After the label encoding, we followed our standard type-casting procedure. To keep the models aware of the distinction between measurement types, the datatypes for the continuous numerical features were converted to float64, while the newly encoded categorical features and the target labels were set to int64. Finally, to check for physiological anomalies in the continuous variables, we visualized the data using Seaborn and Matplotlib box plots. As there were extreme values present, we applied the exact same dual-track configuration used for the previous datasets (as detailed in Section 3.3.1). One variation of the Diahealth dataset was processed using the IQR method to cleanly remove outliers, while a parallel dataset was kept completely intact. This helped us to calculate how well the models can handle natural, extreme clinical variance compared to strictly normalized data.

3.4 Feature selection

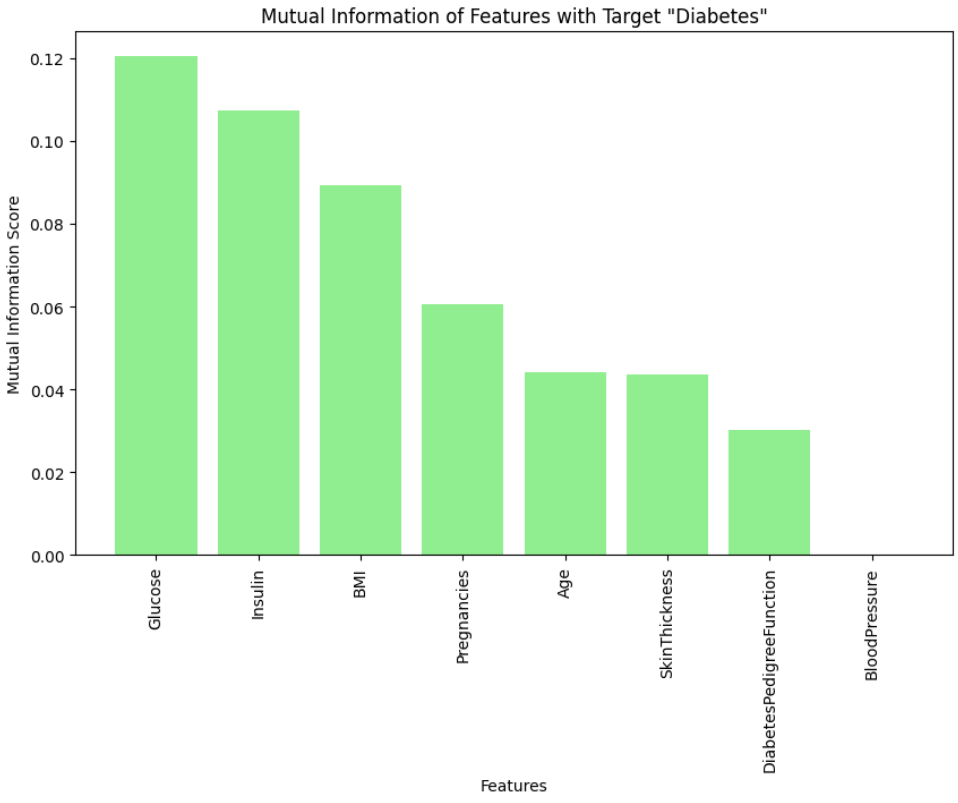
After handling missing values and addressing outliers, we carried out feature selection to maintain the dimensionality of our datasets manageable and to reduce the risk of model overfitting of our training data. Healthcare datasets often contain features that contribute very little useful information for prediction, and including them creates unnecessary noise in the learning process of the model. Synthetic oversampling techniques like SMOTE are popular, but they are often ineffective when applied to datasets with a large number of features. Specifically, in situations where the data contains high levels of noise (a small signal-to-noise ratio), generating artificial data points can actually harm classifier performance, often producing results that are worse than simple undersampling or uncorrected models [96].To address this, we followed a two-step approach before applying any resampling. We used Pearson correlation analysis to find and remove features that contained a low score. Secondly, we calculated Mutual Information (MI) scores to measure how much each remaining feature contributes to diabetes prediction. The features that scored too low on MI were dropped.

3.4.1 PIMA Indians Diabetes Dataset

The PIMA Indians Diabetes Dataset originally included eight clinical features. After running the correlation analysis, we observed that some features are highly correlated with others, which makes them unimportant. And also, MI scores revealed that some features had a very weak relation with diabetes prediction. Based on those findings, DiabetesPedigreeFunction and BloodPressure" were removed. The cleaned dataset was left with six features, and among them, Glucose and Insulin were the strongest predictors based on analysis, and played a significant role in the prediction process.



(a) Correlation-based feature selection



(b) Mutual information-based feature selection

Figure 3.3: Feature selection graphs of the PIMA indian dataset: correlation-based and mutual information-based selection results.

3.4.2 BRFSS-2015 Dataset

The BRFSS 2015 dataset originally included 21 clinical and lifestyle features. After running the correlation analysis, we observed that several features shared overlapping information, which made them statistically redundant. The MI scores demonstrated that some features shared almost no actual connection with the diabetes label. Based on these findings, CholCheck, Smoker, Stroke, PhysActivity, Fruits, Veggies, HvyAlcoholConsump, AnyHealthcare, NoDocbcCost, MentHlth, DiffWalk, Sex, Education, and Income were removed. The cleaned dataset is left with seven features. Among them, GenHlth and HighBP are the most important predictors of the prediction process.

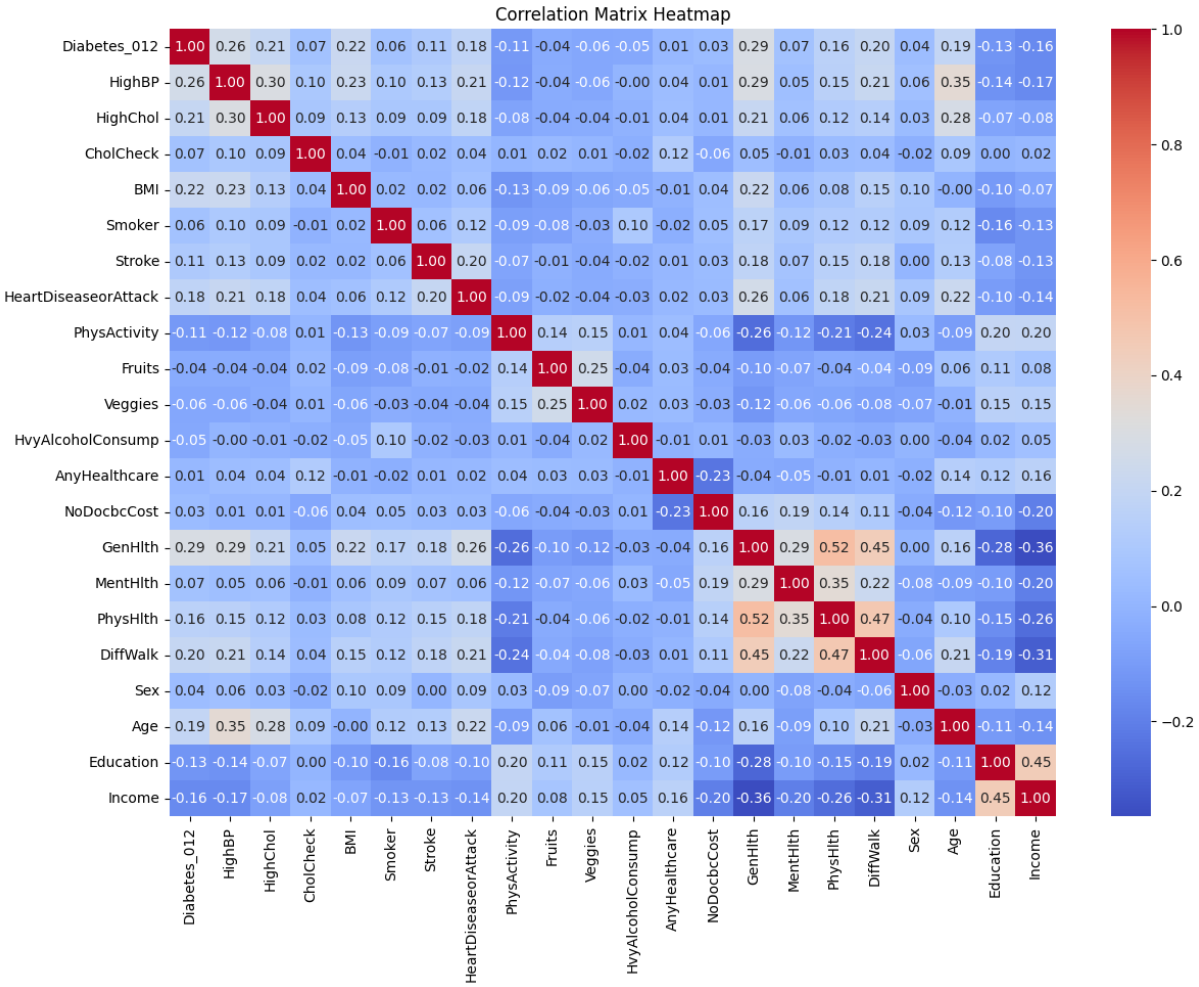


Figure 3.4: Correlation-based feature selection graph of the BRFSS dataset

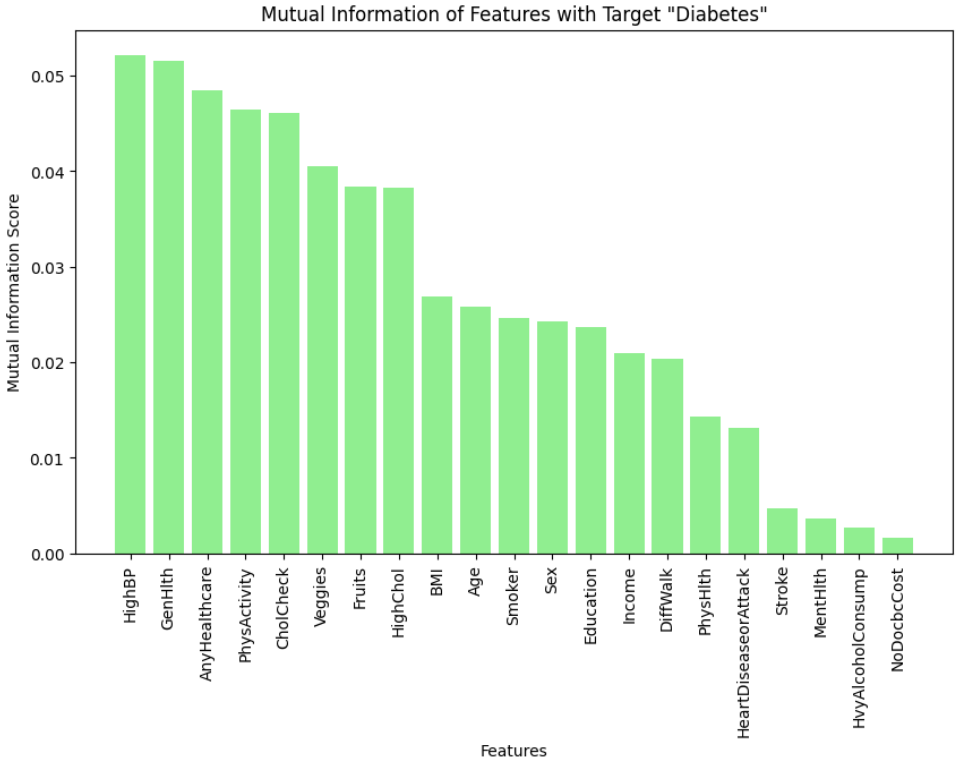
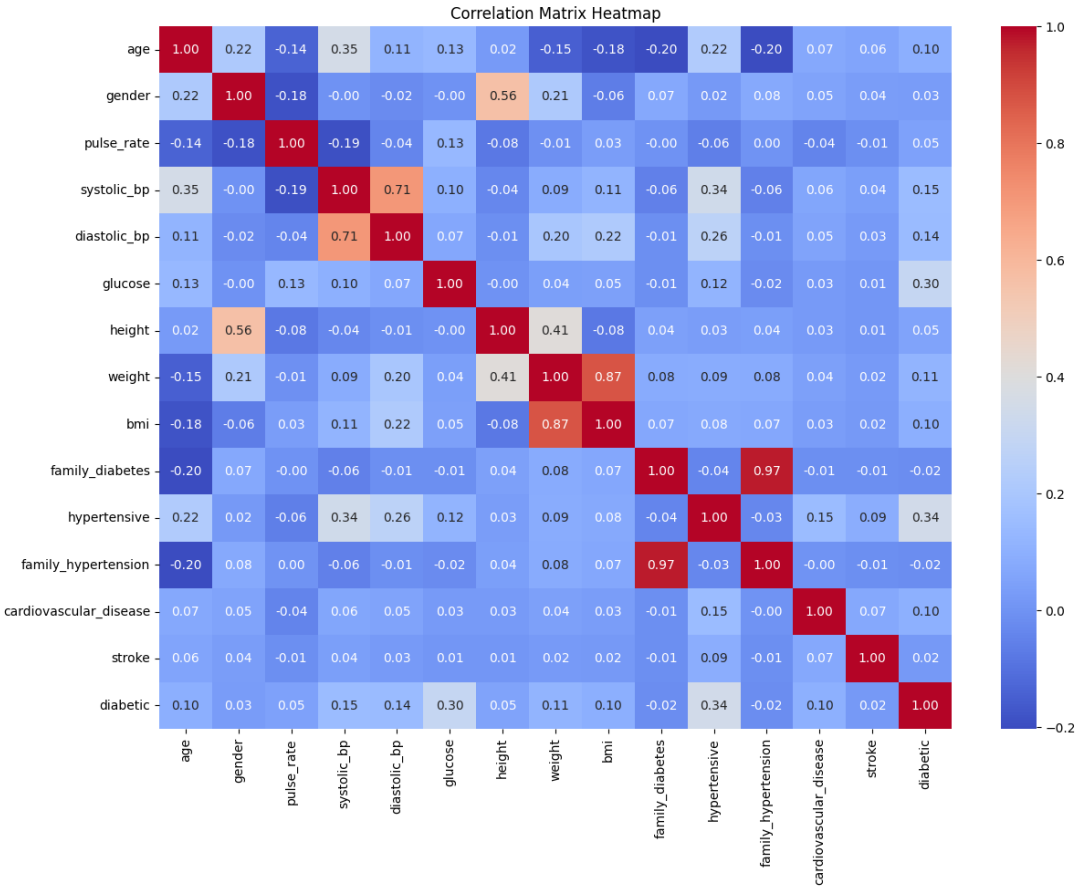


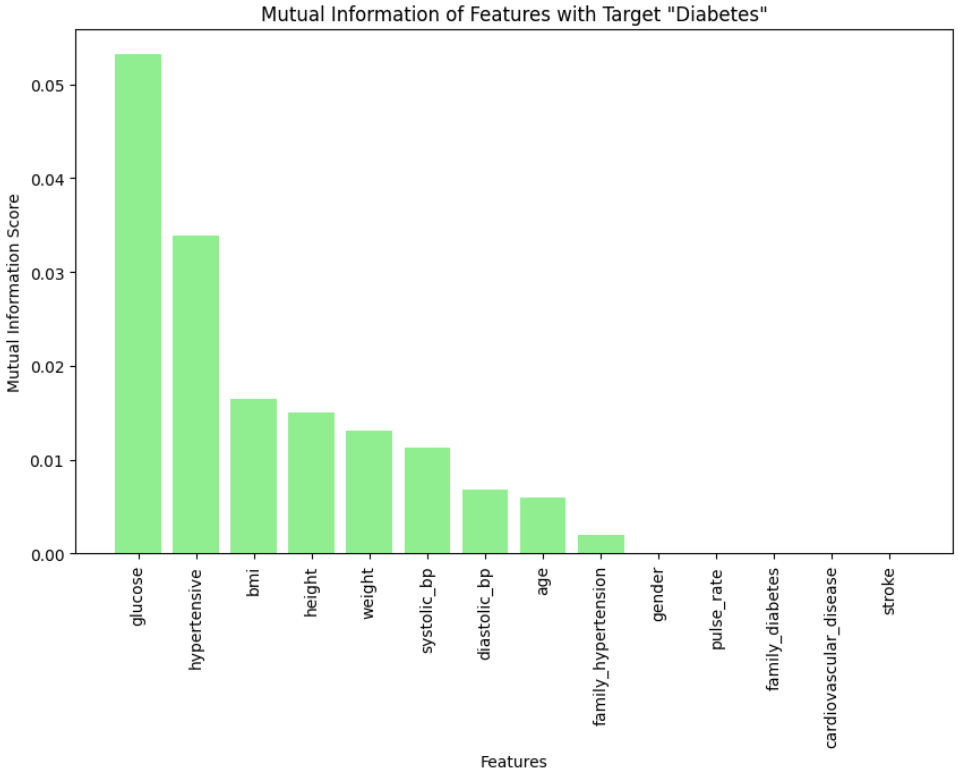
Figure 3.5: Mutual information-based feature selection graph of the BRFSS dataset

3.4.3 DiaHealth Dataset

The DiaHealth dataset originally included 14 clinical features. After running the correlation analysis, we observed that several features shared overlapping information, which made them statistically redundant. The MI scores also demonstrated that some features shared almost no actual connection with the diabetes prediction. Based on these findings, gender, pulse_rate, height, family_diabetes, family_hypertension, cardiovascular_disease, and stroke were removed. The cleaned dataset was left with seven features, and among them, glucose and hypertensive were the strongest predictors based on the analysis, playing a significant role in the prediction process.



(a) Correlation-based feature selection



(b) Mutual information-based feature selection

Figure 3.6: Feature selection graphs of the DiaHealth dataset: correlation-based and mutual information-based selection results.

3.5 Resampling techniques

The datasets in this study have a major issue with class imbalance. It is observed that when the training data contains many more instances of the majority class than those of the minorities [75]. When we train ML models on this kind of uneven data, the system often struggles to learn about the minority class. As a result, the model usually becomes biased and simply guesses the larger group most of the time [74][75]. To fix this problem and balance the data, we used four different resampling methods to help the models learn properly [77].

3.5.1 Synthetic Minority Over-sampling Technique (SMOTE)

Instead of just copying the minority examples for balancing the data, which can cause the model to memorize the data instead of actually learning (overfitting), SMOTE fixes the imbalance by creating brand new synthetic or artificial examples [74]. The algorithm looks at a minority class sample (x_i) and finds its closest minority class neighbors [74]. It picks one of these neighbors (x_{zi}) at random and creates a new data point right on the line connecting the two [74][77].

Mathematically, it creates this new artificial data point (s_i) using this formula:

$$s_i = x_i + (x_{zi} - x_i) \times \lambda \tag{3.4}$$

where λ is a random number between 0 and 1 [77]. This method helps spread out the minority class area, giving the model a much better chance to learn general patterns instead of just focusing on a few specific points [74].

3.5.2 Adaptive Synthetic Sampling (ADASYN)

While SMOTE creates the exact same amount of artificial data for every minority example, ADASYN works a bit differently. It figures out which minority examples are the hardest for the model to learn and focuses its effort there [77]. If a minority data point is heavily surrounded by majority data points, ADASYN sees it as hard to learn and generates more artificial data right in that specific area [77].

Mathematically, ADASYN looks at the K closest neighbors for each minority example (x_i) and calculates a ratio (r_i):

$$r_i = \frac{\Delta_i}{K} \tag{3.5}$$

where Δ_i is the number of majority class examples found within those K neighbors [77]. This ratio is then adjusted to create a density distribution $\hat{r}_i$ (so that $\sum \hat{r}_i = 1$) [77]. To figure out exactly how many new artificial examples (g_i) to make for each specific minority point, it uses this formula:

$$g_i = \hat{r}_i \times G \tag{3.6}$$

where G is the total number of artificial examples needed to successfully balance the dataset [77].

3.5.3 Hybrid Resampling: SMOTETomek and SMOTEENN

While adding artificial data helps balance the classes, it can sometimes push these new minority points too far into the majority group's space. This creates messy, overlapping borders between the classes, which confuses the model [75]. To fix this and create clear

borders, hybrid methods combine adding data (over-sampling) with cleaning the data [75].

- **SMOTETomek:** This method first uses SMOTE to balance the data, then uses 'Tomek links' to clean it up [75]. A Tomek link happens when two data points from different classes are right next to each other as closest neighbors [75]. Since these points usually cause confusion or are just noisy data, SMOTETomek finds them and deletes them. This leaves a much sharper, cleaner dividing line between the two classes [75].
- **SMOTEENN (SMOTE + Edited Nearest Neighbors):** This method uses the Edited Nearest Neighbor (ENN) rule to do a deeper clean after running SMOTE [75]. ENN looks at every single data point in the set. If a point's class is different from the majority of its closest neighbors, it gets removed [75]. Because ENN cleans up messy data from both classes, it leaves behind a very pure and clear dataset for the models to learn from [75].

3.6 Machine Learning Algorithms

3.6.1 Gaussian Naïve Bayes

In order to set the baseline against more advanced ensemble models, a simple Gaussian Naïve Bayes model was included in the methodology. Although it is simpler than tree-based algorithms, it has consistently been recognized as a highly reliable and efficient tool for medical classification and prediction tasks [97]. According to Bayes' Theorem, this algorithm learns quickly by making a "naïve" assumption. It assumes that every clinical feature is completely independent of the others once the final diagnosis (class label) is known [97].

Given a feature vector $X = (X_1, X_2, \ldots, X_n)$ and a target classifier C, the algorithm assigns the most likely class by calculating the posterior probability using the following formula:

$$P(C \mid X) = P(C) \times \prod_{i=1}^{n} P(X_i \mid C) \tag{3.7}$$

Because it assumes every feature operates independently, it uses all attributes equally to make its predictions, making it highly robust against noisy or missing data in medical datasets [97].

3.6.2 XGBoost (eXtreme Gradient Boosting)

XGBoost is a highly scalable and powerful algorithm, commonly regarded as an industry-standard method for handling classification problems [98]. Instead of building decision trees independently, XGBoost builds them sequentially. Each new tree is specifically trained to learn from and correct the lingering errors (residuals) made by the trees that came before it [98]. To stop the model from simply memorizing the training data, a common problem known as overfitting, XGBoost uses built-in regularization. This means it mathematically penalizes the model if the trees become unnecessarily complex [98].

Mathematically, the final prediction $\hat{y}_i$ for a given patient example is the sum of predictions from K additive trees:

$$\hat{y}_i = \phi(x_i) = \sum_{k=1}^{K} f_k(x_i) \tag{3.8}$$

To learn these tree functions, the algorithm minimizes a regularized objective function:

$$\mathcal{L}(\phi) = \sum_i l(\hat{y}_i, y_i) + \sum_k \Omega(f_k) \tag{3.9}$$

Here, l represents the loss function, which measures the difference between the actual label y_i and the prediction $\hat{y}_i$, while the Ω term represents the penalty applied to the complexity of the trees, ensuring that the final model remains smooth, generalizable, and effective on unseen data [98].

3.6.2.1 LightGBM (Light Gradient Boosting Machine)

LightGBM is another highly efficient tree-based model, specifically structured to process large datasets quickly [99]. Traditional boosting algorithms are often computationally slow because they have to scan every single data point to calculate the best way to split a feature [99]. To overcome this bottleneck, LightGBM uses two novel techniques: Gradient-based One-Side Sampling (GOSS) and Exclusive Feature Bundling (EFB) [99]. GOSS speeds up training by temporarily ignoring data points the model has already learned well from those with small errors, forcing the algorithm to focus more heavily on the difficult, under-trained cases. Meanwhile, EFB bundles overlapping features together to reduce the overall complexity of the dataset without losing information [99].

In LightGBM, the information gain used to build the decision trees is measured by variance. The variance gain for splitting a feature j at point d over a dataset O is defined as:

$$V_j(d) = \frac{1}{n_O} \left(\frac{\left(\sum_{i \in O_l} g_i\right)^2}{n_{jl|O}} + \frac{\left(\sum_{i \in O_r} g_i\right)^2}{n_{jr|O}} \right) \tag{3.10}$$

In this formula, n_O is the total number of instances in the dataset, g_i represents the negative gradients, that is, the model's errors, and $n_{jl|O}$ and $n_{jr|O}$ represent the number of data points distributed to the left and right child nodes after the split occurs [99].

3.7 Evaluation metrics

The predictive capabilities of classification models need multiple assessment methods because a single performance evaluation metric cannot provide the complete assessment. A robust evaluation requires the interpretation of the outcomes of a confusion matrix which shows model prediction results in four separate outcomes of True Positives (TP) True Negatives (TN) False Positives (FP) and False Negatives (FN) . The initial values serve as the basis for deriving the following evaluation metrics which assess the complete performance of the models.

1. Accuracy: The ratio of accurately predicted occurrences with total instances is known as accuracy [100]. This metric works well when the class distribution is balanced but the result may be misleading when there is a presence of class imbalance.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{3.11}$$

2. Precision: Precision focuses entirely on the exactness of the positive predictions. It calculates the ratio of correctly identified diabetic patients out of all patients the model flagged as diabetic. In clinical terms, a high precision means fewer false alarms [101][100].

$$Precision = \frac{TP}{TP + FP} \tag{3.12}$$

3. Recall (Sensitivity): Recall, also known as Sensitivity, calculates the completeness of the model. Using recall, it is possible to calculate the rate of actual diabetic patients that the model successfully identified. In medical diagnoses, a high recall value is important because missing a positive case (a false negative) can have severe consequences for a patient's health [101][100].

$$Recall = \frac{TP}{TP + FN} \tag{3.13}$$

4. F1-Score: The F1-Score is the harmonic mean of Precision and Recall. Because standard accuracy struggles with imbalanced datasets, the F1-Score provides a much more realistic measure of a model's performance on the minority class by actively balancing the trade-off between false positives and false negatives [101][100].

$$F1\text{-}Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{3.14}$$

5. Matthews Correlation Coefficient (MCC): The Matthews Correlation Coefficient is widely considered one of the most reliable metrics for imbalanced datasets. The MCC calculation varies from the F1-score because it uses all four confusion matrix elements to produce a single correlation coefficient that ranges from -1 to +1. A high MCC score is only achieved if the model successfully predicts both the majority healthy class and the minority diabetic class [80][100].

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \tag{3.15}$$

6. ROC-AUC (Receiver Operating Characteristic - Area Under the Curve): The ROC curve is a two-dimensional visual graph that plots the True Positive Rate (Recall) on the Y-axis against the False Positive Rate on the X-axis across all possible classification thresholds. The Area Under the Curve (AUC) reduces this graph to a single scalar value between 0 and 1. An AUC score evaluates how well the model separates the diabetic cases from the healthy cases; a higher AUC indicates that a randomly chosen diabetic patient has a much higher probability of being ranked correctly than a randomly chosen healthy patient [81][100].

$$TPR = \frac{TP}{TP + FN} \quad \text{and} \quad FPR = \frac{FP}{FP + TN} \tag{3.16}$$

3.8 Explainability with SHAP

The ensemble ML models like XGBoost and LightGBM are robust models and these models can achieve a high predictive accuracy but their tree based architectures are so complex and difficult to understand that how the model is doing the prediction. Mostly these architectures act as Black Boxes. However, in healthcare predictions it is very important to find out how the model is actually getting the result. So, to ensure clinical transparency, our research includes SHAP framework. Lundberg and Lee [88] proposed this highly mathematical framework to provide a rigorous and transparent interpretation of ML predictions.

3.8.1 Theoretical Foundation: Shapley Values and Additive Attribution

SHAP is grounded in cooperative game theory, where clinical features (e.g., age, BMI) act as players and the final diagnostic prediction acts as the payout. In order to evaluate a model fairly, SHAP calculates the exact marginal contribution of each clinical feature across all possible feature combinations, known as the Shapley value (ϕ):

$$\phi_i = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|!(|N| - |S| - 1)!}{|N|!} [v(S \cup \{i\}) - v(S)] \tag{3.17}$$

Where N is the set of all features, S is a subset of features excluding feature i, and v(S) is the model's prediction using only subset S. SHAP simplifies this complex game theory into an additive feature attribution method. This means the final prediction model g(z') is explained simply as the base expected value (ϕ) plus a linear sum of the individual SHAP values:

$$g(z') = \phi_0 + \sum_{i=1}^{M} \phi_i z'_i \tag{3.18}$$

The total impact of each feature on the model prediction leads to a complete match with the difference between the final predicted probability of the model and the average baseline prediction of the dataset [88].

3.8.2 Algorithm-Specific Explanations: TreeSHAP and KernelSHAP

Calculating exact Shapley values mathematically requires exponential time complexity ($O(2^N)$), which is computationally impossible for medical datasets with lots of variables. So, to efficiently bypass this limitation, our research utilizes two specialized SHAP approximations tailored to the specific ML algorithms deployed:

TreeSHAP (For XGBoost & LightGBM): This highly efficient variant of SHAP pushes the Shapley calculations down the internal routing pathways of decision trees. TreeSHAP uses tree structures to find exact SHAP values through its polynomial time calculations which make it suitable for primary ensemble models according to Lundberg et al. [102].

KernelSHAP (For Gaussian Naïve Bayes): Because TreeSHAP cannot interpret probabilistic models, KernelSHAP is utilized as a model-agnostic fallback. KernelSHAP approximates feature importance by building a locally weighted linear regression model. It generates a background dataset by toggling features on and off, assigning a weight ($\pi_{x'}(z')$) to each sample based on the number of features present:

$$\pi_{x'}(z') = \frac{(M-1)}{\binom{M}{|z'|}|z'|(M-|z'|)} \tag{3.19}$$

By minimizing the loss function of this weighted regression, KernelSHAP isolates the exact diagnostic contribution of each independent variable [88].

3.8.3 Application in this Research

By dynamically applying TreeSHAP for the tree-based ensemble models and KernelSHAP for the Naïve Bayes classifier, this study extracts consistent and comparable feature attributions across fundamentally different algorithmic architectures. The Naïve Bayes model evaluation process used background data samples to achieve efficient computation during KernelSHAP assessment. The XAI pipeline converts complex model probabilities into transparent clinical insights which show the most important risk factors that lead to diabetes predictions across three different datasets.

To bridge the gap between individual patient-level predictions and the global behavioural patterns of each model, this study employs SHAP beeswarm plots as the primary visualisation tool for interpreting predictive results in the subsequent chapters. While standard bar charts convey only the average magnitude of feature importance, effectively hiding distributional and directional information, the beeswarm plot resolves this limitation by projecting the high-dimensional SHAP impacts of all clinical features into a dense, two-dimensional visual summary [102].

Each dot in the plot represents a single patient instance, and the plot is interpreted across four simultaneous visual dimensions [102][103]:

- **Y-Axis (Feature Ranking):** In this axis, the features of the dataset are shown in descending order of their mean absolute SHAP value. That means, the strongest contributor for the models prediction appears at the top of the axis.
- **X-Axis (SHAP Value/Directional Impact):** The horizontal axis holds the magnitude and direction of each feature's contribution. A SHAP value of 0.0 indicates zero influence on the prediction, values to the right (positive) pushes the model toward a diabetic outcome and values to the left (negative) pushes it toward a non-diabetic outcome.
- **Colour (Feature Value Magnitude):** The original clinical value of each feature which was assessed for every patient is represented through the dot colour system. Deep red shows a high feature value which includes elevated glucose measurements while blue shows a low feature value which includes low BMI measurements. This enables immediate assessment of how clinical measurements relate to predicted diabetic risk.
- **Vertical Spread (Data Density):** When many patients share an identical SHAP value, their dots stack vertically, creating a dense swarm of points. The thickness of this cluster indicates the proportion of the cohort exhibiting that particular effect, which provides distributional information that a simple mean would obscure.

The process of assessing these four dimensions together will lead to the discovery of non-linear clinical thresholds. For instance, if a long tail of red dots stretches far to the right for a feature such as BMI, it visually demonstrates to clinicians that high BMI values are a strong driver of positive diabetes predictions [102]. The beeswarm plot displays multiple data elements through its ability to show importance and direction and feature value and patient distribution, which makes it the most information-rich visual display in the SHAP framework and the best research tool for clinical interpretability.

3.9 Experimental setups & Implementation details

To examine how much pre-split resampling inflates model performance, and to understand how feature importance changes under a biased and an unbiased pipeline, we designed a set of parallel experiments for all three datasets: PIMA Indians Diabetes Dataset, BRFSS-2015, and Diahealth. The main idea was to keep the model settings and evaluation process the same, while changing the position of the resampling step so that a fair comparison could be made between the pre-split and post-split pipelines.

For each dataset, the experiments followed a common implementation structure. First, the dataset was loaded and cleaned. Then dataset-specific preprocessing was applied, such as zero-value handling for PIMA indian dataset, optional outlier removal, and optional feature selection. The data was divided into training and test sets using an 80:20 split. In the post-split pipeline, resampling was applied only to the training set. In the pre-split pipeline, resampling was applied before the train-test split. The same classifiers, evaluation metrics, and SHAP-based interpretation process were then used across all matched experiment settings. This design helped ensure that any major difference in performance or feature importance came mainly from the placement of resampling, not from unrelated changes in the model setup.

The full implementation was carried out through multiple matched configurations. For PIMA indian dataset, both imputation and elimination versions were tested, each again divided into datasets with outlier removal/without outlier removal and with feature selection/without feature selection. For BRFSS-2015 and Diahealth, the experiments were run in four matched settings based on outlier removal and feature selection. In every case, both the pre-split and post-split versions were created so that the results could be compared directly.

3.9.1 Hardware and Software Environment

The experiments were conducted in Python 3 using Google Colab as the execution environment. The software libraries that used in the experiment were pandas and NumPy for data handling, scikit-learn for preprocessing and evaluation, imbalanced-learn for resampling, XGBoost and LightGBM for classification, SHAP for explainability analysis, and Matplotlib and Seaborn for visualization.

3.9.2 Hyperparameter Configuration

To ensure equal testing conditions throughout the experiment, fixed hyperparameters were used. XGBoost was run with n_estimators = 100, random_state = 42, and eval_metric = ‘logloss’. LightGBM was run with objective = ‘binary’, n_estimators = 100, metric = ‘binary_logloss’, random_state = 42, and verbose = -1. Gaussian Naïve Bayes was used with default settings. The resampling methods SMOTE, ADASYN, SMOTEENN, and SMOTETomek were all used with random_state = 42. In the PIMA indian dataset

imputation experiments, IterativeImputer with max_iter = 10 and random_state = 42 was used. Feature selection was based on mutual_info_classif with random_state = 42.

Table 3.11: Summary of experimental components and their configurations

Component	**Configuration Used**
Train-test split	80:20
Split seed	random_state = 0
XGBoost	n_estimators = 100, random_state = 42, eval_metric = 'logloss', use_label_encoder = False
LightGBM	objective = 'binary', n_estimators = 100, metric = 'binary_logloss', random_state = 42, verbose = -1
Gaussian Naive Bayes	Default settings
SMOTE	random_state = 42
ADASYN	random_state = 42
SMOTEENN	random_state = 42
SMOTETomek	random_state = 42
PIMA Indian imputation	IterativeImputer, max_iter = 10, random_state = 42
Feature selection	mutual_info_classif, random_state = 42
Scaling	StandardScaler applied to selected numerical features

3.9.3 Execution Protocol

The experiments used datasets that were divided into 80 percent training data and 20 percent test data. The saved implementation created the split through random_state = 0, which generated the first valid split. The system included a validity check to confirm that the test set contained all necessary target classes. In the final saved runs, the split was successfully accepted at the first attempt, so the effective split seed remained 0.

After the split, the two pipeline variants were handled differently. In the post-split pipeline, the dataset was first divided into training and test sets, and then the selected resampling technique was applied only to the training set. This was the corrected and methodologically proper workflow, because it kept the test set independent from synthetic samples. In the pre-split pipeline, the selected resampling method was applied to the full dataset first, and only then was the data split into training and test sets. This setup was intentionally used to measure how much data leakage could inflate model performance and affect SHAP-based feature importance.

To make the comparison fair, the same split ratio, model settings, resampling choices, and evaluation metrics were used in both pipelines. After training, each model was evaluated using Accuracy, Precision, Recall, F1-score, AUC-ROC, and MCC. Then, SHAP analysis was applied to the best-performing tree-based model in each setting so that the feature importance patterns could be compared between the biased pre-split pipeline and the corrected post-split pipeline.

4 Results and Analysis

4.1 Pre-split Prediction Result

The performance of each model, Naive Bayes, XGBoost, and LightGBM, acquired by using resampling strategies, such as SMOTE, SMOTEENN, SMOTETomek, and ADASYN, prior to splitting the dataset into training and testing sets to solve class imbalance, is referred to as pre-split prediction results in this segment. This segment aims to exhibit the findings to answer our research questions like identifying if it involves any synthetic data leakage into the test set, resulting in overly optimistic performance estimations [41][42].

4.1.1 PIMA Indian Diabetes Dataset (PIDD) Prediction Result

This section reveals the results attained from applying resampling methods before the train and test split of the PIMA indian dataset in eight different circumstances.

Imputation & Outlier removed without feature selection

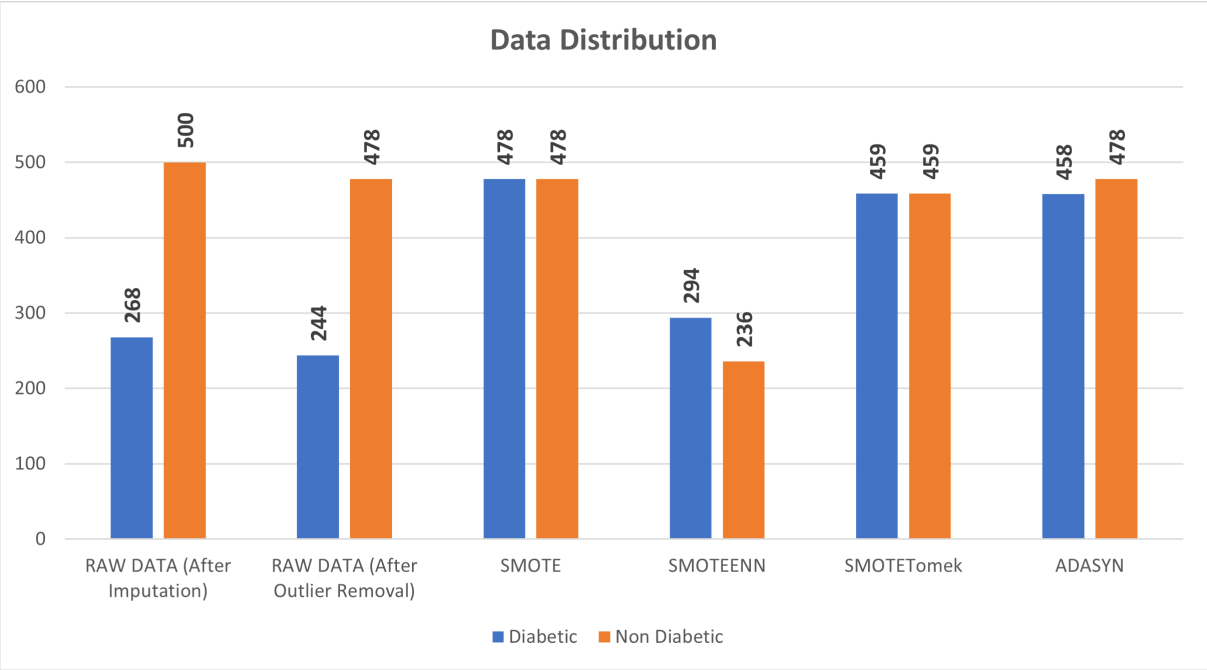
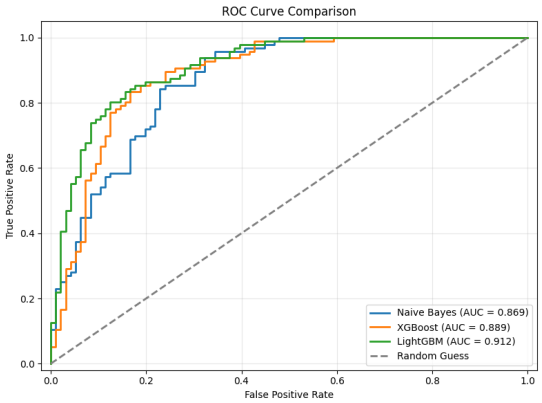


Figure 4.1: Data Distribution for Imputation & Outlier removed without feature selection

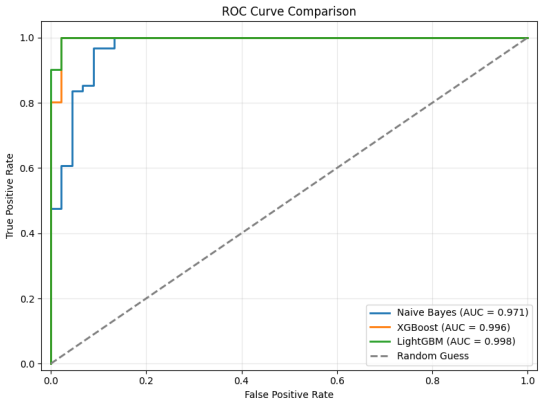
Table 4.1: Performance comparison of models after imputation, outlier removal, and without feature selection in the PIMA indian dataset

Resampling Technique	ML Model	Accuracy	Precision	Recall	F1-Score	AUC-ROC	MCC
SMOTE	Naive Bayes	0.7865	0.7778	0.8021	0.7897	0.8689	0.5732
	XGBoost	0.8073	0.7565	0.9062	0.8246	0.8891	0.6270
	LightGBM	0.8073	0.7757	0.8646	0.8177	0.9122	0.6187
SMOTEENN	Naive Bayes	0.9434	0.9104	**1.0000**	0.9531	0.9712	0.8883
	XGBoost	**0.9906**	**0.9839**	**1.0000**	**0.9919**	0.9956	**0.9808**
	LightGBM	**0.9906**	**0.9839**	**1.0000**	**0.9919**	**0.9978**	**0.9808**
SMOTETomek	Naive Bayes	0.7174	0.7073	0.6744	0.6905	0.8162	0.4312
	XGBoost	0.7880	0.7474	0.8256	0.7845	0.8741	0.5798
	LightGBM	0.8261	0.7812	0.8721	0.8242	0.8900	0.6570
ADASYN	Naive Bayes	0.7074	0.7159	0.6774	0.6961	0.7909	0.4151
	XGBoost	0.8191	0.7921	0.8602	0.8247	0.8851	0.6409
	LightGBM	0.8191	0.7864	0.8710	0.8265	0.8883	0.6423

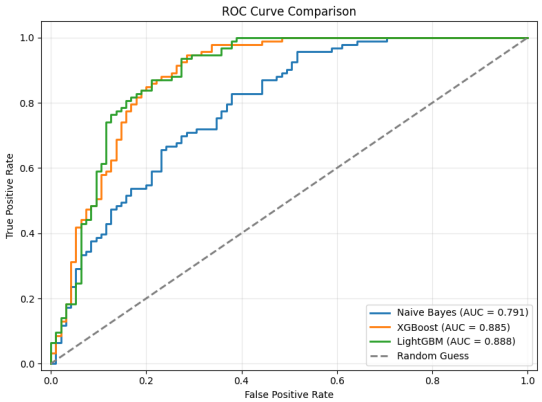
Table 4.1 demonstrates that the highest performance attained without feature selection after imputation and removing the outliers is by the XGBoost and LightGBM models after applying pre-split SMOTEENN to balance the data. In contrast, Naive Bayes showed the lowest results under SMOTETomek and ADASYN resampling.



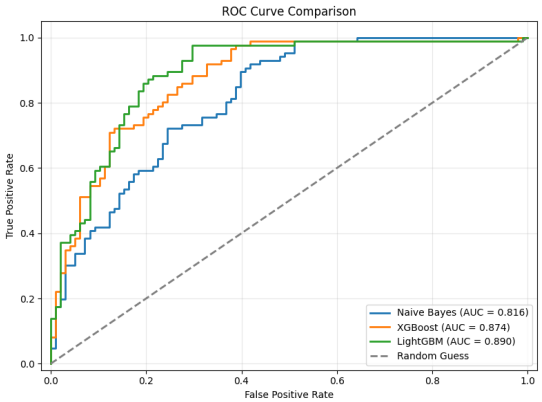
(a) SMOTE



(b) SMOTEENN



(c) ADASYN



(d) SMOTETomek

Figure 4.2: ROC curves demonstrating model performance on the PIMA indian dataset for pre-split data resampling Imputation and Outlier removed without feature selection.

Elimination & Outlier removed without feature selection

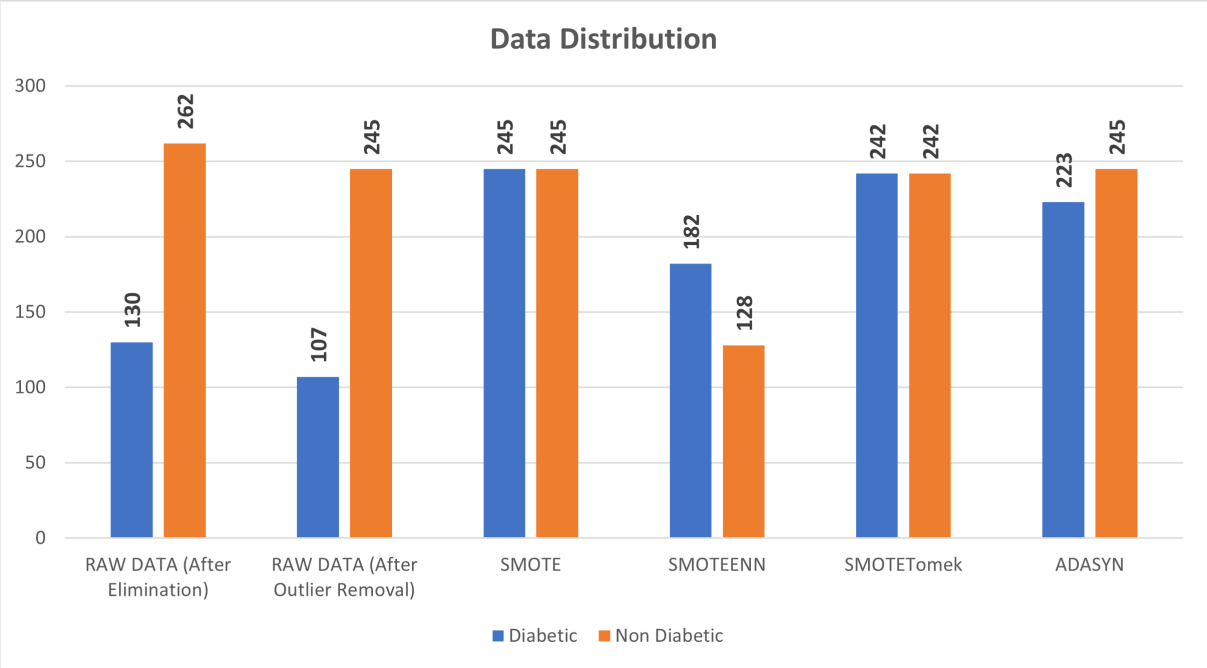
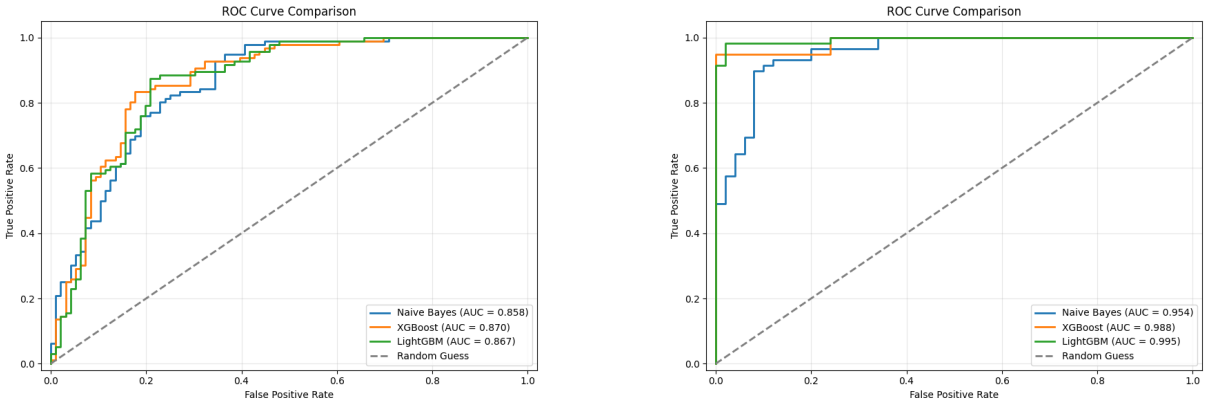


Figure 4.3: Data Distribution for Elimination & Outlier removed without feature selection

Table 4.2: Performance comparison of models after elimination, outlier removal, and without feature selection in the PIMA indian dataset

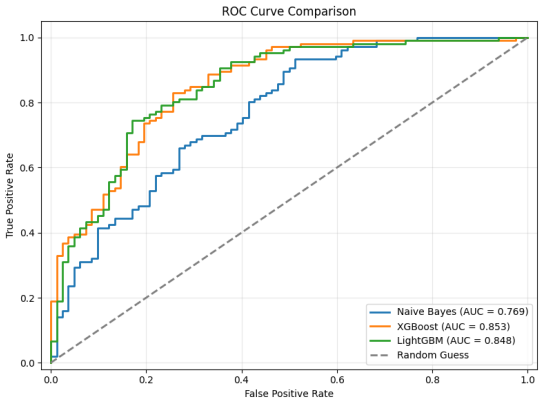
Resampling Technique	ML Model	Accuracy	Precision	Recall	F1-Score	AUC-ROC	MCC
SMOTE	Naive Bayes	0.8163	0.7959	0.8298	0.8125	0.8861	0.6332
	XGBoost	0.8776	0.8571	0.8936	0.8750	0.9157	0.7557
	LightGBM	0.8673	0.8400	0.8936	0.8660	0.9166	0.7363
SMOTEENN	Naive Bayes	0.9194	0.8947	0.9714	0.9315	0.9704	0.8380
	XGBoost	**0.9516**	**0.9211**	**1.0000**	**0.9589**	**0.9799**	**0.9048**
	LightGBM	**0.9516**	**0.9211**	**1.0000**	**0.9589**	0.9778	**0.9048**
SMOTETomek	Naive Bayes	0.7732	0.8511	0.7273	0.7843	0.8991	0.5558
	XGBoost	0.8144	0.8136	0.8727	0.8421	0.9039	0.6200
	LightGBM	0.7938	0.8070	0.8364	0.8214	0.9061	0.5782
ADASYN	Naive Bayes	0.7872	0.8039	0.8039	0.8039	0.8003	0.5714
	XGBoost	0.8617	0.8800	0.8627	0.8713	0.9138	0.7221
	LightGBM	0.8298	0.8302	0.8627	0.8462	0.8906	0.6564

Table 4.2 demonstrates that the highest performance attained without feature selection after elimination and removing the outliers is by the XGBoost and LightGBM models after applying pre-split SMOTEENN to balance the data. In contrast, Naive Bayes showed the lowest results under SMOTE, SMOTETomek and ADASYN resampling alongside the least MCC value.

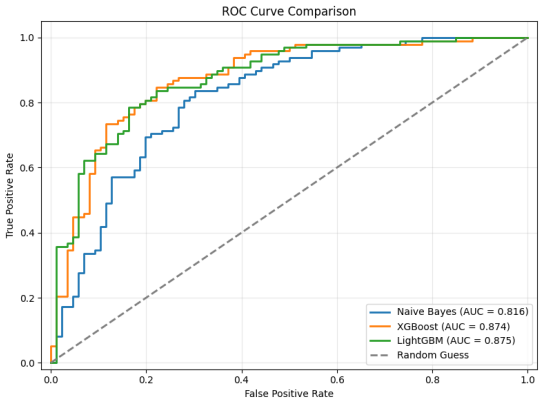


(a) SMOTE

(b) SMOTEENN



(c) ADASYN



(d) SMOTETomek

Figure 4.4: ROC curves demonstrating model performance on the PIMA indian dataset for pre-split data resampling Elimination and Outlier removed without feature selection.

Imputation & Outlier removed with feature selection

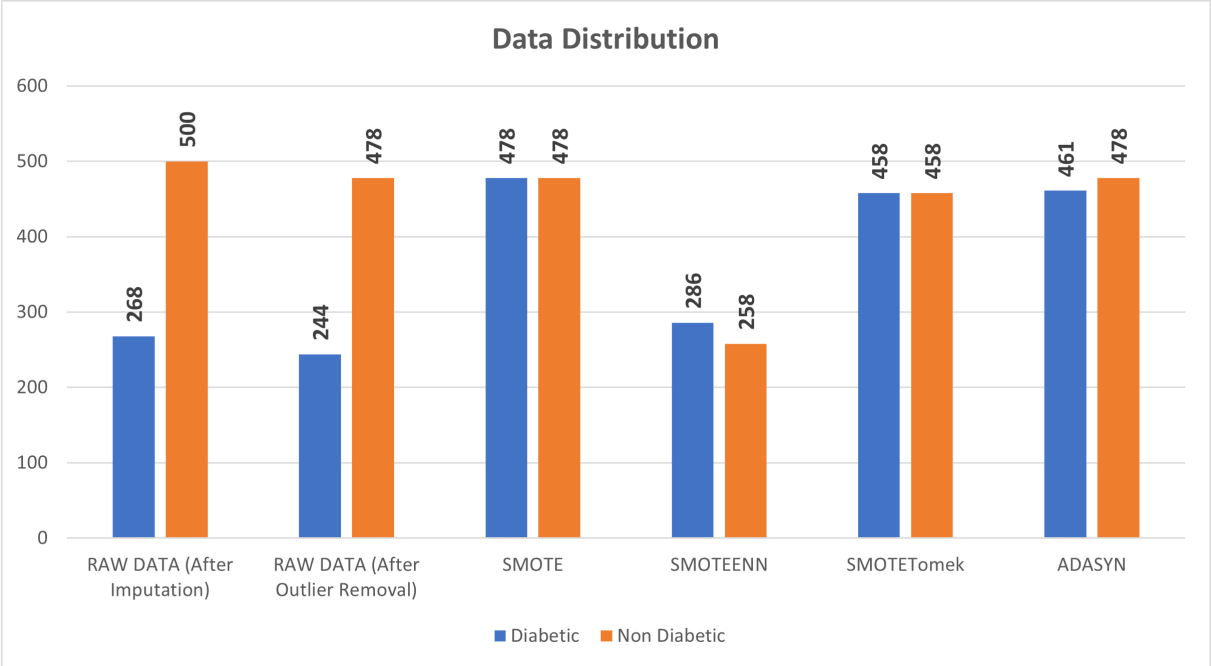
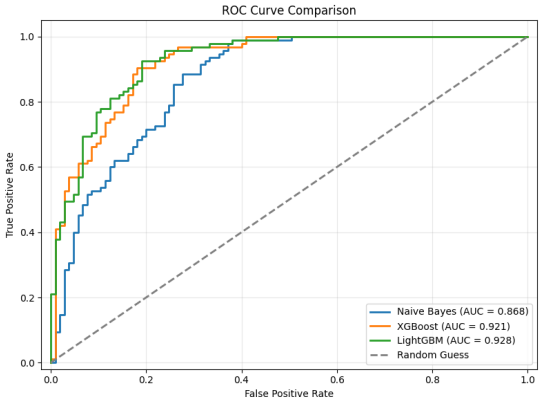


Figure 4.5: Data Distribution for Imputation & Outlier removed with feature selection

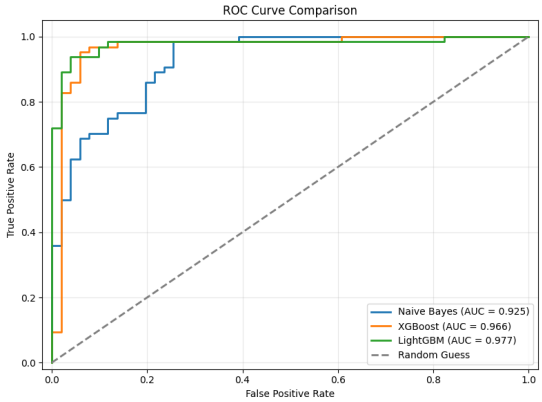
Table 4.3: Performance comparison of models after imputation, outlier removal, and with feature selection in the PIMA indian dataset

Resampling Technique	ML Model	Accuracy	Precision	Recall	F1-Score	AUC-ROC	MCC
SMOTE	Naive Bayes	0.7812	0.7755	0.7917	0.7835	0.8581	0.5626
	XGBoost	0.8125	0.7885	0.8542	0.8200	0.8702	0.6272
	LightGBM	0.8281	0.8000	0.8750	0.8358	0.8666	0.6592
SMOTEENN	Naive Bayes	0.8991	0.9138	0.8983	0.9060	0.9542	0.7972
	XGBoost	**0.9633**	**0.9825**	**0.9492**	**0.9655**	0.9878	**0.9270**
	LightGBM	0.9270	0.9270	0.9270	**0.9655**	**0.9946**	**0.9270**
SMOTETomek	Naive Bayes	0.7337	0.7579	0.7347	0.7461	0.8156	0.4665
	XGBoost	0.8098	0.8058	0.8469	0.8259	0.8745	0.6175
	LightGBM	0.7880	0.7757	0.8469	0.8098	0.8749	0.5743
ADASYN	Naive Bayes	0.6755	0.7528	0.6321	0.6872	0.7693	0.3613
	XGBoost	0.7606	0.8144	0.7453	0.7783	0.8525	0.5217
	LightGBM	0.7713	0.8119	0.7736	0.7923	0.8483	0.5389

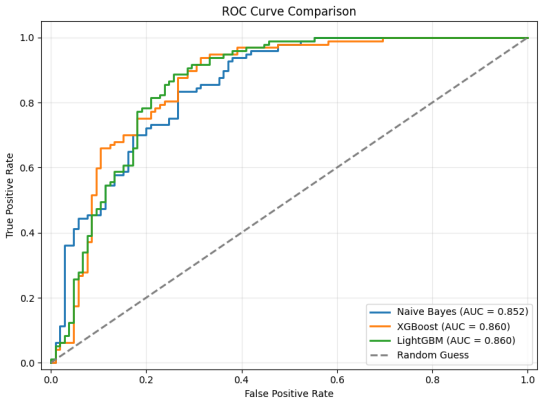
Table 4.3. signifies that XGBoost and LightGBM have obtained the highest result with feature selection after imputation and removing the outliers via pre-split SMOTEENN implementation. Conversely, Naive Bayes showed the lowest results under ADASYN resampling even after feature selection.



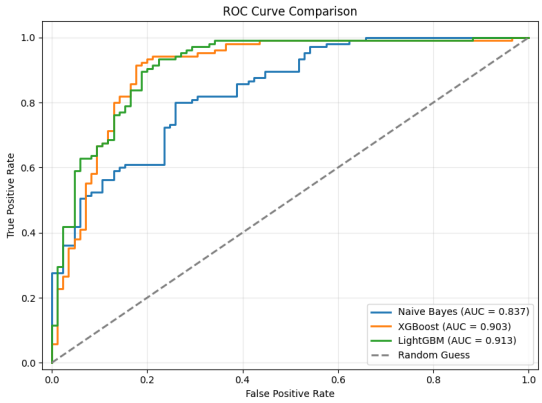
(a) SMOTE



(b) SMOTEENN



(c) ADASYN



(d) SMOTETomek

Figure 4.6: ROC curves demonstrating model performance on the PIMA indian dataset for pre-split data resampling Imputation and Outlier removed with feature selection.

Elimination & Outlier removed with feature selection

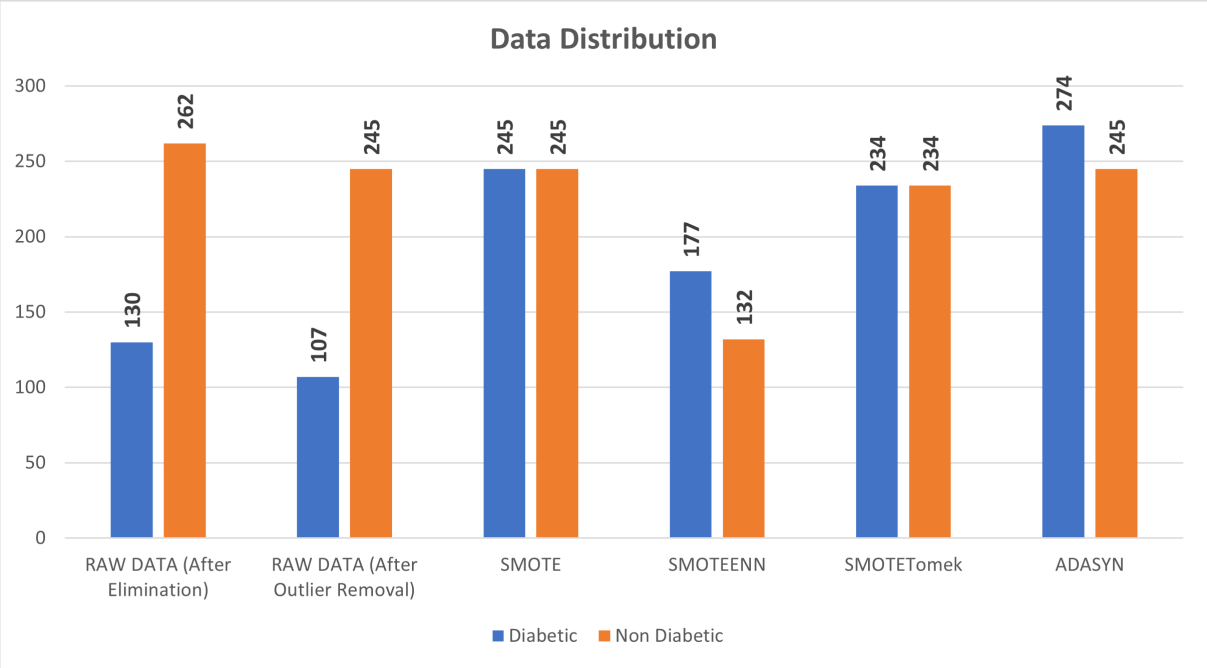
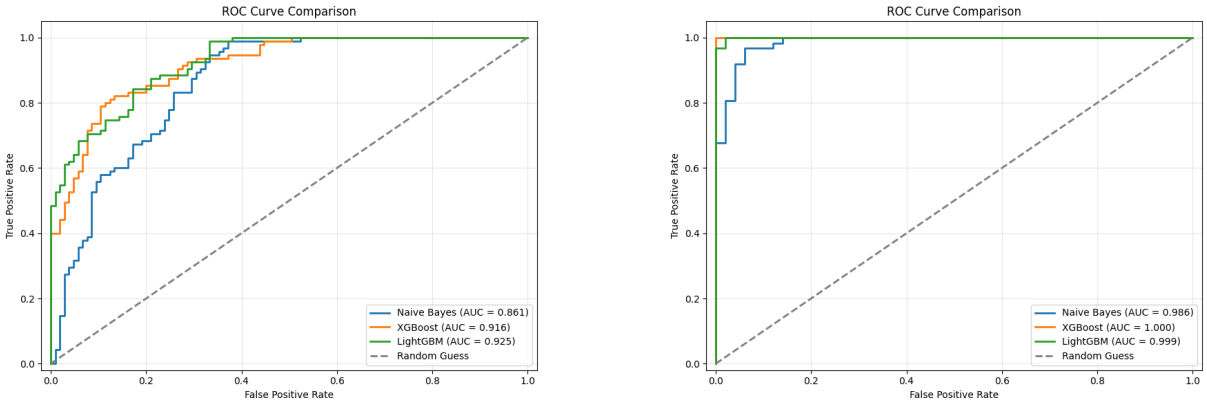


Figure 4.7: Data Distribution for Elimination & Outlier removed with feature selection

Table 4.4: Performance comparison of models after elimination, outlier removal, and with feature selection in the PIMA indian dataset

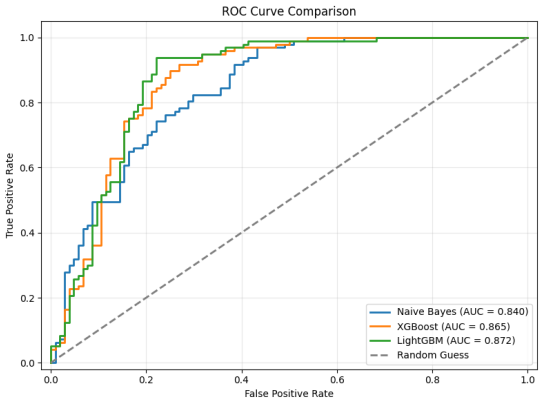
Resampling Technique	ML Model	Accuracy	Precision	Recall	F1-Score	AUC-ROC	MCC
SMOTE	Naive Bayes	0.8265	0.8125	0.8298	0.8211	0.8778	0.6529
	XGBoost	0.8367	0.8298	0.8298	0.8298	0.8865	0.6729
	LightGBM	0.8265	0.8000	0.8511	0.8247	0.8861	0.6546
SMOTEENN	Naive Bayes	0.9355	0.9231	0.9730	0.9474	0.9632	0.8662
	XGBoost	**0.9677**	**0.9487**	**1.0000**	**0.9737**	0.9686	**0.9342**
	LightGBM	0.9516	0.9474	0.9730	0.9600	**0.9859**	0.8993
SMOTETomek	Naive Bayes	0.8404	0.8444	0.8261	0.8352	0.8687	0.6807
	XGBoost	0.8617	0.8367	0.8913	0.8632	0.9334	0.7251
	LightGBM	0.8617	0.8235	0.9130	0.8660	0.9443	0.7280
ADASYN	Naive Bayes	0.7596	0.7727	0.6939	0.7312	0.8564	0.5174
	XGBoost	0.8750	0.8462	0.8980	0.8713	0.9098	0.7513
	LightGBM	0.8846	0.8491	0.9184	0.8824	0.9184	0.7718

Table 4.4 reveals that XGBoost and LightGBM have obtained the highest result with feature selection after elimination and removing the outliers via pre-split SMOTEENN implementation. Conversely, Naive Bayes showed the lowest results under ADASYN resampling even after feature selection.

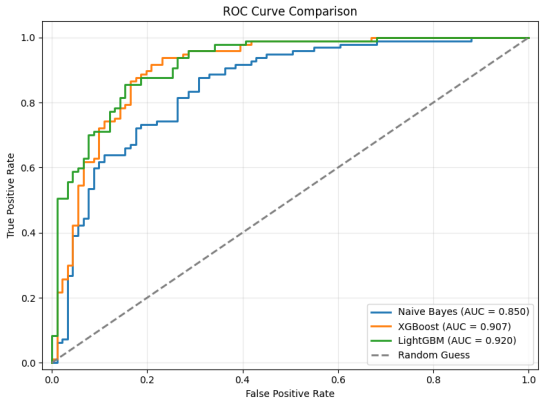


(a) SMOTE

(b) SMOTEENN



(c) ADASYN



(d) SMOTETomek

Figure 4.8: ROC curves demonstrating model performance on the PIMA indian dataset for pre-split data resampling Elimination and Outlier removed with feature selection.

Imputation With outlier without feature selection

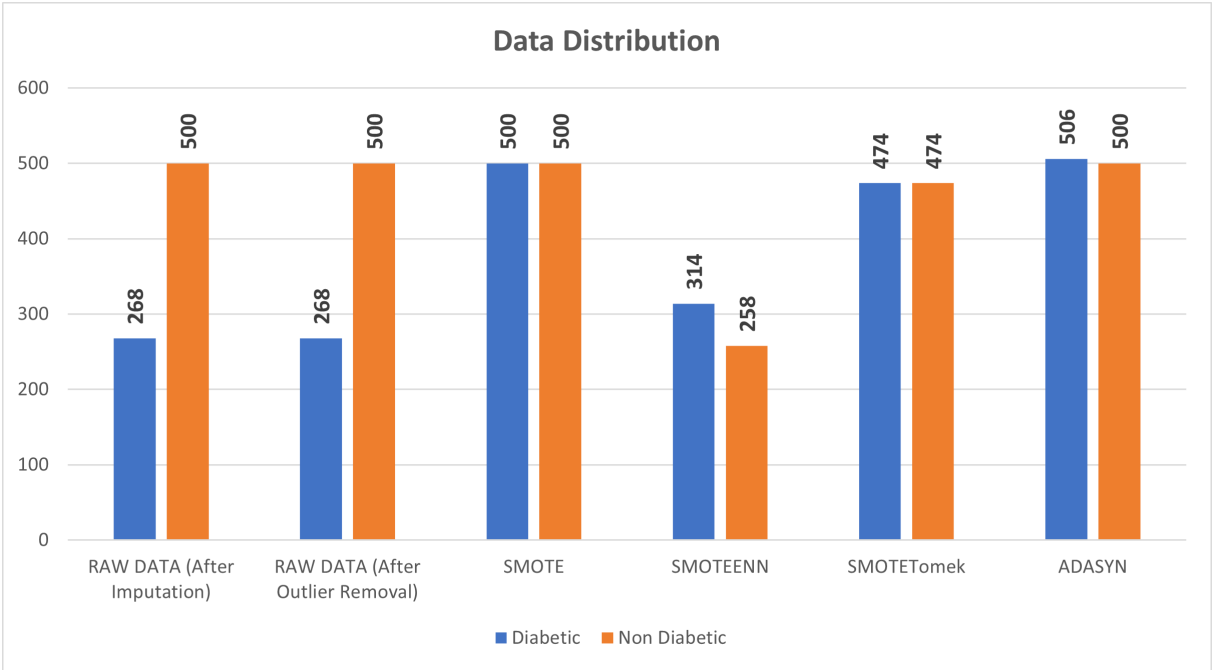
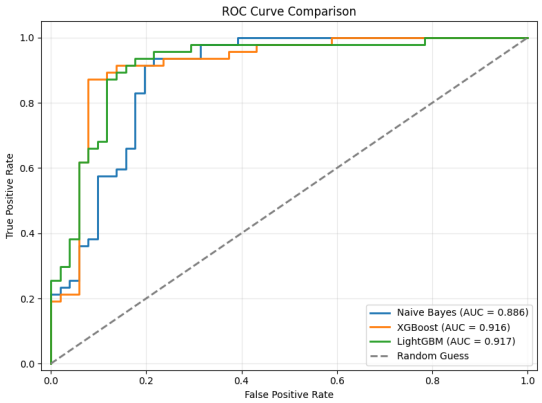


Figure 4.9: Data Distribution for Imputation With outlier without feature selection

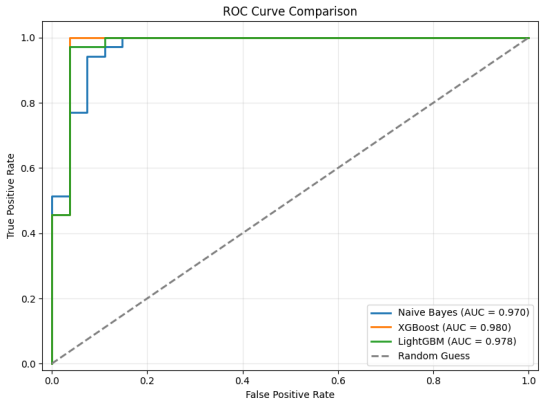
Table 4.5: Performance comparison of models after imputation, with outliers, and without feature selection in the PIMA indian dataset

Resampling Technique	ML Model	Accuracy	Precision	Recall	F1-Score	AUC-ROC	MCC
SMOTE	Naive Bayes	0.7450	0.7683	0.6632	0.7119	0.8679	0.4896
	XGBoost	0.8500	0.8155	0.8842	0.8485	0.9214	0.7027
	LightGBM	0.8500	0.8095	0.8947	0.8500	0.7043	0.7043
SMOTEENN	Naive Bayes	0.8348	0.8261	0.8906	0.8571	0.9249	0.6646
	XGBoost	0.9304	**0.9118**	0.9688	0.9394	0.9660	0.8601
	LightGBM	**0.9391**	0.9130	**0.9844**	**0.9474**	**0.9770**	**0.8789**
SMOTETomek	Naive Bayes	0.7263	0.7849	0.6952	0.7374	0.8370	0.4575
	XGBoost	0.8579	0.8250	0.9429	0.8800	0.9027	0.7172
	LightGBM	0.8579	0.8362	0.9238	0.8778	0.9128	0.7141
ADASYN	Naive Bayes	0.7426	0.7778	0.6495	0.7079	0.8520	0.4873
	XGBoost	0.7921	0.7391	0.8763	0.8019	0.8601	0.5959
	LightGBM	0.8119	0.7611	0.8866	0.8190	0.8597	0.6334

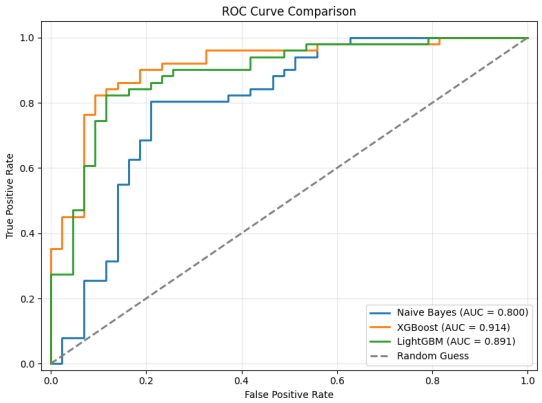
Table 4.5 indicates that XGBoost and LightGBM have achieved the highest results without feature selection, after imputation, across different evaluation metrics under pre-split SMOTEENN application. On the other hand, XGBoost gives the least precision value in addition to Naive Bayes showing the lowest recall and F1-score with ADASYN resampling, Naive Bayes also shows minimum accuracy and MCC score under SMOTETomek. Surprisingly, LightGBM showed the least AUC-ROC value after the SMOTE application.



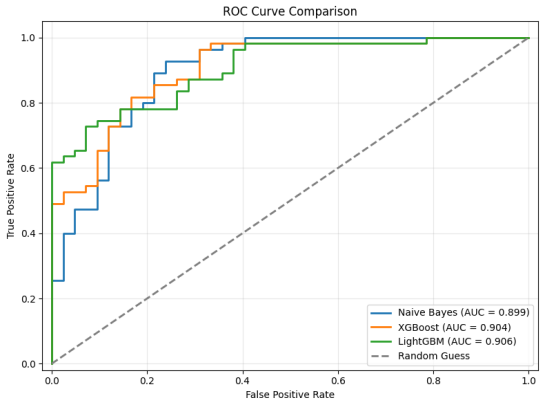
(a) SMOTE



(b) SMOTEENN



(c) ADASYN



(d) SMOTETomek

Figure 4.10: ROC curves demonstrating model performance on the PIMA indian dataset for pre-split data resampling Imputation With outlier without feature selection.

Elimination With outlier without feature selection

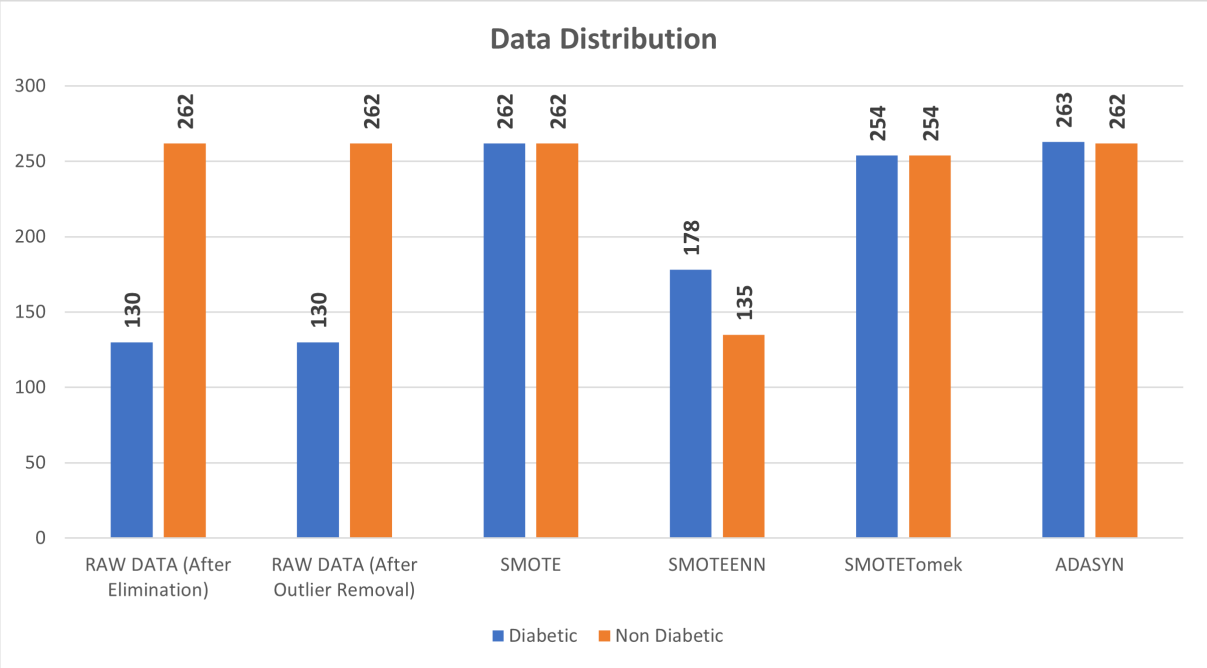
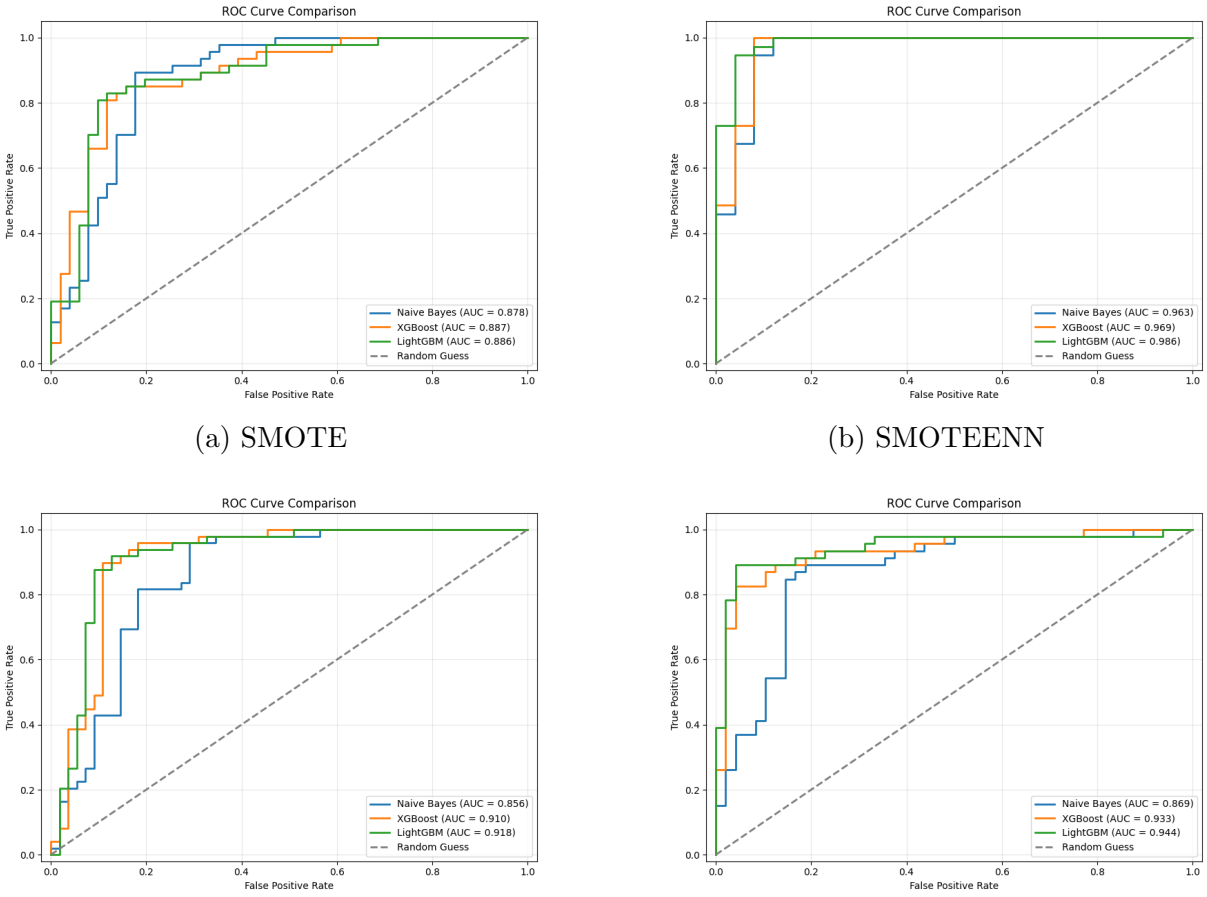


Figure 4.11: Data Distribution for Elimination With outlier without feature selection

Table 4.6: Performance comparison of models after elimination, with outliers, and without feature selection in the PIMA indian dataset

Resampling Technique	ML Model	Accuracy	Precision	Recall	F1-Score	AUC-ROC	MCC
SMOTE	Naive Bayes	0.6762	0.6744	0.5918	0.6304	0.7784	0.3468
	XGBoost	0.7905	0.7547	0.8163	0.7843	0.8812	0.5829
	LightGBM	0.7905	0.7647	0.7959	0.7800	0.8684	0.5806
SMOTEENN	Naive Bayes	0.9048	0.9167	0.9167	0.9167	0.9650	0.8056
	XGBoost	**0.9841**	**0.9730**	**1.0000**	**0.9863**	0.9990	**0.9680**
	LightGBM	**0.9841**	**0.9730**	**1.0000**	**0.9863**	**1.0000**	**0.9680**
SMOTETomek	Naive Bayes	0.6667	0.7111	0.6038	0.6531	0.7740	0.3406
	XGBoost	0.8333	0.7903	0.9245	0.8522	0.8802	0.6746
	LightGBM	0.8235	0.7612	0.9623	0.8500	0.8706	0.6690
ADASYN	Naive Bayes	0.6762	0.6909	0.6909	0.6909	0.7847	0.3509
	XGBoost	0.7905	0.7538	0.8909	0.8167	0.8895	0.5871
	LightGBM	0.8095	0.7869	0.8727	0.8276	0.8844	0.6202

Table 4.6 indicates that XGBoost and LightGBM have achieved the highest results without feature selection, after elimination, across different evaluation metrics under pre-split SMOTEENN application. On the other hand, Naive Bayes showed the lowest results after SMOTE and SMOTETomek resampling.



(a) SMOTE

(b) SMOTEENN

(c) ADASYN

(d) SMOTETomek

Figure 4.12: ROC curves demonstrating model performance on the PIMA indian dataset for pre-split data resampling Elimination With outlier without feature selection.

Imputation With outlier with feature selection

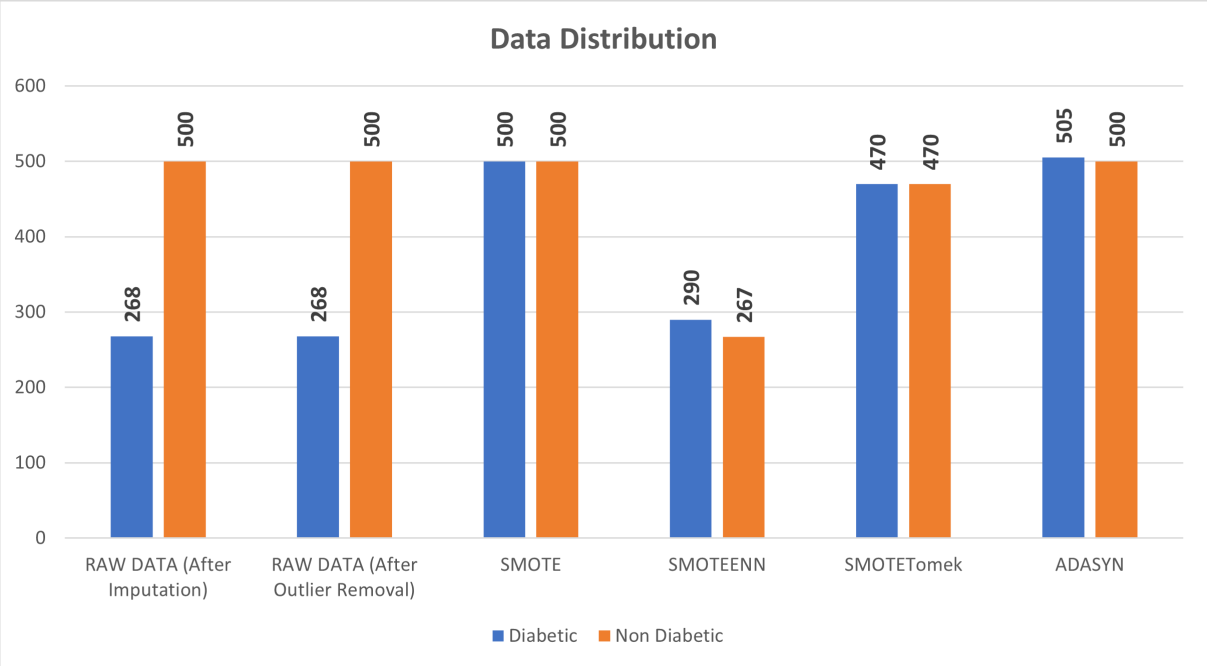
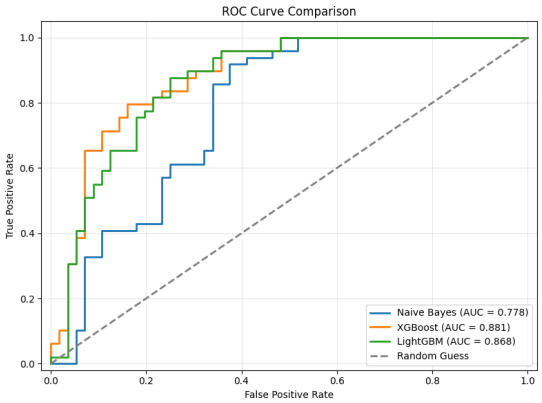


Figure 4.13: Data Distribution for Imputation With outlier with feature selection

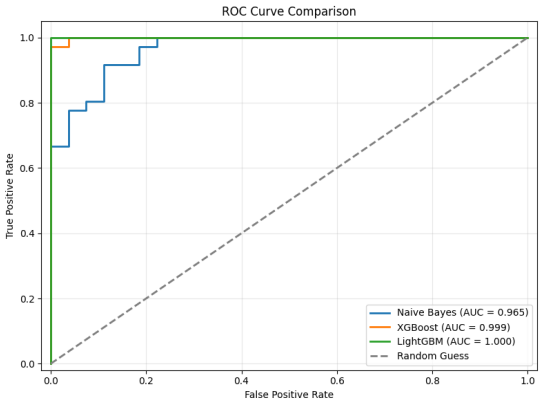
Table 4.7: Performance comparison of models after imputation, with outliers, and with feature selection in the PIMA indian dataset

Resampling Technique	ML Model	Accuracy	Precision	Recall	F1-Score	AUC-ROC	MCC
SMOTE	Naive Bayes	0.7500	0.7778	0.6632	0.7159	0.8611	0.5002
	XGBoost	0.8250	0.7941	0.8526	0.8223	0.9165	0.6519
	LightGBM	0.8350	0.8163	0.8421	0.8290	0.9253	0.6700
SMOTEENN	Naive Bayes	0.9375	0.9508	0.9355	0.9431	0.9858	0.8739
	XGBoost	**0.9732**	**1.0000**	0.9516	0.9752	**1.0000**	**0.9475**
	LightGBM	**0.9732**	0.9836	**0.9677**	**0.9756**	0.9994	0.9461
SMOTETomek	Naive Bayes	0.7553	0.7742	0.7423	0.7579	0.8501	0.5113
	XGBoost	0.8457	0.8091	0.9175	0.8599	0.9069	0.6966
	LightGBM	0.8245	0.8019	0.8763	0.8374	0.9197	0.6505
ADASYN	Naive Bayes	0.7413	0.7778	0.6495	0.7079	0.8403	0.4853
	XGBoost	0.8060	0.7685	0.8557	0.8098	0.8655	0.6166
	LightGBM	0.8358	0.7909	0.8969	0.8406	0.8718	0.6784

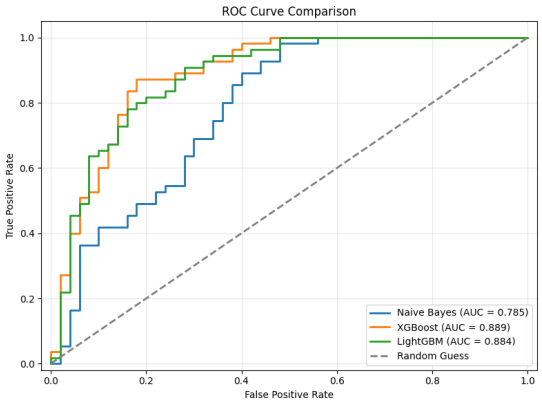
Table 4.7 demonstrate that the highest outcomes after imputation and feature selection across different evaluation metrics under pre-split SMOTEENN application was by XGBoost and LightGBM. However, Naive Bayes showed the lowest results with ADASYN resampling here as usual but this time XGBoost also gave the lowest precision under ADASYN.



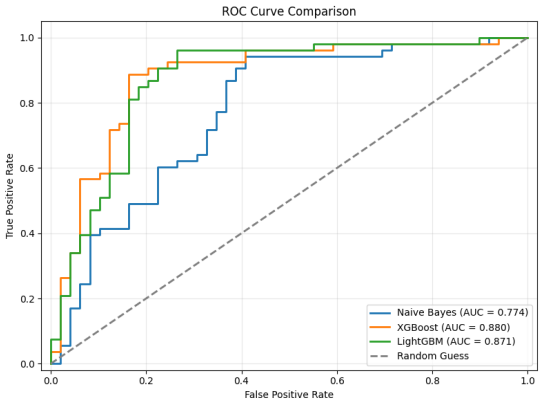
(a) SMOTE



(b) SMOTEENN



(c) ADASYN



(d) SMOTETomek

Figure 4.14: ROC curves demonstrating model performance on the PIMA indian dataset for pre-split data resampling Imputation With outlier with feature selection.

Elimination With outlier with feature selection

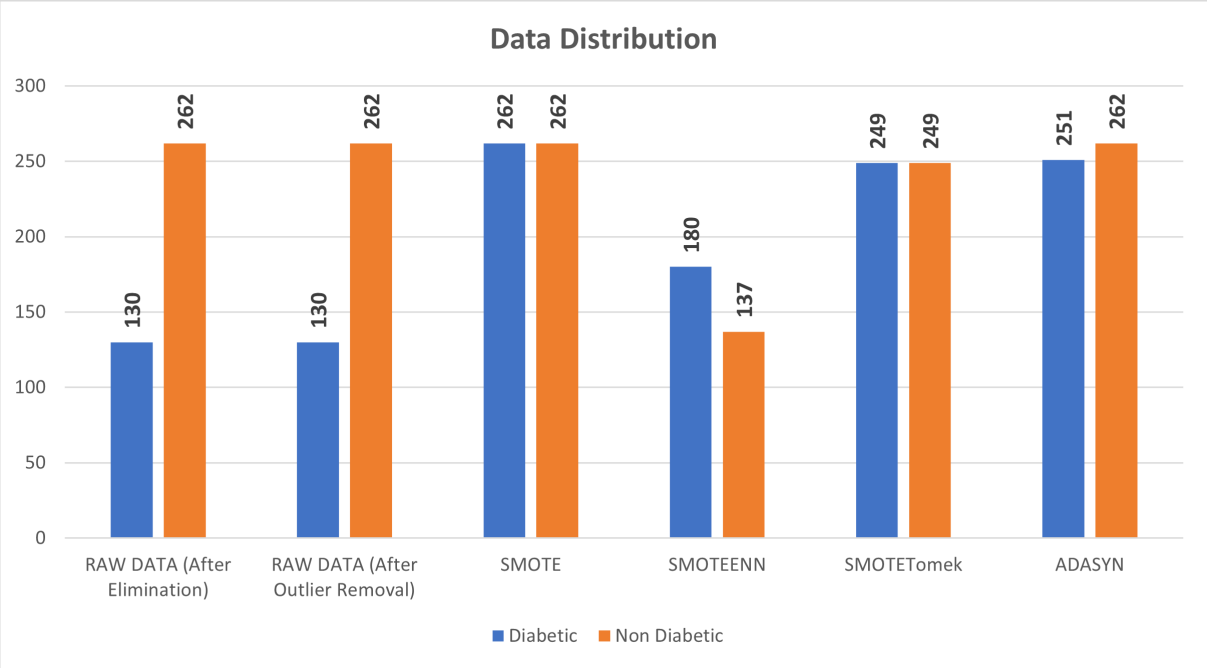
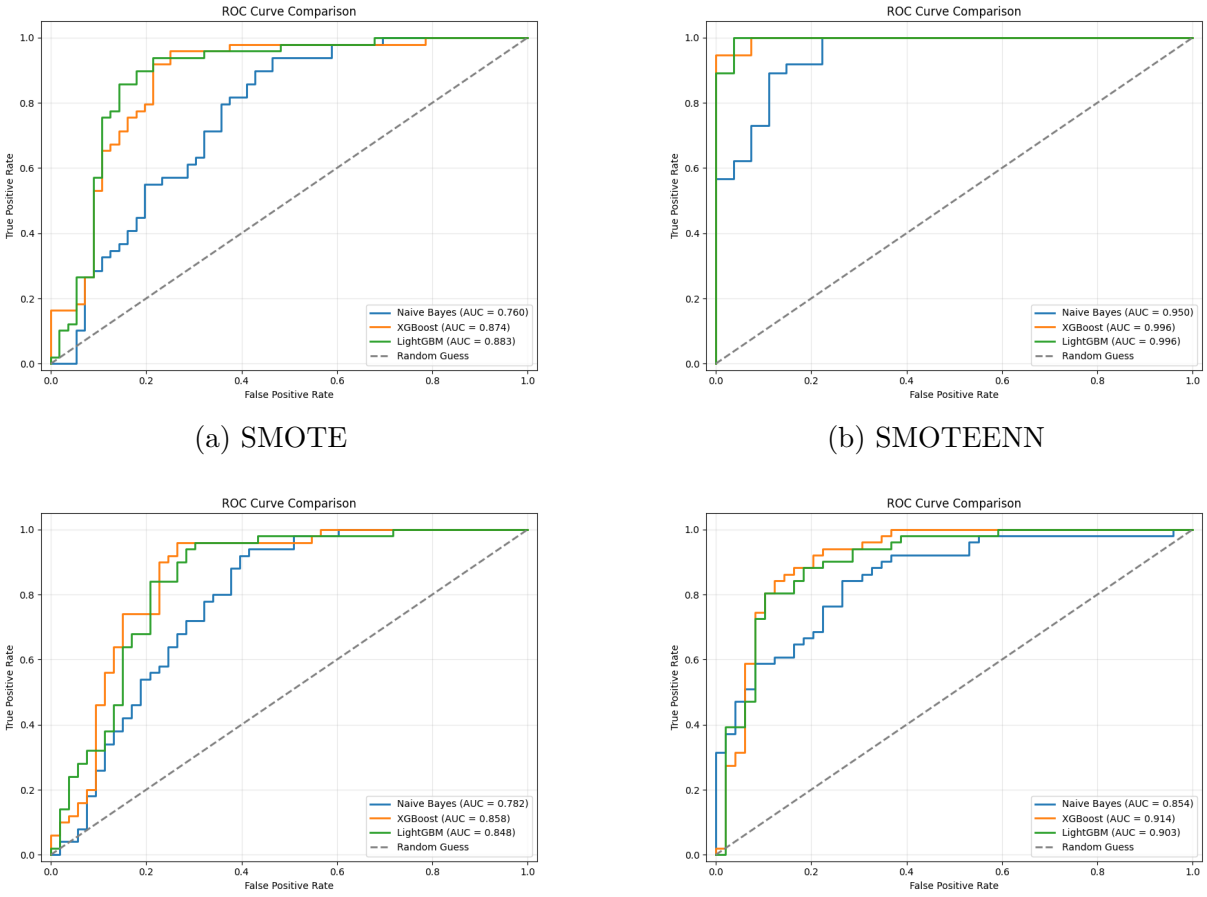


Figure 4.15: Data Distribution for Elimination With outlier with feature selection

Table 4.8: Performance comparison of models after elimination, with outliers, and with feature selection in the PIMA indian dataset

Resampling Technique	ML Model	Accuracy	Precision	Recall	F1-Score	AUC-ROC	MCC
SMOTE	Naive Bayes	0.6571	0.6512	0.5714	0.6087	0.7602	0.3080
	XGBoost	0.8190	0.7778	0.8571	0.8155	0.8735	0.6417
	LightGBM	0.8571	0.8148	0.8980	0.8544	0.8827	0.7181
SMOTEENN	Naive Bayes	0.8906	0.9167	0.8919	0.9041	0.9499	0.7773
	XGBoost	**0.9688**	**0.9487**	**1.0000**	**0.9737**	**0.9960**	**0.9373**
	LightGBM	**0.9688**	**0.9487**	**1.0000**	**0.9737**	**0.9960**	**0.9373**
SMOTETomek	Naive Bayes	0.7400	0.7907	0.6667	0.7234	0.8543	0.4877
	XGBoost	0.8600	0.8491	0.8824	0.8654	0.9144	0.7202
	LightGBM	0.8400	0.8302	0.8627	0.8462	0.9028	0.6802
ADASYN	Naive Bayes	0.6699	0.6905	0.5800	0.6304	0.7815	0.3404
	XGBoost	0.8350	0.7895	0.9000	0.8411	0.8577	0.6772
	LightGBM	0.7961	0.7636	0.8400	0.8000	0.8479	0.5958

Table 4.8 exhibit that the highest outcomes after elimination and feature selection across different evaluation metrics under pre-split SMOTEENN application was by XGBoost and LightGBM. Unlike the previous circumstances, Naive Bayes showed the lowest results with SMOTE resampling here.



(a) SMOTE

(b) SMOTEENN

(c) ADASYN

(d) SMOTETomek

Figure 4.16: ROC curves demonstrating model performance on the PIMA indian dataset for pre-split data resampling Elimination With outlier with feature selection.

No matter which of the eight scenarios we look at for the PIMA indian dataset, whether we handled missing values through imputation or elimination, kept or removed outliers, or applied feature selection the data distribution graphs consistently highlight the clear class imbalance right from the start. More importantly, these charts do a great job of showing how the four resampling techniques (SMOTE, SMOTEENN, SMOTETomek, and ADASYN) successfully smoothed out this imbalance before we even began training the models. This gave us a solid, uniform starting point to fairly evaluate everything moving forward.

While observing the ROC curves throughout all the experimental configurations for the PIMA indian dataset, a clear and consistent pattern is observed for the Naive Bayes, XGBoost, and LightGBM models. In nearly every single test, SMOTEENN achieved the highest AUC-ROC scores, highlighting its powerful ability to effectively separate the classes and identify a higher proportion of actual positive cases (Recall) compared to the other methods. Meanwhile, SMOTE and SMOTETomek behaved very similarly to one another, offering steady, moderate performance across the models. On the downside, ADASYN consistently struggled, resulting in the lowest and most unstable ROC curves throughout the experiments. It is worth noting a minor exception: in the specific case of imputation with outliers but without feature selection, LightGBM showed its lowest AUC-ROC value after the application of SMOTE, rather than ADASYN.

4.1.2 BRFSS-2015 Prediction Result

This section shows the results obtained from applying resampling methods before the train and test split of the BRFSS-2015 dataset in four different circumstances.

Outlier removed without feature selection

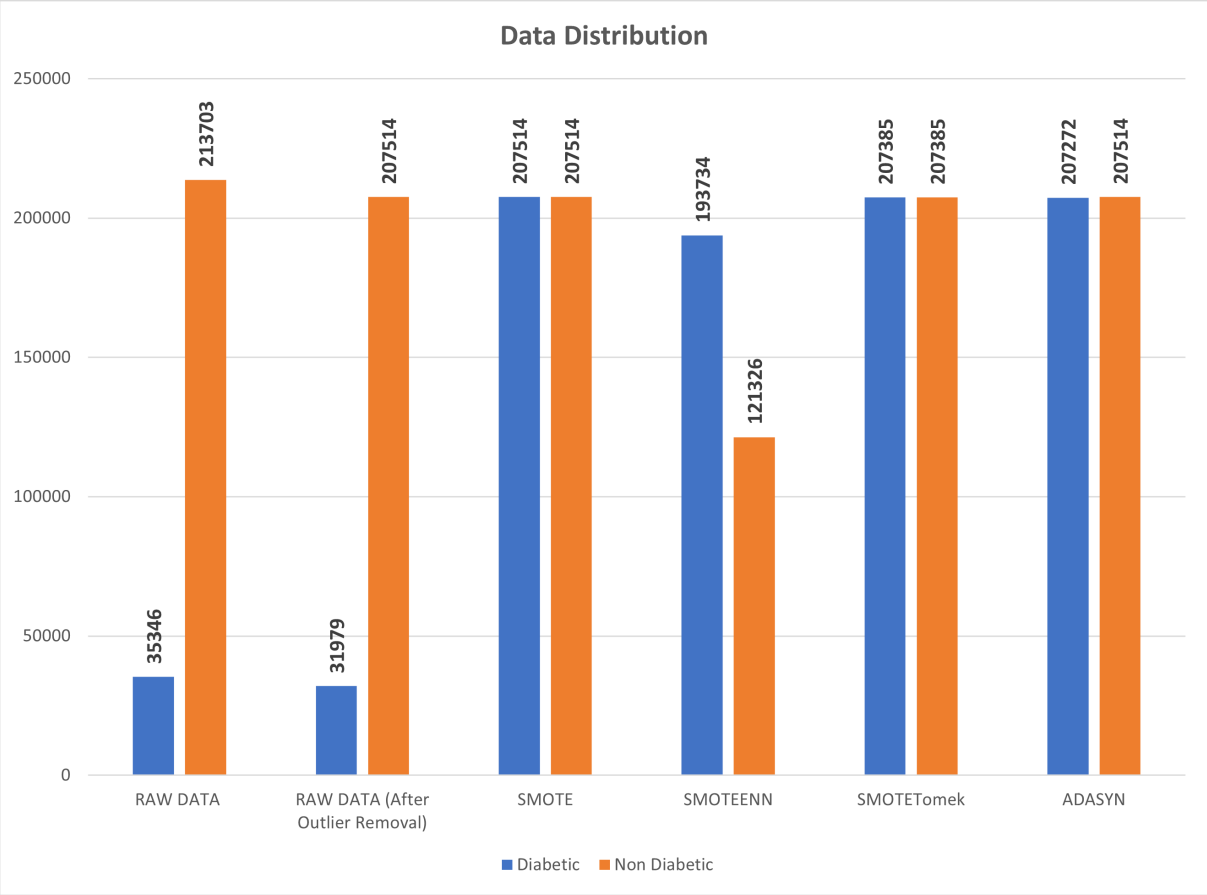
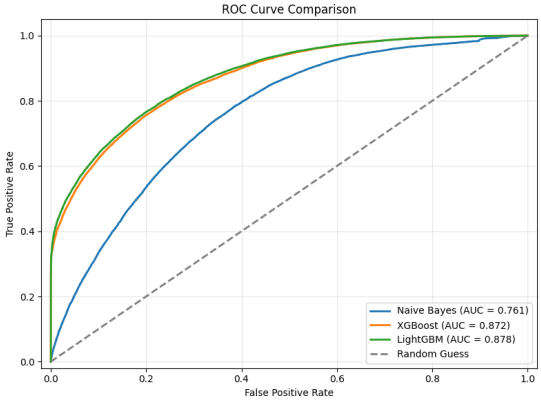


Figure 4.17: Data Distribution for Outlier removed without feature selection

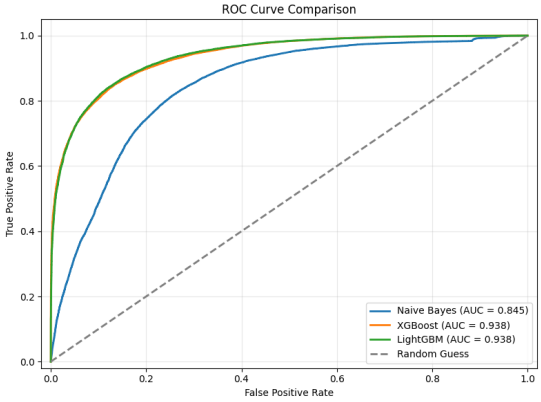
Table 4.9: Performance comparison of models after outlier removal and without feature selection in the BRFSS dataset

Resampling Technique	ML Model	Accuracy	Precision	Recall	F1-Score	AUC-ROC	MCC
SMOTE	Naive Bayes	0.6963	0.6568	0.8217	0.7300	0.7612	0.4056
	XGBoost	0.7777	0.7788	0.7756	0.7772	0.8721	0.5554
	LightGBM	0.7821	0.7799	0.7859	0.7829	0.8776	0.5642
SMOTEENN	Naive Bayes	0.7777	0.8464	0.7798	0.8117	0.8454	0.5441
	XGBoost	0.8605	**0.8812**	0.8936	0.8874	0.9377	0.7044
	LightGBM	**0.8633**	0.8804	**0.8999**	**0.8901**	**0.9385**	**0.7098**
SMOTETomek	Naive Bayes	0.6962	0.6565	0.8210	0.7296	0.7625	0.4056
	XGBoost	0.7757	0.7778	0.7710	0.7744	0.8713	0.5515
	LightGBM	0.7822	0.7807	0.7838	0.7822	0.8763	0.5643
ADASYN	Naive Bayes	0.6798	0.6355	0.8396	0.7234	0.7444	0.3802
	XGBoost	0.7632	0.7620	0.7636	0.7628	0.8565	0.5264
	LightGBM	0.7720	0.7706	0.7729	0.7717	0.8683	0.5441

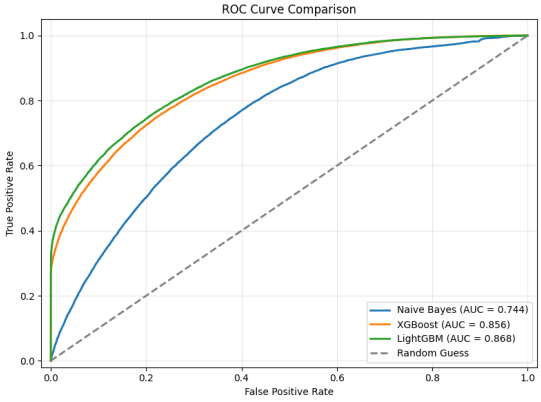
Table 4.9 demonstrates that the highest performance attained without feature selection after removing the outliers is by the XGBoost and LightGBM models after applying pre-split SMOTEENN to balance the data. In contrast, Naive Bayes showed the lowest results under ADASYN resampling alongside the minimum recall value of XGBoost within the same resampling approach.



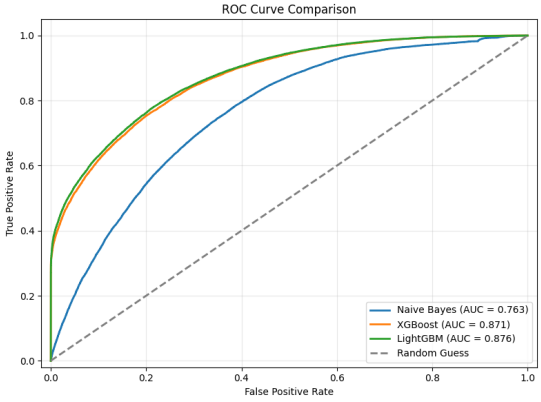
(a) SMOTE



(b) SMOTEENN



(c) ADASYN



(d) SMOTETomek

Figure 4.18: ROC curves demonstrating model performance on the BRFSS dataset for pre-split data resampling Outlier removed without feature selection.

Outlier removed with feature selection

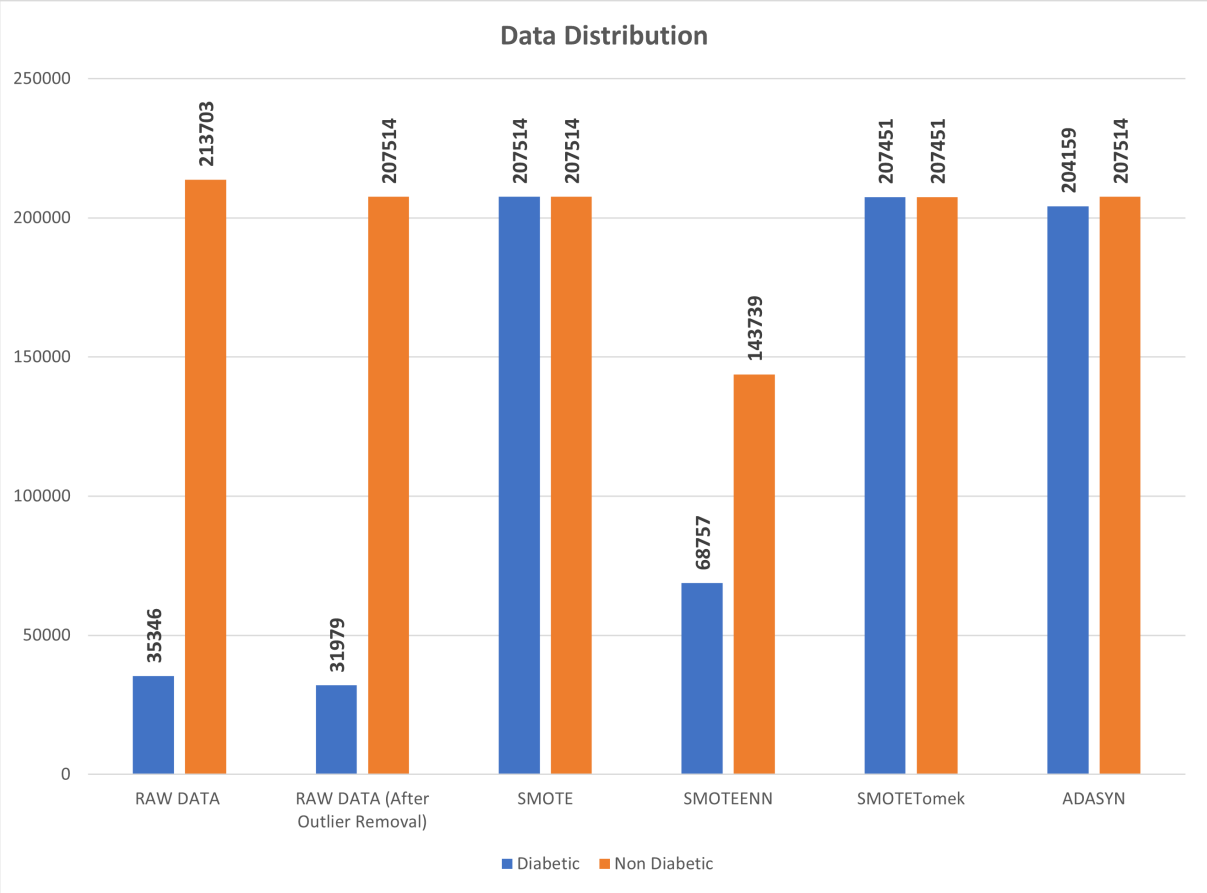
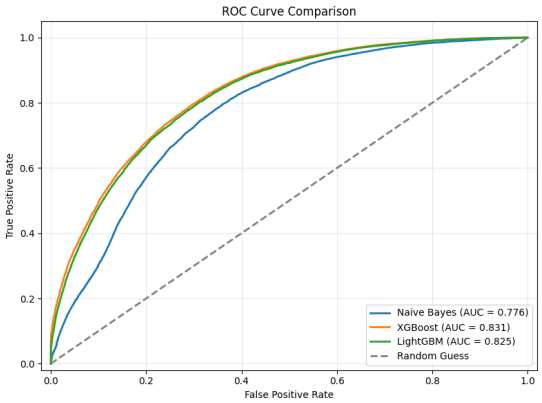


Figure 4.19: Data Distribution for Outlier removed with feature selection

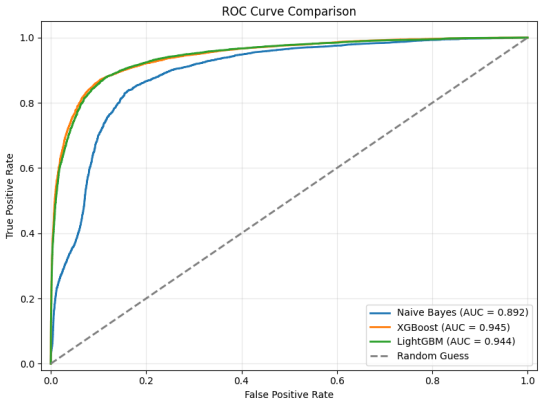
Table 4.10: Performance comparison of models after outlier removal and with feature selection in the BRFSS dataset

Resampling Technique	ML Model	Accuracy	Precision	Recall	F1-Score	AUC-ROC	MCC
SMOTE	Naive Bayes	0.7065	0.7222	0.6707	0.6955	0.7761	0.4140
	XGBoost	0.7483	0.7305	0.7865	0.7575	0.8312	0.4980
	LightGBM	0.7445	0.7245	0.7889	0.7553	0.8247	0.4910
SMOTEENN	Naive Bayes	0.8364	0.7666	0.7196	0.7424	0.8925	0.6234
	XGBoost	**0.8947**	**0.8605**	**0.8096**	**0.8343**	**0.9449**	**0.7579**
	LightGBM	0.8905	0.8516	0.8062	0.8282	0.9437	0.7486
SMOTETomek	Naive Bayes	0.7084	0.7244	0.6723	0.6974	0.7791	0.4178
	XGBoost	0.7491	0.7303	0.7897	0.7588	0.8324	0.4999
	LightGBM	0.7464	0.7248	0.7942	0.7579	0.8273	0.4950
ADASYN	Naive Bayes	0.6784	0.6888	0.6362	0.6615	0.7444	0.3571
	XGBoost	0.7280	0.7031	0.7775	0.7384	0.8038	0.4590
	LightGBM	0.7252	0.6970	0.7849	0.7383	0.7994	0.4547

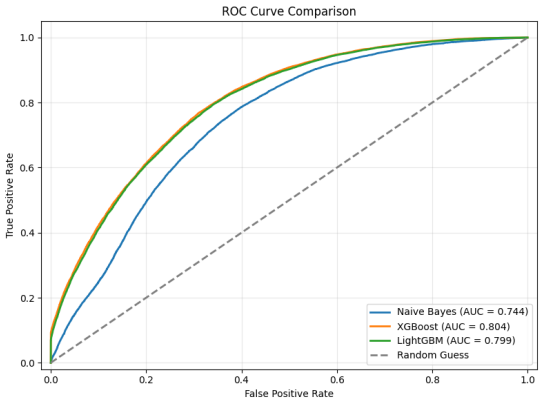
Table 4.10. signifies that XGBoost obtained the highest result with feature selection after removing the outliers via pre-split SMOTEENN implementation. Conversely, Naive Bayes still showed the lowest results under ADASYN resampling even after feature selection.



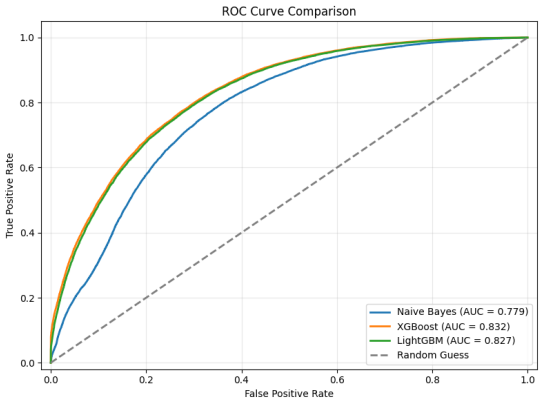
(a) SMOTE



(b) SMOTEENN



(c) ADASYN



(d) SMOTETomek

Figure 4.20: ROC curves demonstrating model performance on the BRFSS dataset for pre-split data resampling Outlier removed with feature selection.

With outlier without feature selection

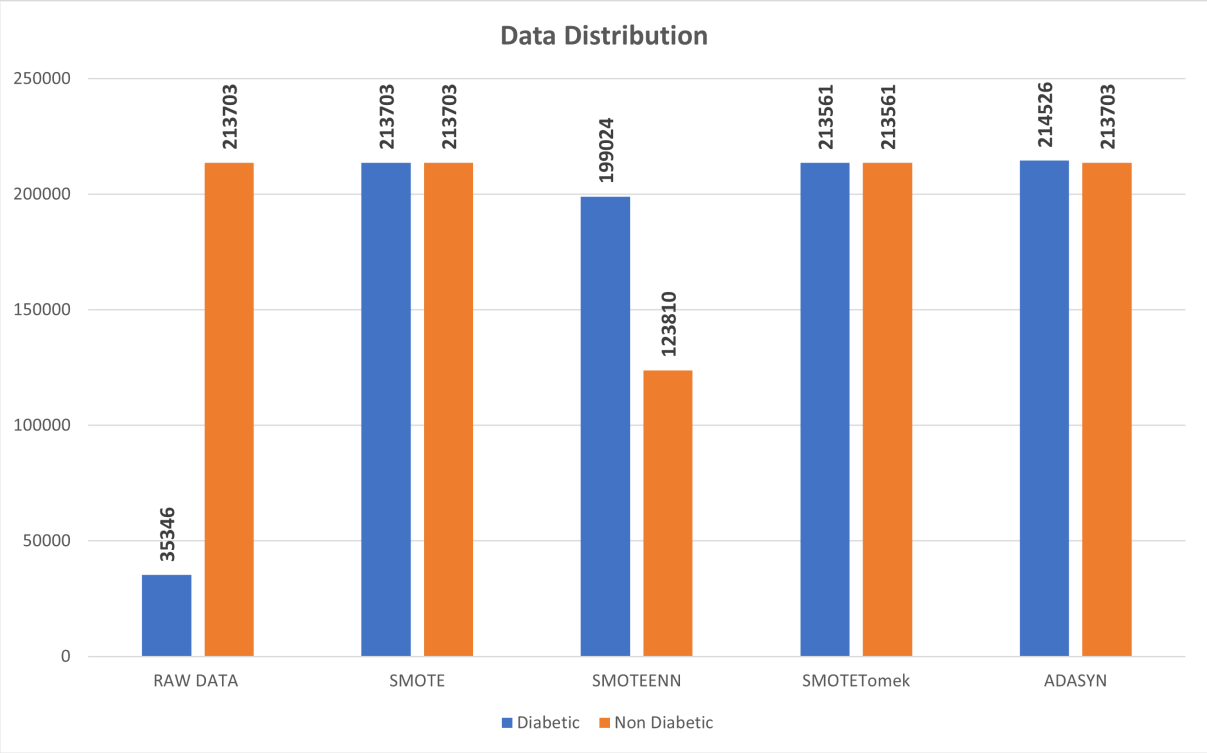
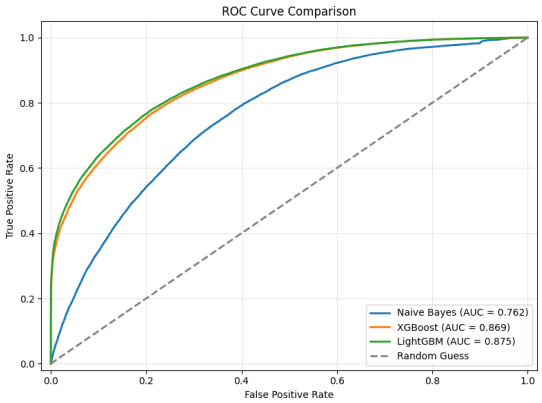


Figure 4.21: Data Distribution for With outlier without feature selection

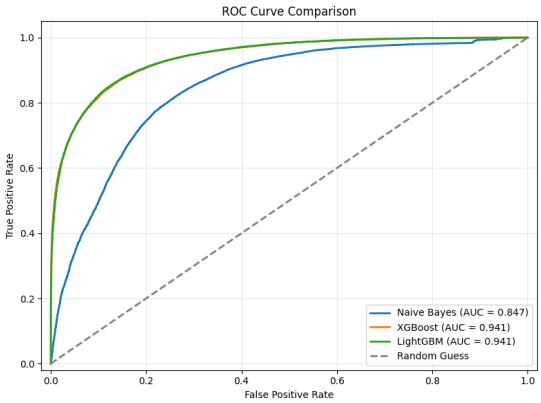
Table 4.11: Performance comparison of models with outliers and without feature selection in the BRFSS dataset

Resampling Technique	ML Model	Accuracy	Precision	Recall	F1-Score	AUC-ROC	MCC
SMOTE	Naive Bayes	0.6953	0.6595	0.8002	0.7230	0.7616	0.4003
	XGBoost	0.7772	0.7753	0.7771	0.7762	0.8695	0.5544
	LightGBM	0.7828	0.7773	0.7892	0.7832	0.8755	0.5656
SMOTEENN	Naive Bayes	0.7765	0.8499	0.7742	0.8103	0.8466	0.5433
	XGBoost	0.8665	**0.8841**	0.9017	0.8928	**0.9407**	0.7161
	LightGBM	**0.8673**	0.8832	**0.9043**	**0.8937**	**0.9407**	**0.7177**
SMOTETomek	Naive Bayes	0.6971	0.6633	0.7990	0.7248	0.7626	0.4030
	XGBoost	0.7785	0.7796	0.7755	0.7775	0.8716	0.5569
	LightGBM	0.7852	0.7831	0.7881	0.7856	0.8777	0.5705
ADASYN	Naive Bayes	0.6821	0.6416	0.8212	0.7204	0.7468	0.3798
	XGBoost	0.7669	0.7635	0.7715	0.7675	0.8589	0.5338
	LightGBM	0.7737	0.7695	0.7798	0.7746	0.8685	0.5474

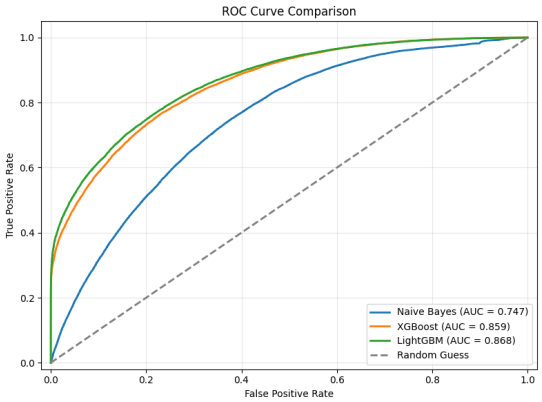
Table 4.11 indicates that XGBoost and LightGBM have achieved the highest results without feature selection across different evaluation metrics under pre-split SMOTEENN application. On the other hand, XGBoost gives the least recall value in addition to Naive Bayes showing the lowest results with ADASYN resampling.



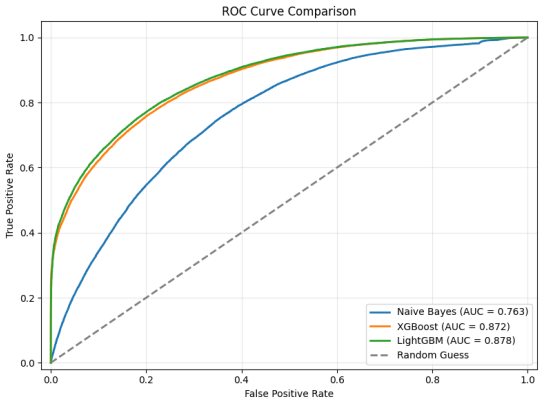
(a) SMOTE



(b) SMOTEENN



(c) ADASYN



(d) SMOTETomek

Figure 4.22: ROC curves demonstrating model performance on the BRFSS dataset for pre-split data resampling With outlier without feature selection.

With outlier with feature selection

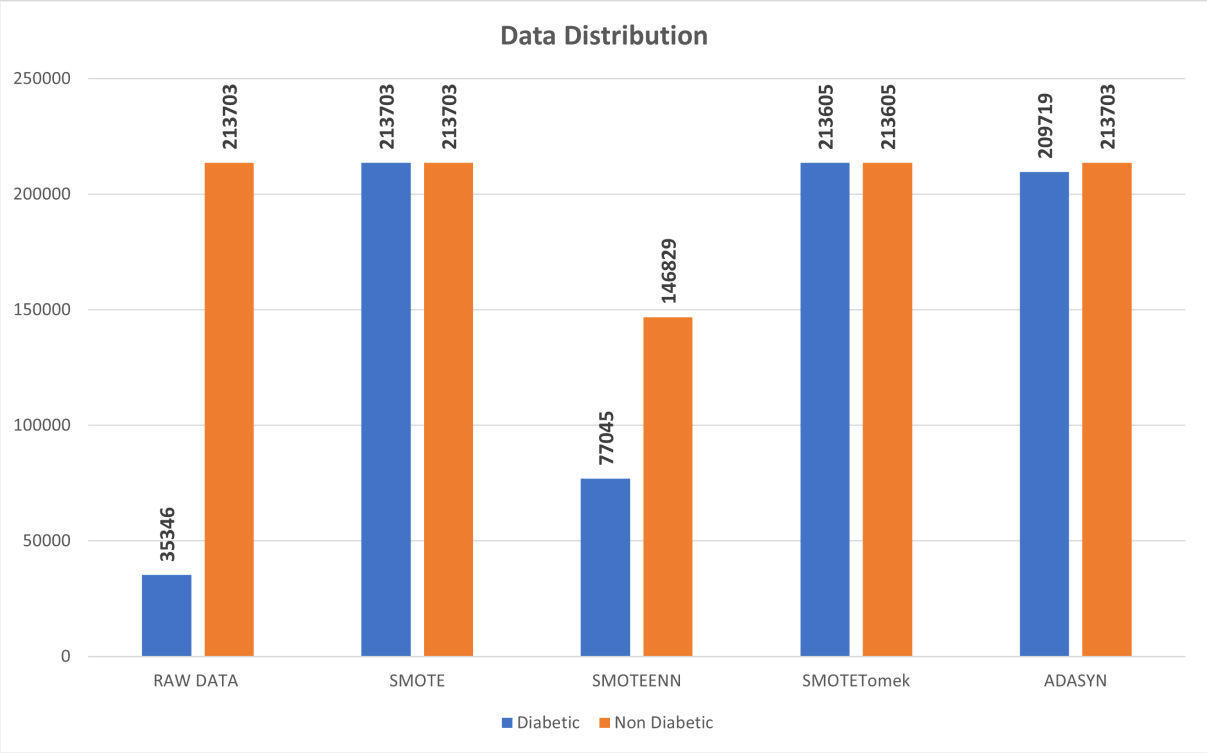
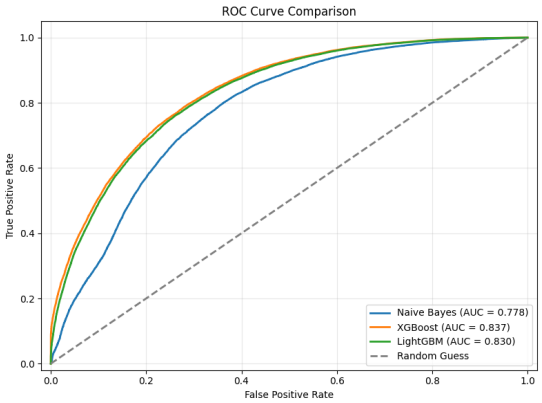


Figure 4.23: Data Distribution for With outlier with feature selection

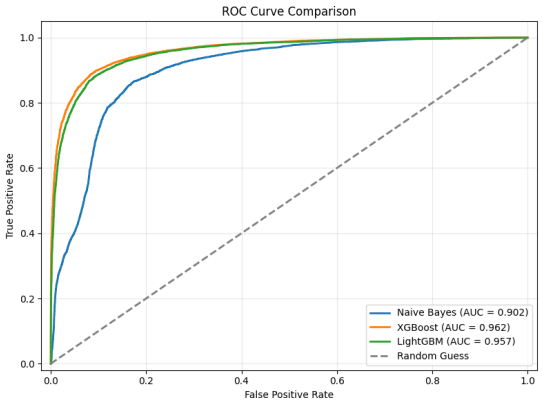
Table 4.12: Performance comparison of models with outliers and with feature selection in the BRFSS dataset

Resampling Technique	ML Model	Accuracy	Precision	Recall	F1-Score	AUC-ROC	MCC
SMOTE	Naive Bayes	0.7065	0.7238	0.6623	0.6917	0.7775	0.4141
	XGBoost	0.7536	0.7320	0.7956	0.7625	0.8369	0.5093
	LightGBM	0.7487	0.7250	0.7970	0.7593	0.8300	0.5002
SMOTEENN	Naive Bayes	0.8438	0.7837	0.7526	0.7679	0.9023	0.6505
	XGBoost	**0.9095**	**0.8840**	**0.8476**	**0.8654**	**0.9622**	**0.7977**
	LightGBM	0.9001	0.8650	0.8400	0.8523	0.9566	0.7770
SMOTETomek	Naive Bayes	0.7081	0.7242	0.6650	0.6933	0.7791	0.4171
	XGBoost	0.7531	0.7322	0.7924	0.7611	0.8370	0.5082
	LightGBM	0.7495	0.7250	0.7977	0.7596	0.8313	0.5018
ADASYN	Naive Bayes	0.6754	0.6910	0.6208	0.6540	0.7427	0.3518
	XGBoost	0.7249	0.7020	0.7702	0.7345	0.8058	0.4523
	LightGBM	0.7223	0.6954	0.7795	0.7351	0.7991	0.4485

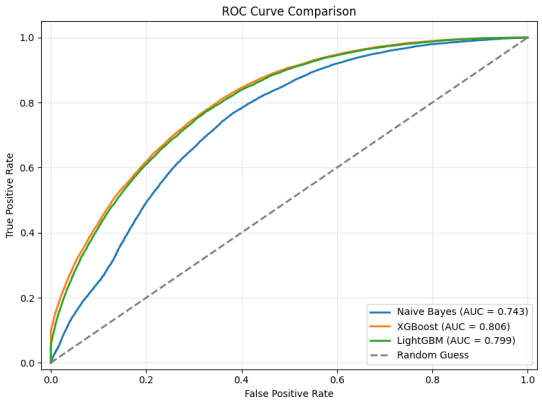
Table 4.12 demonstrate that the highest outcomes after feature selection across different evaluation metrics under pre-split SMOTEENN application was by XGBoost. However, Naive Bayes showed the lowest results with ADASYN resampling here as well.



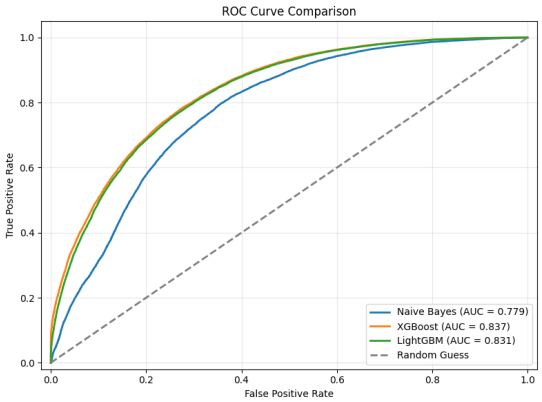
(a) SMOTE



(b) SMOTEENN



(c) ADASYN



(d) SMOTETomek

Figure 4.24: ROC curves demonstrating model performance on the BRFSS dataset for pre-split data resampling With outlier with feature selection.

Figures 4.17, 4.19, 4.21, and 4.23 illustrate the data distribution of the BRFSS-2015 diabetes dataset across all four experimental scenarios. Across every conditionwhether outliers were removed or retained, and whether feature selection was applied or notthe figures consistently demonstrate the initial class imbalance present in the raw data. Furthermore, they visually capture how the application of the four pre-split resampling techniques (SMOTE, SMOTEENN, SMOTETomek, and ADASYN) systematically resolves this imbalance prior to model training.

When we look at the ROC curves across Figures (AUC-ROC)2, 4, 6, and 8, a clear and consistent pattern is observerd for the Naive Bayes, XGBoost, and LightGBM models, regardless of how the data was initially set up. In every single test, SMOTEENN got the highest AUC-ROC scores. This really highlights its ability to effectively separate the classes and catch more of the actual positive cases (Recall) compared to the other methods. Meanwhile, SMOTE and SMOTETomek behaved very similarly to one another, offering a steady, middle-of-the-road performance across all the models. On the downside, ADASYN consistently struggled, leaving us with the lowest and most unstable ROC curves throughout the entire experiment.

4.1.3 DiaHealth Prediction Result

This segment of the study represents the results obtained from applying resampling methods before the train and test split of the DiaHealth dataset in four distinguishable circumstances.

Outlier removed without feature selection

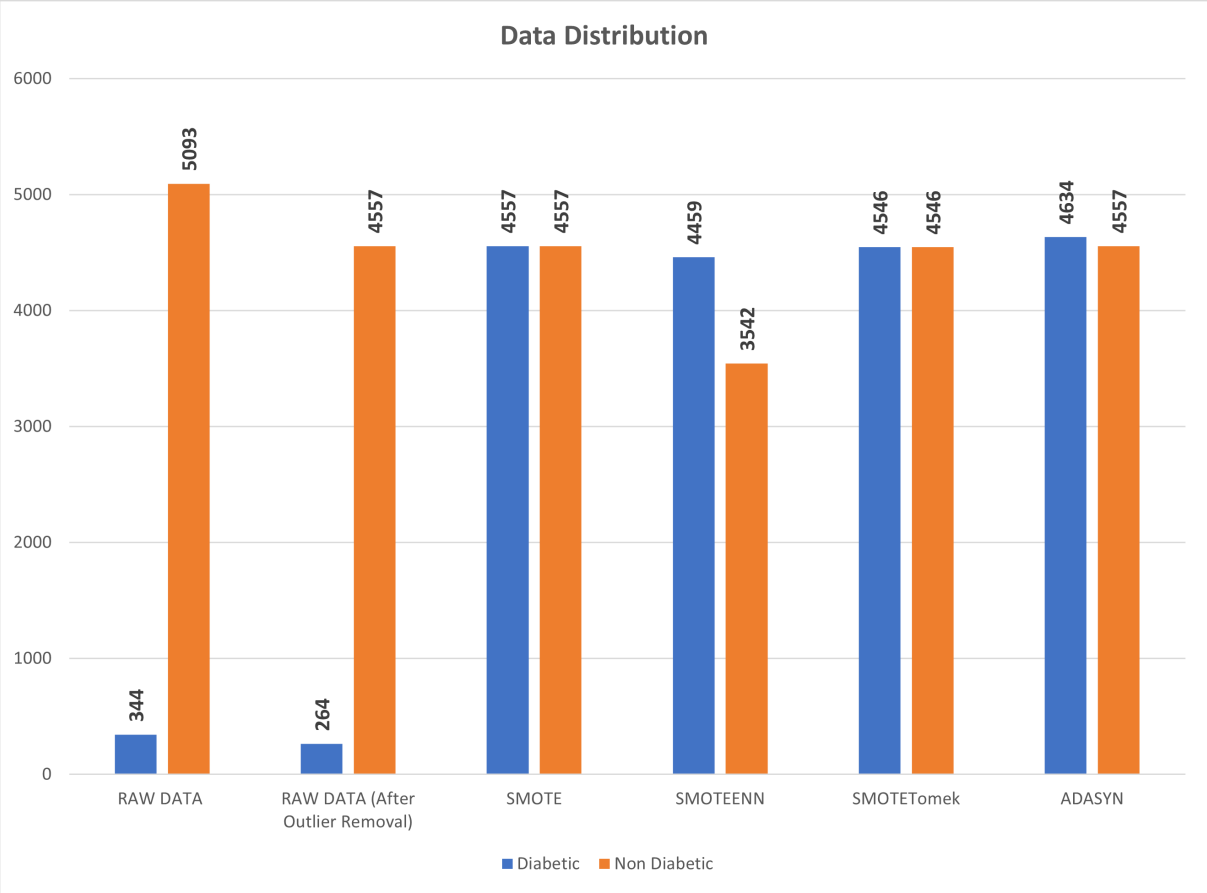


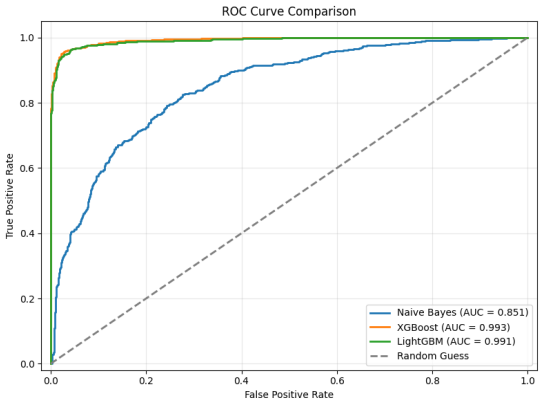
Figure 4.25: Data Distribution for Outlier removed without feature selection

Table 4.13: Performance comparison of models after outlier removal and without feature selection in the DiaHealth dataset

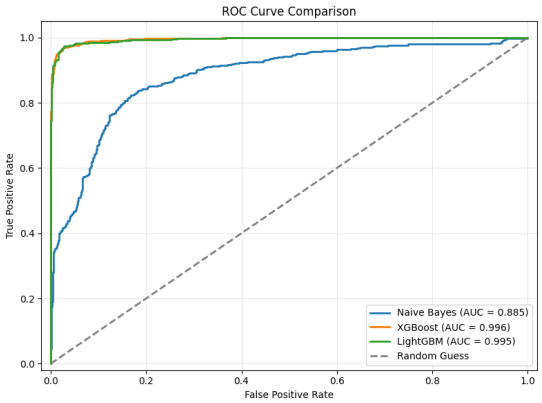
Resampling Technique	ML Model	Accuracy	Precision	Recall	F1-Score	AUC-ROC	MCC
SMOTE	Naive Bayes	0.6703	0.6088	0.9618	0.7457	0.8509	0.4174
	XGBoost	0.9600	0.9557	0.9651	0.9603	0.9930	0.9200
	LightGBM	0.9605	0.9597	0.9618	0.9607	0.9910	0.9210
SMOTEENN	Naive Bayes	0.6602	0.6228	**0.9774**	0.7608	0.8851	0.3602
	XGBoost	**0.9675**	0.9675	0.9740	0.9707	**0.9957**	0.9343
	LightGBM	0.9694	**0.9697**	0.9751	**0.9724**	0.9948	**0.9381**
SMOTETomek	Naive Bayes	0.6449	0.5892	0.9815	0.7363	0.8669	0.3871
	XGBoost	0.9665	0.9613	0.9728	0.9670	0.9956	0.9330
	LightGBM	0.9670	0.9613	0.9739	0.9676	0.9951	0.9341
ADASYN	Naive Bayes	0.6661	0.6048	0.9576	0.7414	0.8376	0.4091
	XGBoost	0.9608	0.9549	0.9674	0.9611	0.9932	0.9218
	LightGBM	0.9570	0.9585	0.9554	0.9569	0.9915	0.9141

Table 4.13 illustrates that the highest performance attained without feature selection after removing the outliers is by the XGBoost and LightGBM models, but the maximum recall was attained by Naive Bayes, after applying pre-split SMOTEENN to balance the data. In contrast, Naive Bayes showed the lowest results under SMOTETomek and ADASYN

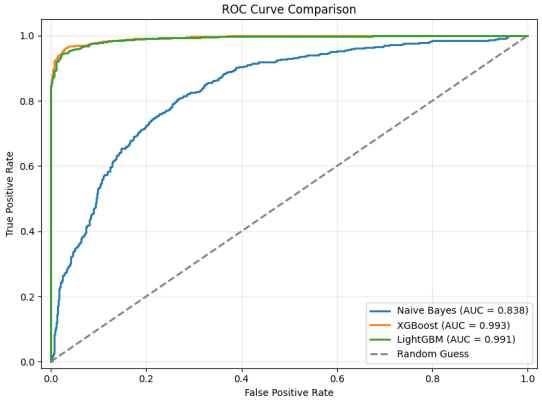
resamplings alongside the LightGBMs lowest recall with ADASYN and Naive Bayess lowest MCC under SMOTEENN.



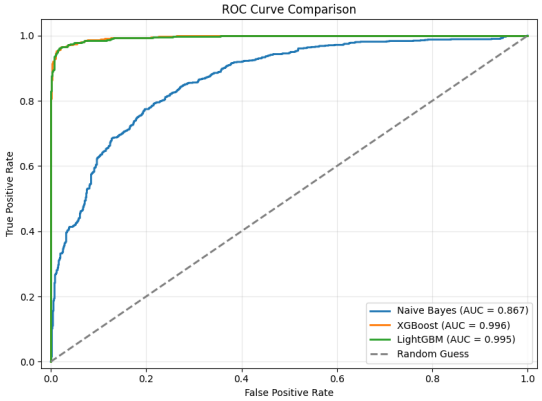
(a) SMOTE



(b) SMOTEENN



(c) ADASYN



(d) SMOTETomek

Figure 4.26: ROC curves demonstrating model performance on the DiaHealth dataset for pre-split data resampling Outlier removed without feature selection.

Outlier removed with feature selection

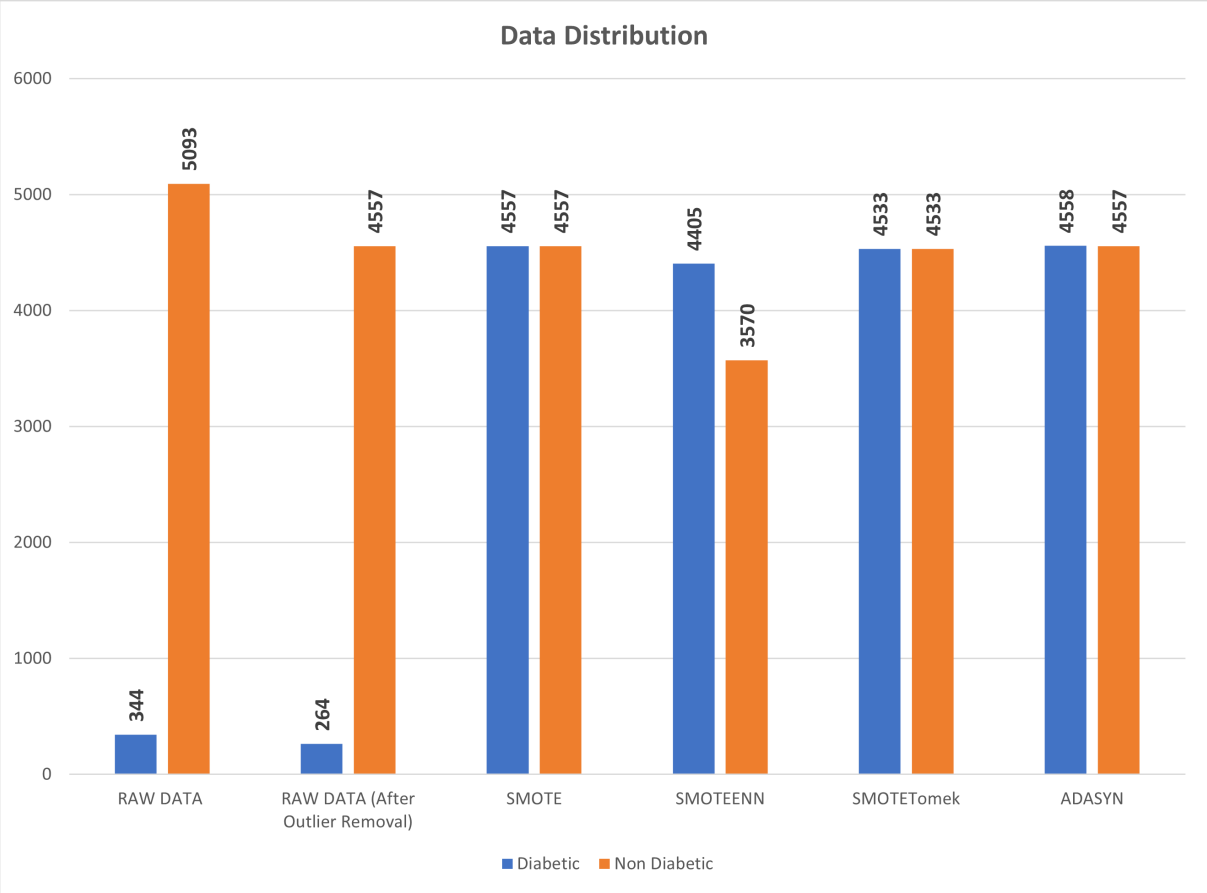
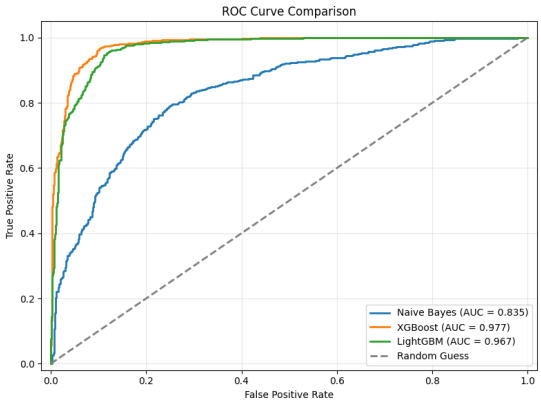


Figure 4.27: Data Distribution for Outlier removed with feature selection

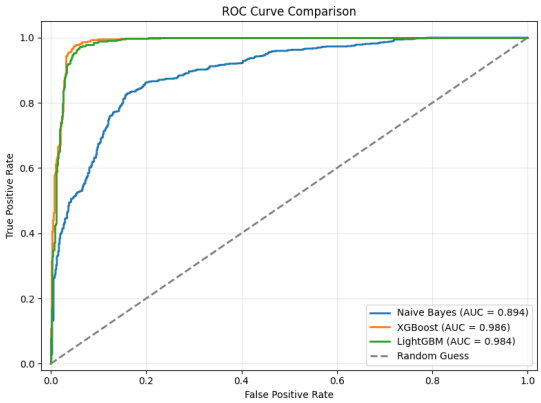
Table 4.14: Performance comparison of models after outlier removal and with feature selection in the DiaHealth dataset

Resampling Technique	ML Model	Accuracy	Precision	Recall	F1-Score	AUC-ROC	MCC
SMOTE	Naive Bayes	0.7499	0.8026	0.6659	0.7279	0.8350	0.5076
	XGBoost	0.9292	0.9086	0.9552	0.9313	0.9772	0.8596
	LightGBM	0.9161	0.8913	0.9487	0.9191	0.9674	0.8338
SMOTEENN	Naive Bayes	0.8169	0.8728	0.7865	0.8274	0.8943	0.6375
	XGBoost	**0.9636**	**0.9502**	**0.9865**	**0.9680**	**0.9863**	**0.9268**
	LightGBM	0.9549	0.9455	0.9753	0.9602	0.9836	0.9087
SMOTETomek	Naive Bayes	0.7635	0.8045	0.6996	0.7484	0.8450	0.5319
	XGBoost	0.9454	0.9230	0.9726	0.9471	0.9837	0.8921
	LightGBM	0.9283	0.9039	0.9594	0.9309	0.9775	0.8583
ADASYN	Naive Bayes	0.7499	0.8005	0.6652	0.7266	0.8311	0.5070
	XGBoost	0.9325	0.9087	0.9616	0.9344	0.9798	0.8665
	LightGBM	0.9155	0.8866	0.9528	0.9185	0.9692	0.8334

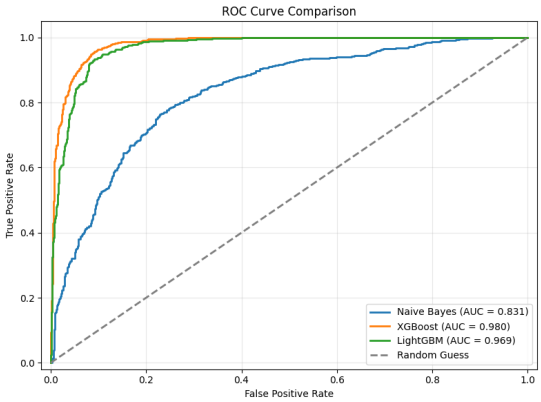
Table 4.14 signifies that XGBoost obtained the highest result with feature selection after removing the outliers via pre-split SMOTEENN implementation. Conversely, Naive Bayes showed the lowest results under ADASYN resampling even after feature selection



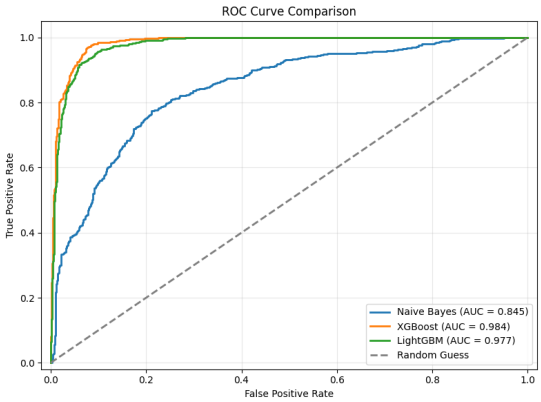
(a) SMOTE



(b) SMOTEENN



(c) ADASYN



(d) SMOTETomek

Figure 4.28: ROC curves demonstrating model performance on the DiaHealth dataset for pre-split data resampling Outlier removed with feature selection.

With outlier without feature selection

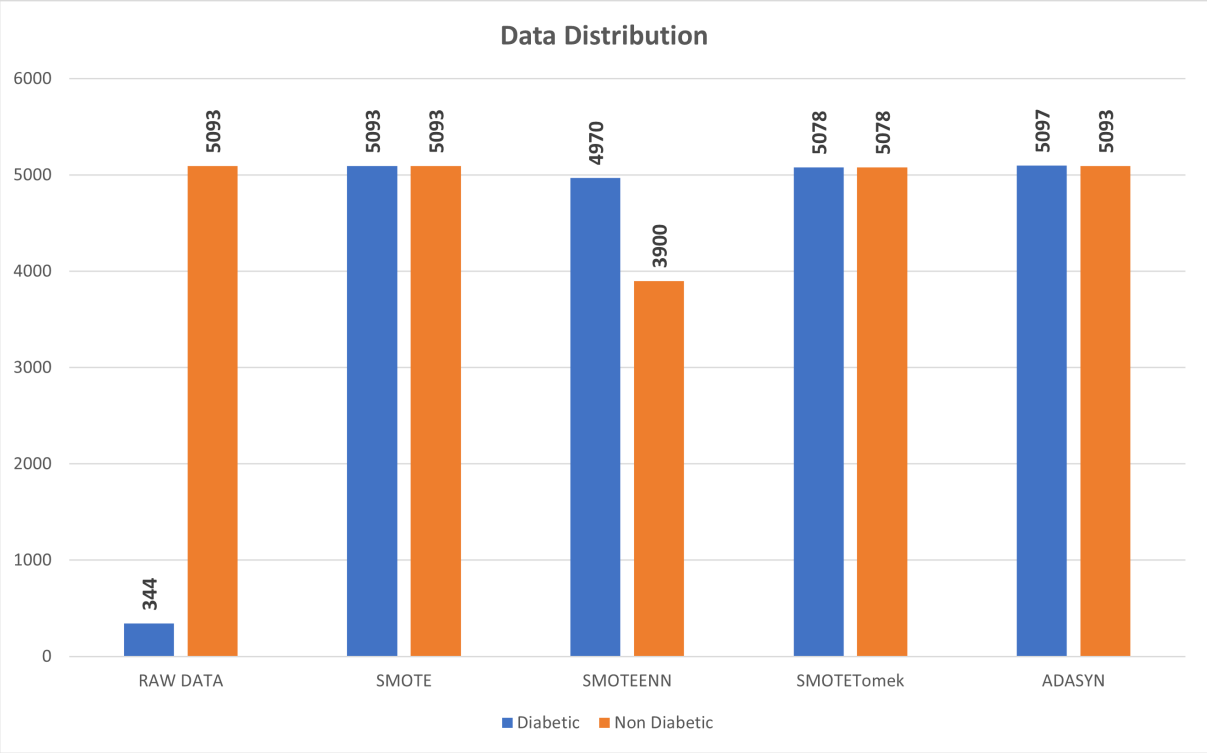
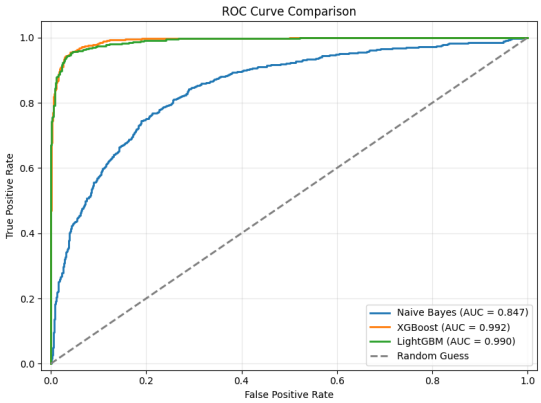


Figure 4.29: Data Distribution for With outlier without feature selection

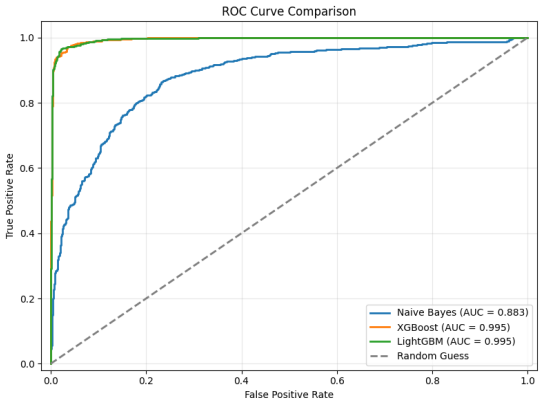
Table 4.15: Performance comparison of models with outliers and without feature selection in the DiaHealth dataset

Resampling Technique	ML Model	Accuracy	Precision	Recall	F1-Score	AUC-ROC	MCC
SMOTE	Naive Bayes	0.5937	0.5605	0.9723	0.7111	0.8471	0.2662
	XGBoost	0.9539	0.9517	0.9590	0.9553	0.9917	0.9077
	LightGBM	0.9539	0.9534	0.9571	0.9552	0.9900	0.9077
SMOTEENN	Naive Bayes	0.6804	0.6460	0.9703	0.7756	0.8827	0.3762
	XGBoost	**0.9679**	**0.9676**	**0.9762**	**0.9719**	**0.9948**	**0.9344**
	LightGBM	0.9639	0.9619	0.9752	0.9685	0.9947	0.9264
SMOTETomek	Naive Bayes	0.6698	0.6184	0.9710	0.7556	0.8458	0.4038
	XGBoost	0.9542	0.9535	0.9597	0.9566	0.9931	0.9082
	LightGBM	0.9547	0.9639	0.9494	0.9566	0.9912	0.9094
ADASYN	Naive Bayes	0.6001	0.5675	0.9746	0.7173	0.8351	0.2727
	XGBoost	0.9504	0.9528	0.9519	0.9524	0.9907	0.9007
	LightGBM	0.9563	0.9594	0.9566	0.9580	0.9908	0.9125

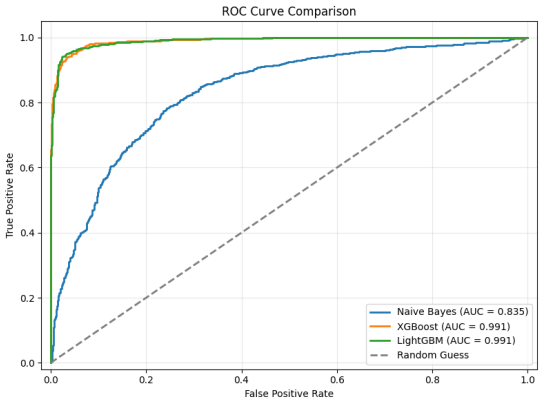
Table 4.15 indicates that XGBoost has achieved the highest results without feature selection across different evaluation metrics under pre-split SMOTEENN application. Whereas, Naive Bayes showed the lowest results with SMOTE and least AUC-ROCwith ADASYN resampling. Lowest recall was displayed by LightGBM under SMOTETomek resampling.



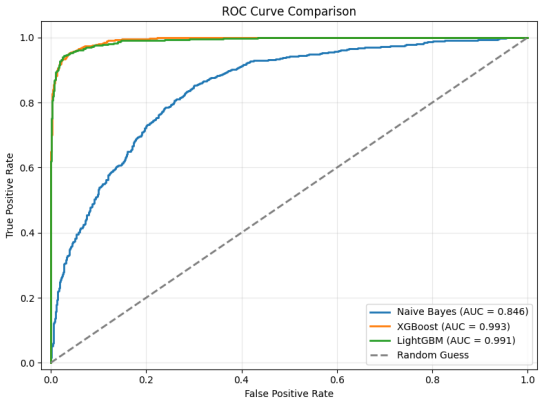
(a) SMOTE



(b) SMOTEENN



(c) ADASYN



(d) SMOTETomek

Figure 4.30: ROC curves demonstrating model performance on the DiaHealth dataset for pre-split data resampling With outlier without feature selection.

With outlier with feature selection

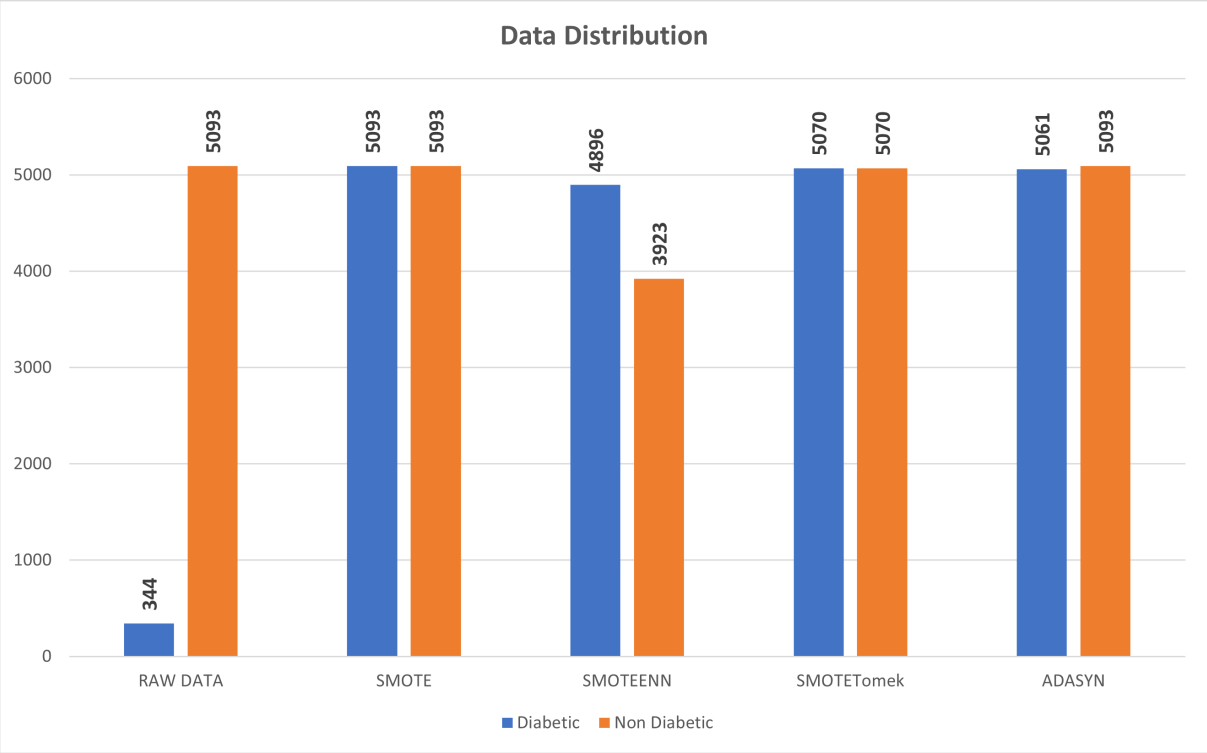
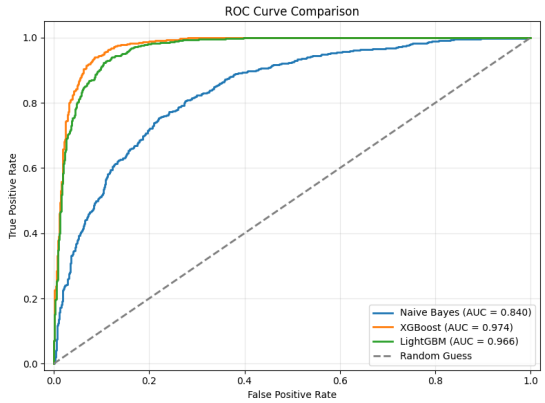


Figure 4.31: Data Distribution for With outlier with feature selection

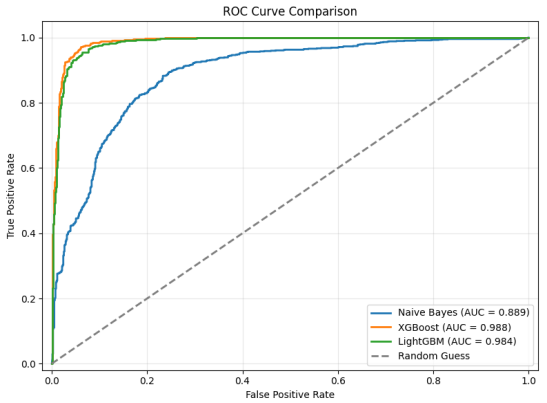
Table 4.16: Performance comparison of models with outliers and with feature selection in the DiaHealth dataset

Resampling Technique	ML Model	Accuracy	Precision	Recall	F1-Score	AUC-ROC	MCC
SMOTE	Naive Bayes	0.7645	0.7840	0.7481	0.7656	0.8397	0.5298
	XGBoost	0.9249	0.9035	0.9561	0.9291	0.9740	0.8510
	LightGBM	0.9082	0.8854	0.9437	0.9136	0.9655	0.8179
SMOTEENN	Naive Bayes	0.8118	0.8567	0.7916	0.8228	0.8887	0.6250
	XGBoost	**0.9546**	**0.9452**	**0.9743**	**0.9596**	**0.9879**	**0.9085**
	LightGBM	0.9461	0.9364	0.9682	0.9520	0.9841	0.8913
SMOTETomek	Naive Bayes	0.7712	0.7926	0.7483	0.7698	0.8392	0.5437
	XGBoost	0.9305	0.9140	0.9537	0.9335	0.9732	0.8616
	LightGBM	0.9147	0.8993	0.9383	0.9184	0.9672	0.8299
ADASYN	Naive Bayes	0.7420	0.7663	0.7195	0.7421	0.8183	0.4855
	XGBoost	0.9276	0.9004	0.9666	0.9324	0.9754	0.8572
	LightGBM	0.8981	0.8692	0.9447	0.9053	0.9625	0.7986

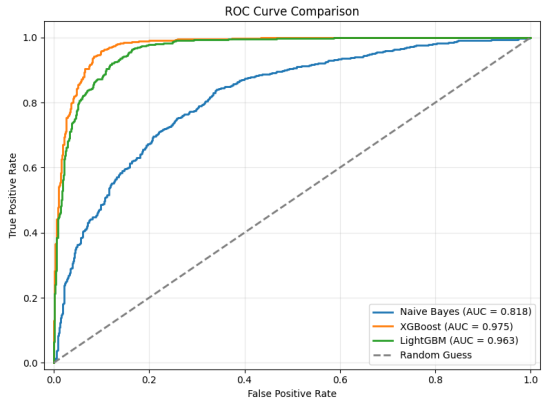
Table 4.16 demonstrate that the highest outcomes after feature selection across different evaluation metrics under pre-split SMOTEENN application was by XGBoost. However, Naive Bayes showed the lowest results with SMOTE resampling here accompanying with the least AUC-ROC under ADASYN. LightGBM exhibited the minimum recall with SMOTETomek.



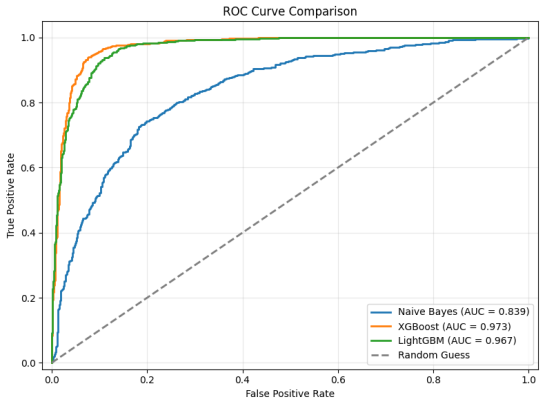
(a) SMOTE



(b) SMOTEENN



(c) ADASYN



(d) SMOTETomek

Figure 4.32: ROC curves demonstrating model performance on the DiaHealth dataset for pre-split data resampling With outlier with feature selection.

Looking at the data distribution for the DiaHealth dataset across all four circumstances whether outliers were removed or kept, and with or without feature selectionthe visuals consistently highlight the original class imbalance found in the raw data. More importantly, they effectively show how employing the four resampling methods (SMOTE, SMOTEENN, SMOTETomek, and ADASYN) successfully levels out this imbalance before any model training begins.

When evaluating the ROC curves across the different scenarios for the DiaHealth dataset, a consistent performance trend is clear for the Naive Bayes, XGBoost, and LightGBM models. In every setup, SMOTEENN shows much higher AUC-ROC values which refers to an improved True Positive Rate (Recall) and far better class separation than the other techniques.Also, SMOTE and SMOTETomek deliver a moderate and closely similar performance to one another across the models. On the other hand, ADASYN generally falls a bit short, showing slightly lower and less stable ROC curves, which is particularly noticeable when running the Naive Bayes model.

4.2 Post-split Prediction Result

Post-split prediction result refers to results achieved by splitting the dataset into training and testing sets before resampling to solve the class imbalance. In this study, we have applied resampling techniques like SMOTE, SMOTEENN, SMOTETomek, and ADASYN only on the training set by ensuring that the test set is entirely hidden during resampling

to avoid any data leakage, preserving the integrity of evaluation in diabetes prediction research and providing an impartial review of model generalization [31][38].

In the result tables in this section contains rows named RAW data. Here, RAW data is refering to the dataset after pre-processing.

4.2.1 PIMA Indian Diabetes Dataset (PIDD) Prediction Result

This section shows the results attained from applying resampling methods after the train and test split of the Pima Indian Diabetes Dataset in eight different circumstances.

Imputation & Outlier removed without feature selection

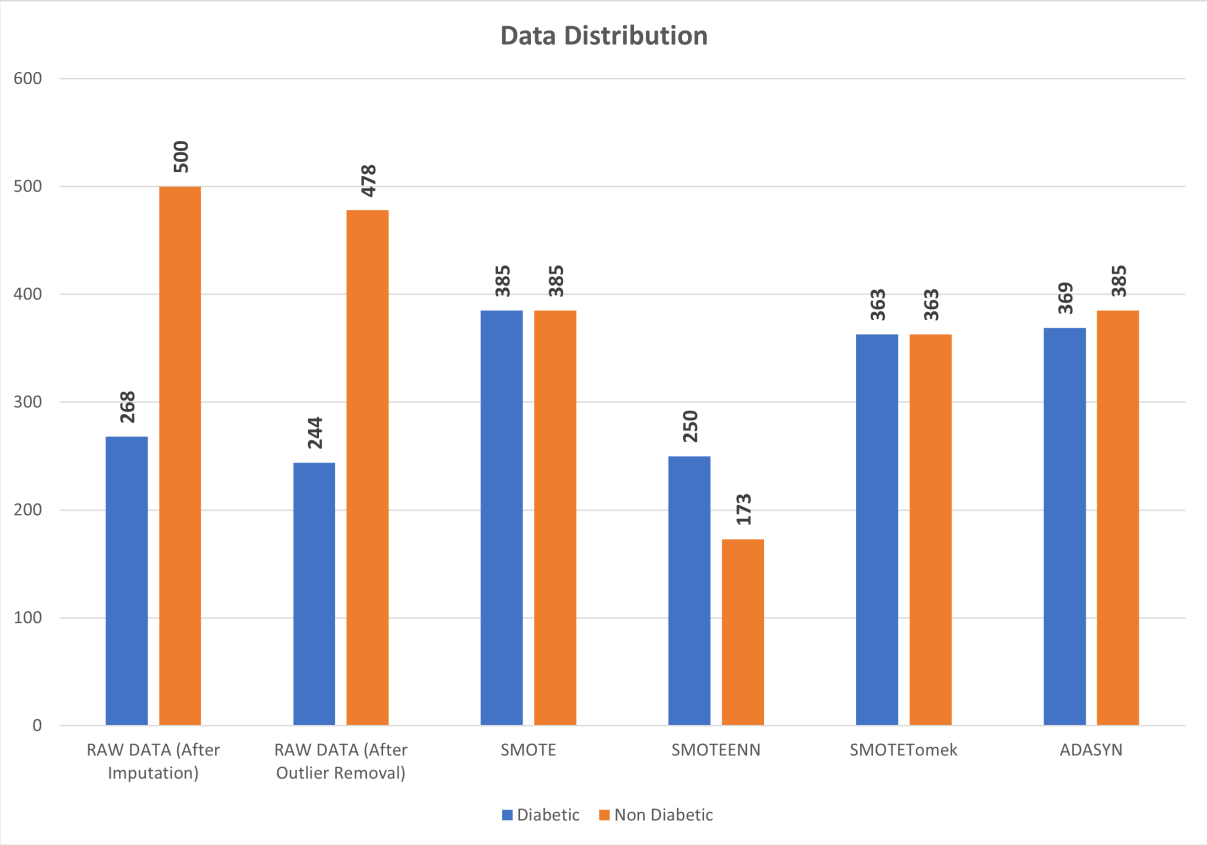
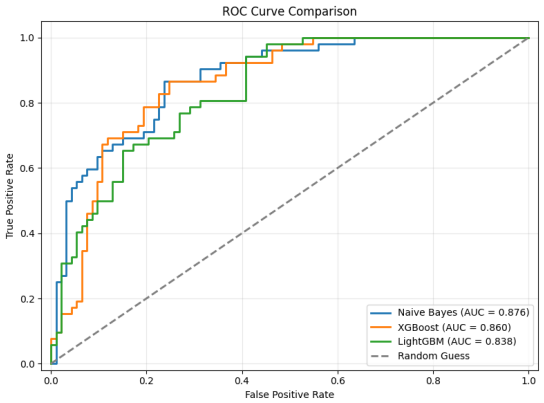


Figure 4.33: Data Distribution for Imputation & Outlier removed without feature selection

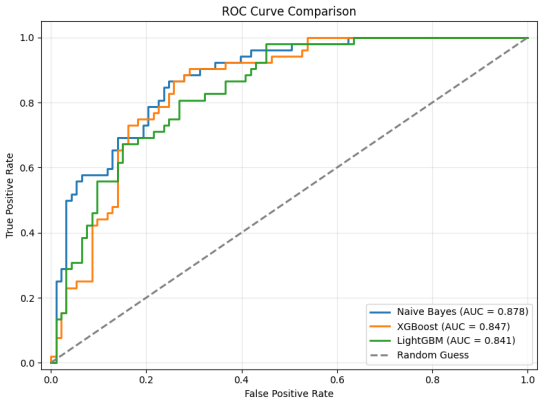
Table 4.17: Performance comparison of models with imputation, outliers removed, and without feature selection in the PIMA indian dataset

Resampling Technique	ML Model	Accuracy	Precision	Recall	F1-Score	AUC-ROC	MCC
RAW DATA	Naive Bayes	0.7862	0.7143	0.6731	0.6931	0.8759	0.5298
	XGBoost	**0.8138**	**0.7778**	0.6731	0.7216	0.8604	**0.5863**
	LightGBM	0.7655	0.6957	0.6154	0.6531	0.8383	0.4790
SMOTE	Naive Bayes	0.7724	0.6727	0.7115	0.6916	0.8776	0.5120
	XGBoost	0.7793	0.7083	0.6538	0.6800	0.8472	0.5129
	LightGBM	0.7655	0.6731	0.6731	0.6731	0.8406	0.4903
SMOTEENN	Naive Bayes	0.7862	0.6522	**0.8654**	**0.7438**	0.8631	0.5832
	XGBoost	0.7586	0.6232	0.8269	0.7107	0.8368	0.5256
	LightGBM	0.7586	0.6164	**0.8654**	0.7200	0.8381	0.5413
SMOTETomek	Naive Bayes	0.7655	0.6607	0.7115	0.6852	0.8766	0.4996
	XGBoost	0.8069	0.7500	0.6923	0.7200	0.8648	0.5741
	LightGBM	0.7793	0.6923	0.6923	0.6923	0.8565	0.5203
ADASYN	Naive Bayes	0.7862	0.6667	0.8077	0.7304	**0.8790**	0.5630
	XGBoost	0.7517	0.6667	0.6154	0.6400	0.8433	0.4518
	LightGBM	0.7724	0.6792	0.6923	0.6857	0.8515	0.5074

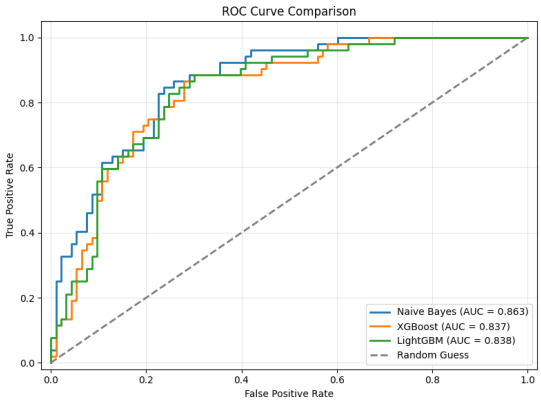
Table 4.17 demonstrates that the highest performance in accuracy, precision and MCC were attained without feature selection after imputation and removing the outliers is by the XGBoost model with raw data. In contrast, LightGBM showed lowest recall with raw data, XGBoost showed least AUC-ROC value and LightGBM displayed least precision both with SMOTEENN, but XGBoost obtained most of the lowest results like in accuracy, recall, F1-score and MCC under ADASYN.



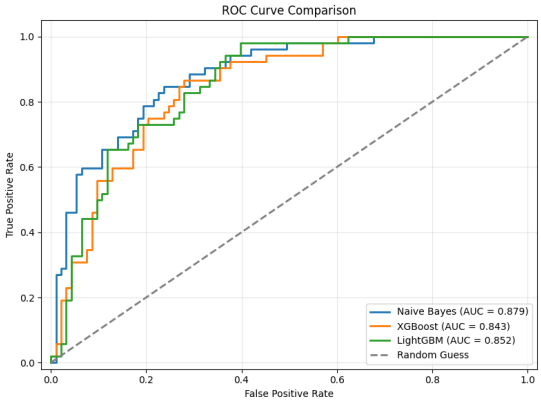
(a) RAW(No Resampling techniques)



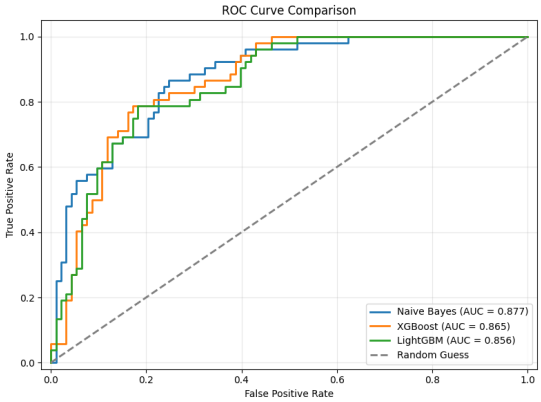
(b) SMOTE



(c) SMOTEENN



(d) ADASYN



(e) SMOTETomek

Figure 4.34: ROC curves demonstrating model performance on the PIMA indian dataset for post-split data resampling Imputation and Outlier removed without feature selection.

Elimination & Outlier removed without feature selection

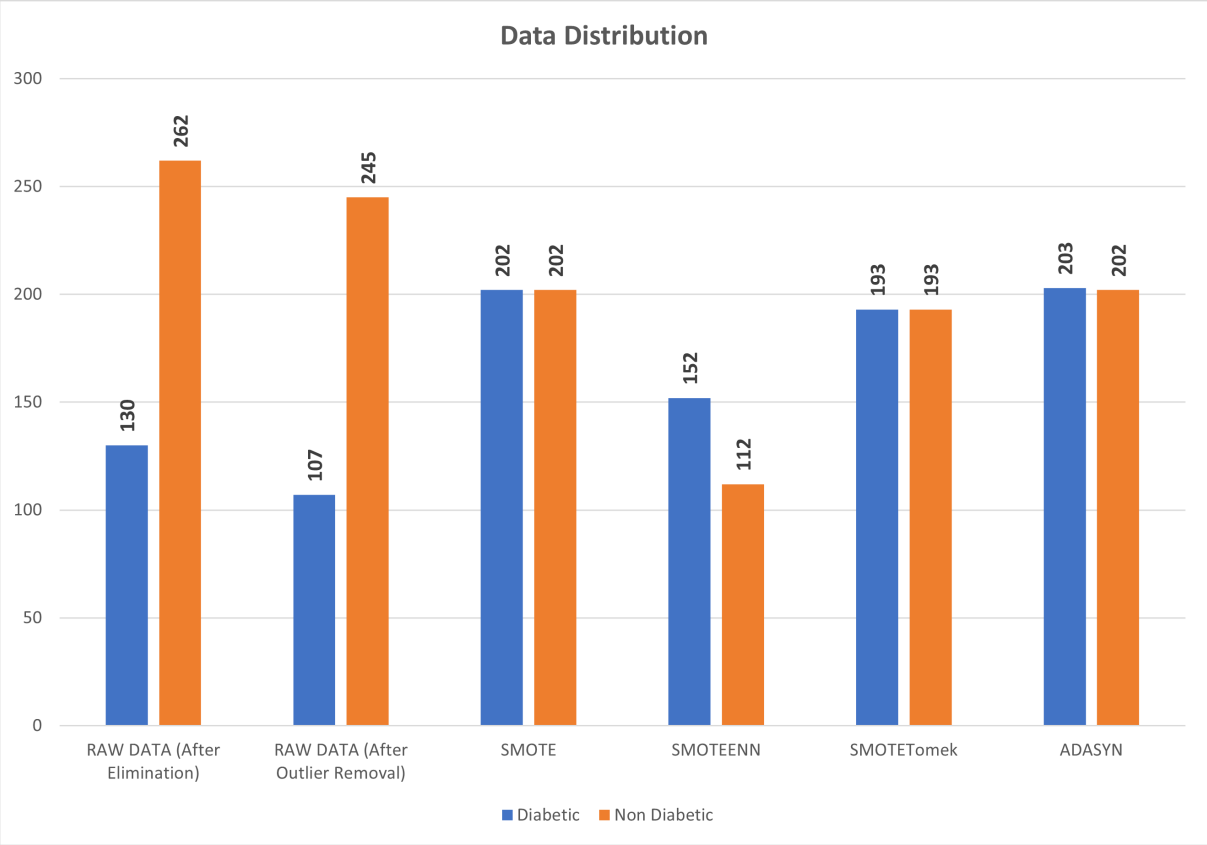


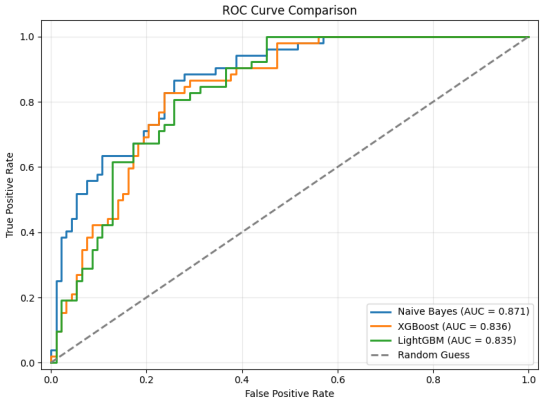
Figure 4.35: Data Distribution for Elimination & Outlier removed without feature selection

Table 4.18: Performance comparison of models with elimination, outliers removed, and without feature selection in the PIMA indian dataset

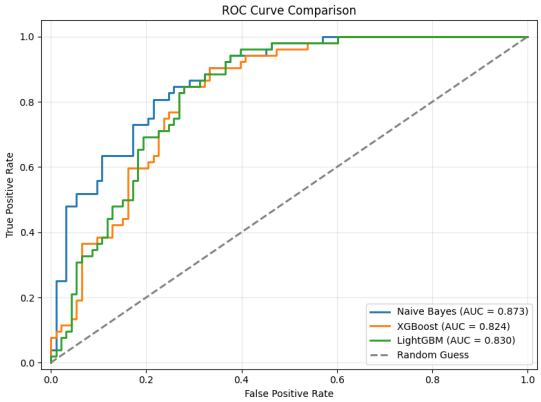
Resampling Technique	ML Model	Accuracy	Precision	Recall	F1-Score	AUC-ROC	MCC
RAW DATA	Naive Bayes	0.7324	0.68	0.6071	0.6415	0.8455	0.4309
	XGBoost	0.7042	0.6667	0.5	0.5714	0.7782	0.3611
	LightGBM	0.7606	**0.7619**	0.5714	0.6531	0.8306	0.4874
SMOTE	Naive Bayes	0.7324	0.6552	0.6786	0.6667	0.8455	0.4434
	XGBoost	0.6761	0.5758	0.6786	0.623	0.789	0.3459
	LightGBM	0.7042	0.6296	0.6071	0.6182	0.8189	0.3771
SMOTEENN	Naive Bayes	0.7606	0.6667	**0.7857**	0.7213	0.8447	0.5192
	XGBoost	0.7465	0.6786	0.6786	0.6786	0.8015	0.4693
	LightGBM	0.7606	0.7037	0.6786	0.6909	0.8472	0.4958
SMOTETomek	Naive Bayes	0.7324	0.6552	0.6786	0.6667	0.8439	0.4434
	XGBoost	**0.7746**	0.7	0.75	**0.7241**	0.8056	**0.535**
	LightGBM	0.7465	0.7083	0.6071	0.6538	0.8206	0.4591
ADASYN	Naive Bayes	0.7465	0.6667	0.7143	0.6897	**0.8497**	0.4766
	XGBoost	0.7042	0.6129	0.6786	0.6441	0.7874	0.3937
	LightGBM	0.7042	0.6207	0.6429	0.6316	0.3937	0.3848

Table 4.18 demonstrates that the highest performance in accuracy, F1-score and MCC were attained without feature selection after elimination and removing the outliers is by

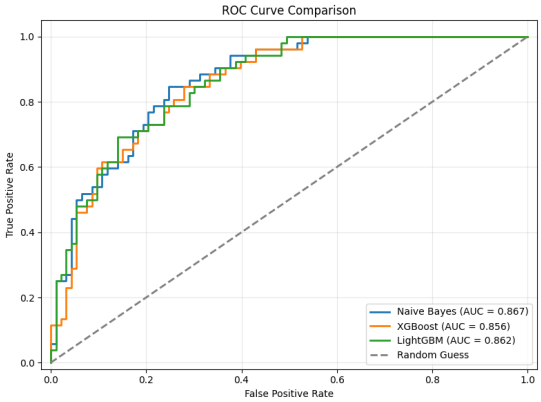
the XGBoost after applying pre-split SMOTETomek to balance the train data. LightGBM gave maximum precision with raw data, Naive Bayes gave highest recall with SMOTEENN, and Naive Bayes also gave maximum AUC-ROC with ADASYN. In contrast, XGBoost showed the lowest results in accuracy, precision and MCC under SMOTE alongside showing the lowest recall and F1-score with raw data. Additionally, LightGBM showed the lowest AUC-ROC with ADASYN.



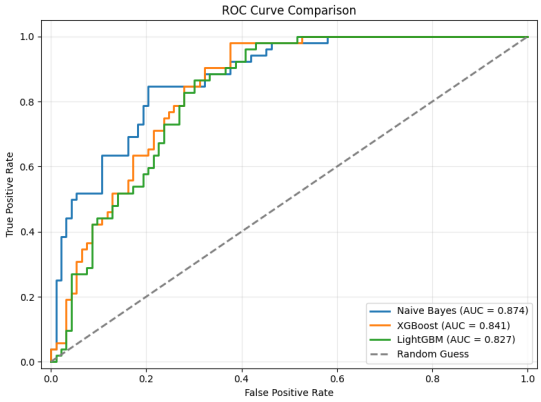
(a) RAW(No Resampling techniques)



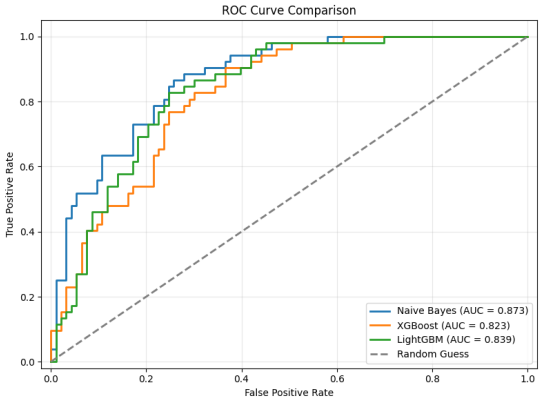
(b) SMOTE



(c) SMOTEENN



(d) ADASYN



(e) SMOTETomek

Figure 4.36: ROC curves demonstrating model performance on the PIMA indian dataset for post-split data resampling Elimination and Outlier removed without feature selection.

Imputation & Outlier removed with feature selection

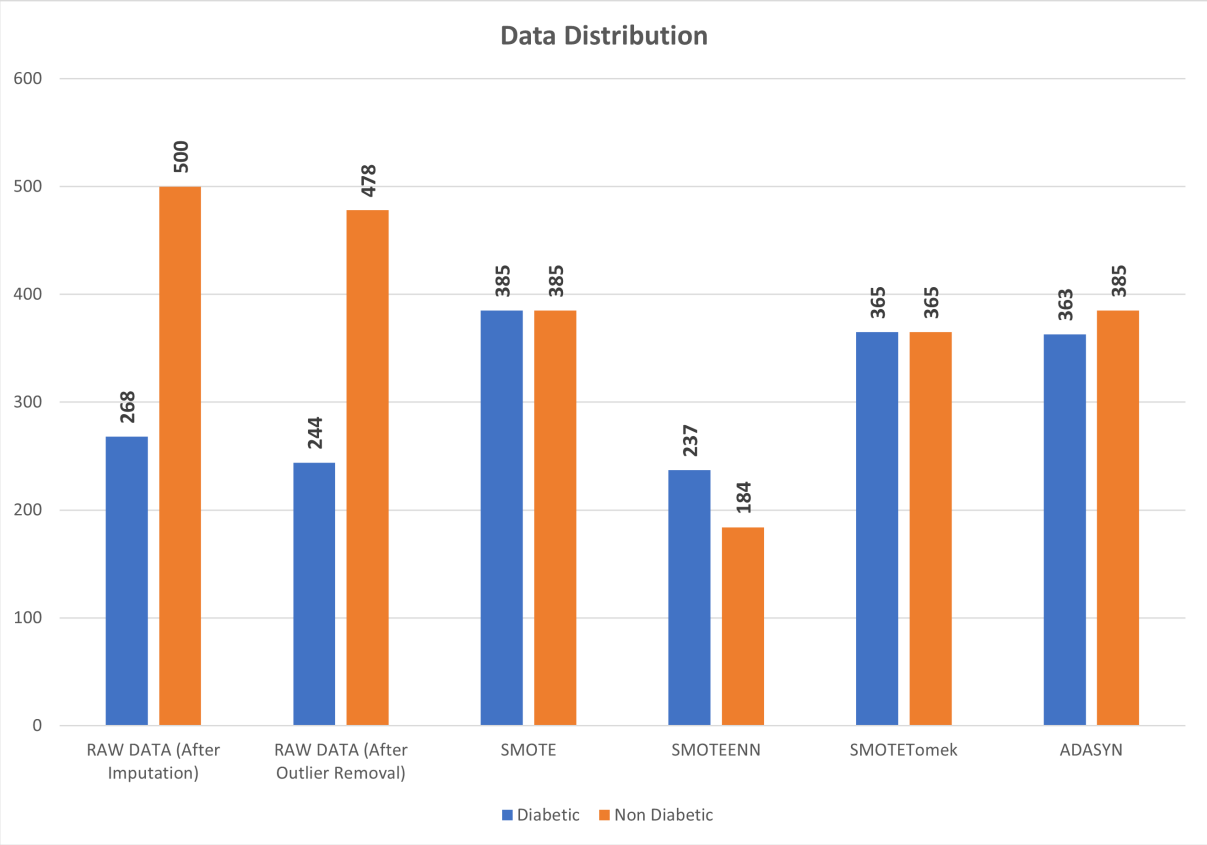


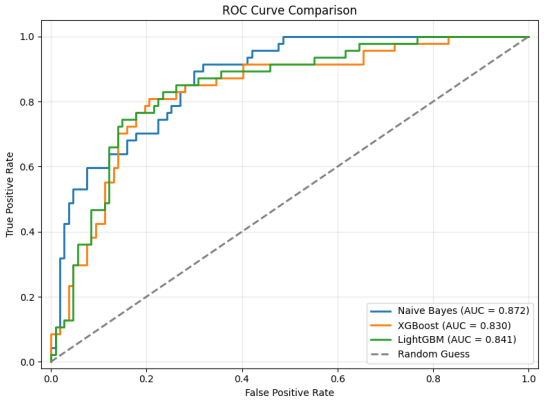
Figure 4.37: Data Distribution for Imputation & Outlier removed with feature selection

Table 4.19: Performance comparison of models with imputation, outliers removed, and with feature selection in the PIMA indian dataset

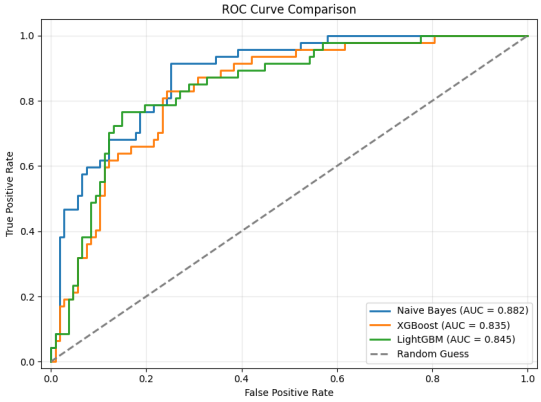
Resampling Technique	ML Model	Accuracy	Precision	Recall	F1-Score	AUC-ROC	MCC
RAW DATA	Naive Bayes	0.7724	**0.7021**	0.6346	0.6667	0.8706	0.4960
	XGBoost	0.7517	0.6667	0.6154	0.6400	0.8356	0.4518
	LightGBM	0.7517	0.6667	0.6154	0.6400	0.8352	0.4518
SMOTE	Naive Bayes	0.7724	0.6610	0.7500	0.7027	0.8726	0.5222
	XGBoost	0.7241	0.6111	0.6346	0.6226	0.8240	0.4056
	LightGBM	0.7448	0.6271	0.7115	0.6667	0.8300	0.4637
SMOTEENN	Naive Bayes	0.7793	0.6515	**0.8269**	0.7288	0.8675	0.5582
	XGBoost	0.7655	0.6406	0.7885	0.7069	0.8565	0.5227
	LightGBM	0.7586	0.6308	0.7885	0.7009	0.8625	0.5115
SMOTETomek	Naive Bayes	0.7724	0.6610	0.7500	0.7027	0.8734	0.5222
	XGBoost	0.7241	0.6071	0.6538	0.6296	0.8230	0.4110
	LightGBM	0.7586	0.6545	0.6923	0.6729	0.8389	0.4824
ADASYN	Naive Bayes	**0.7931**	0.6833	0.7885	**0.7321**	**0.8741**	**0.5688**
	XGBoost	0.7517	0.6379	0.7115	0.6727	0.8406	0.4755
	LightGBM	0.7241	0.6154	0.6154	0.6154	0.8269	0.4003

Table 4.19 demonstrates that the highest performance in accuracy, F1-score, AUC-ROC and MCC were attained with feature selection after imputation and removing the outliers is by the Naive bayes after applying pre-split ADASYN to balance the train data.

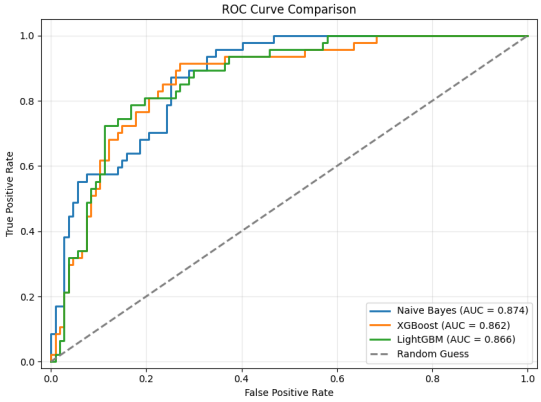
Naive Bayes also got the highest precision with raw data and maximum recall with SMOTEENN. In contrast, LightGBM showed the lowest results in accuracy, recall, F1-score and MCC under ADASYN alongside the same recall value under raw data. XGBoost also displayed lowest results in accuracy with raw data and least precision and AUC-ROC under SMOTETomek.



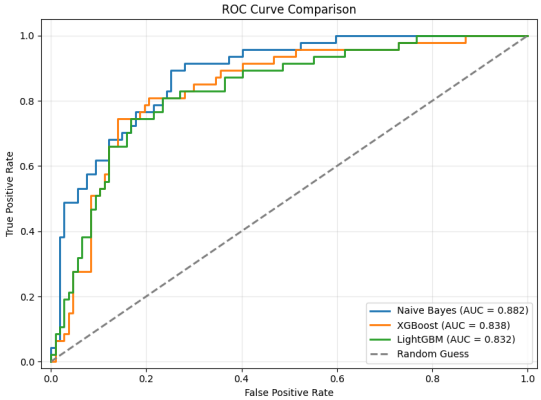
(a) RAW(No Resampling techniques)



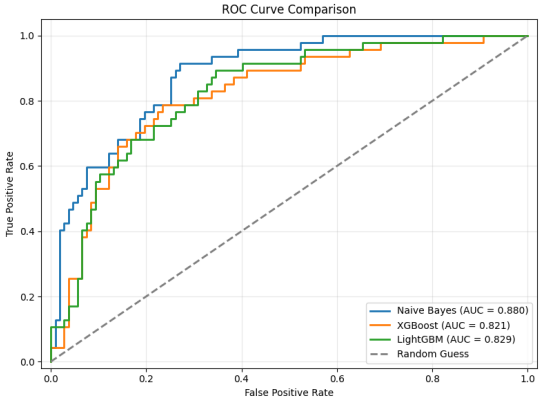
(b) SMOTE



(c) SMOTEENN



(d) ADASYN



(e) SMOTETomek

Figure 4.38: ROC curves demonstrating model performance on the PIMA indian dataset for post-split data resampling Imputation and Outlier removed with feature selection.

Elimination & Outlier removed with feature selection

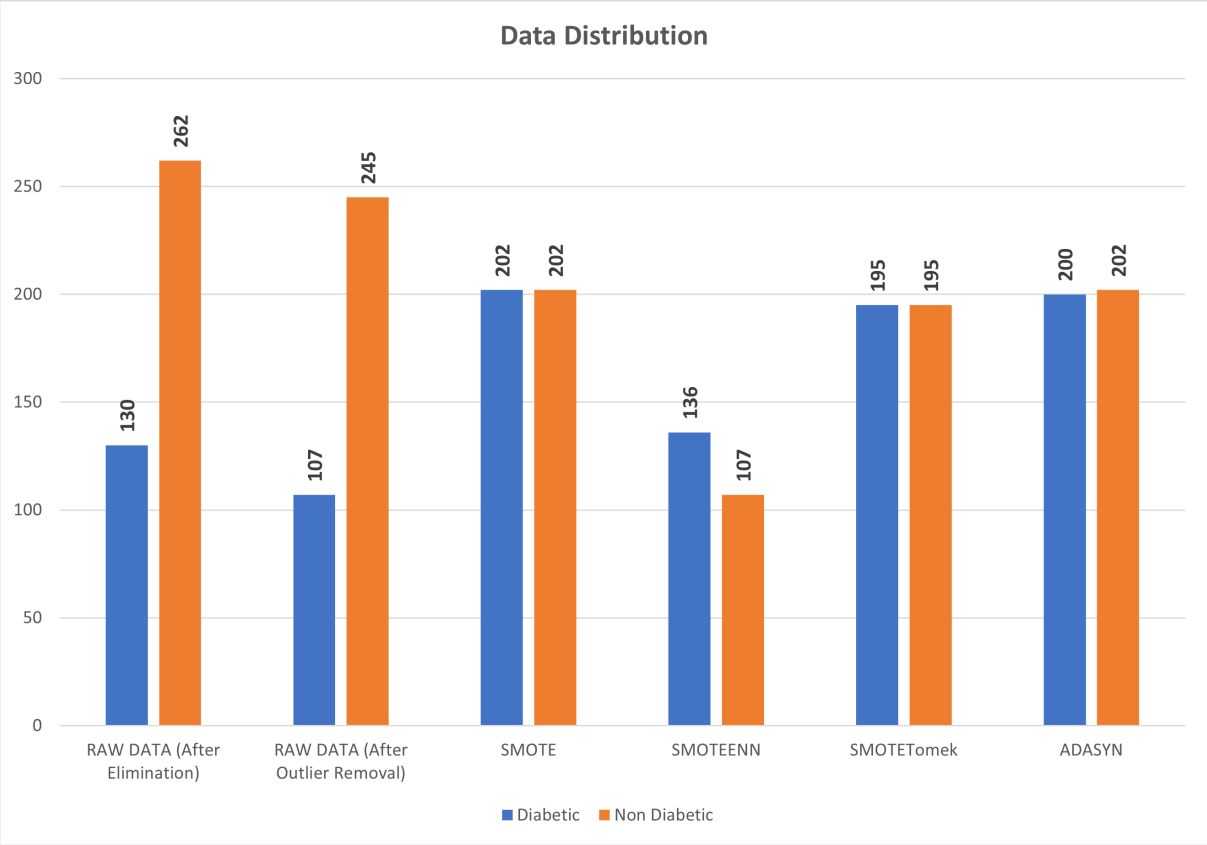


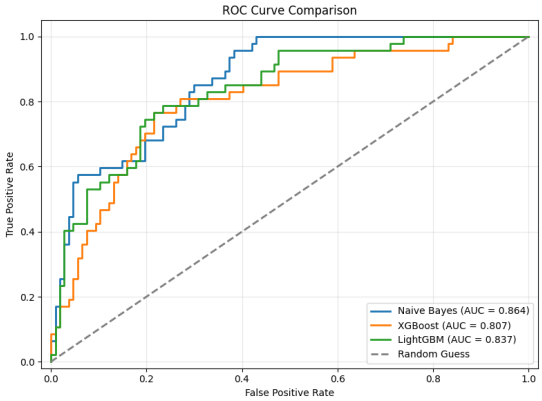
Figure 4.39: Data Distribution for Elimination & Outlier removed with feature selection

Table 4.20: Performance comparison of models with elimination, outliers removed, and with feature selection in the PIMA indian dataset

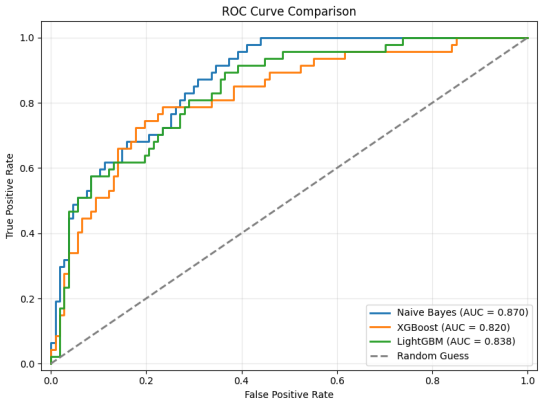
Resampling Technique	ML Model	Accuracy	Precision	Recall	F1-Score	AUC-ROC	MCC
RAW DATA	Naive Bayes	0.7042	0.6667	0.5	0.5714	0.838	0.3611
	XGBoost	**0.7746**	**0.75**	0.6429	**0.6923**	0.8048	**0.52**
	LightGBM	**0.7746**	0.8	0.5714	0.6667	0.8214	0.5198
SMOTE	Naive Bayes	0.6901	0.6154	0.5714	0.5926	0.838	0.3438
	XGBoost	0.7042	0.64	0.5714	0.6038	0.8065	0.3705
	LightGBM	0.7183	0.6538	0.6071	0.6296	0.8148	0.4036
SMOTEENN	Naive Bayes	0.7465	0.6667	**0.7143**	0.6897	**0.8422**	0.4766
	XGBoost	0.7324	0.68	0.6071	0.6415	0.7973	0.4309
	LightGBM	0.7183	0.6667	0.5714	0.6154	0.8264	0.3982
SMOTETomek	Naive Bayes	0.7042	0.6296	0.6071	0.6182	0.8372	0.3771
	XGBoost	0.7183	0.6667	0.5714	0.6154	0.8015	0.3982
	LightGBM	0.7465	0.7273	0.5714	0.64	0.7965	0.4564
ADASYN	Naive Bayes	0.7183	0.6429	0.6429	0.6429	0.8355	0.4103
	XGBoost	0.6901	0.6071	0.6071	0.6071	0.7392	0.3513
	LightGBM	0.7606	0.7037	0.6786	0.6909	0.7841	0.4958

Table 4.20 demonstrates that the highest performance in accuracy, precision, F1-score and MCC were attained with feature selection after elimination and removing the outliers is by the XGBoost with raw data. LightGBM also shows the same highest accuracy with

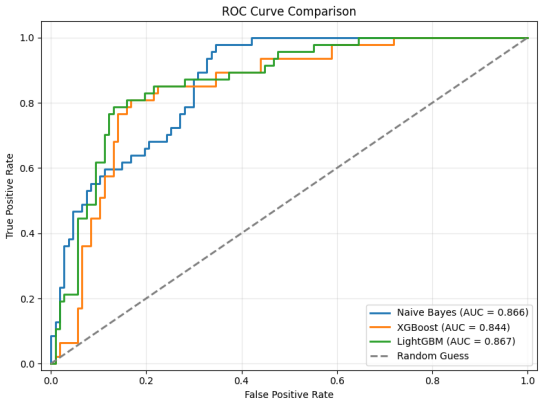
raw data besides Naive Bayes also illustrates the maximum recall and AUC-ROC under SMOTEENN. Conversely, XGBoost also showed the lowest results in accuracy, precision and AUC-ROC under ADASYN. Additionally, Naive Bayes also showed the least recall and F1-score with raw data, and minimum accuracy and MCC after applying SMOTE.



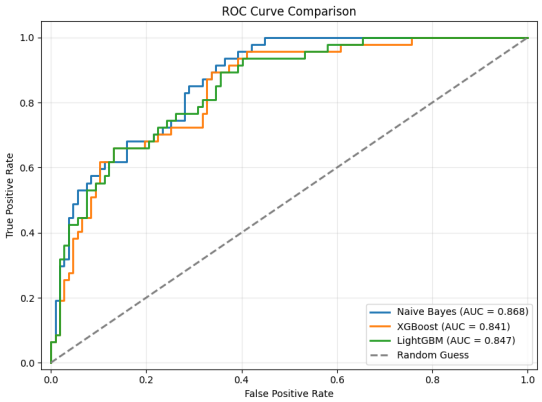
(a) RAW(No Resampling techniques)



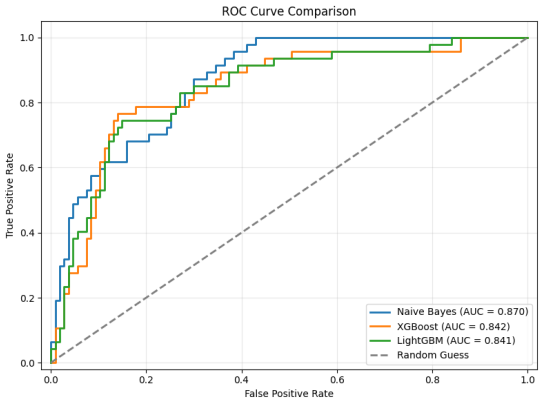
(b) SMOTE



(c) SMOTEENN



(d) ADASYN



(e) SMOTETomek

Figure 4.40: ROC curves demonstrating model performance on the PIMA indian dataset for post-split data resampling Elimination and Outlier removed with feature selection.

Imputation With outlier without feature selection

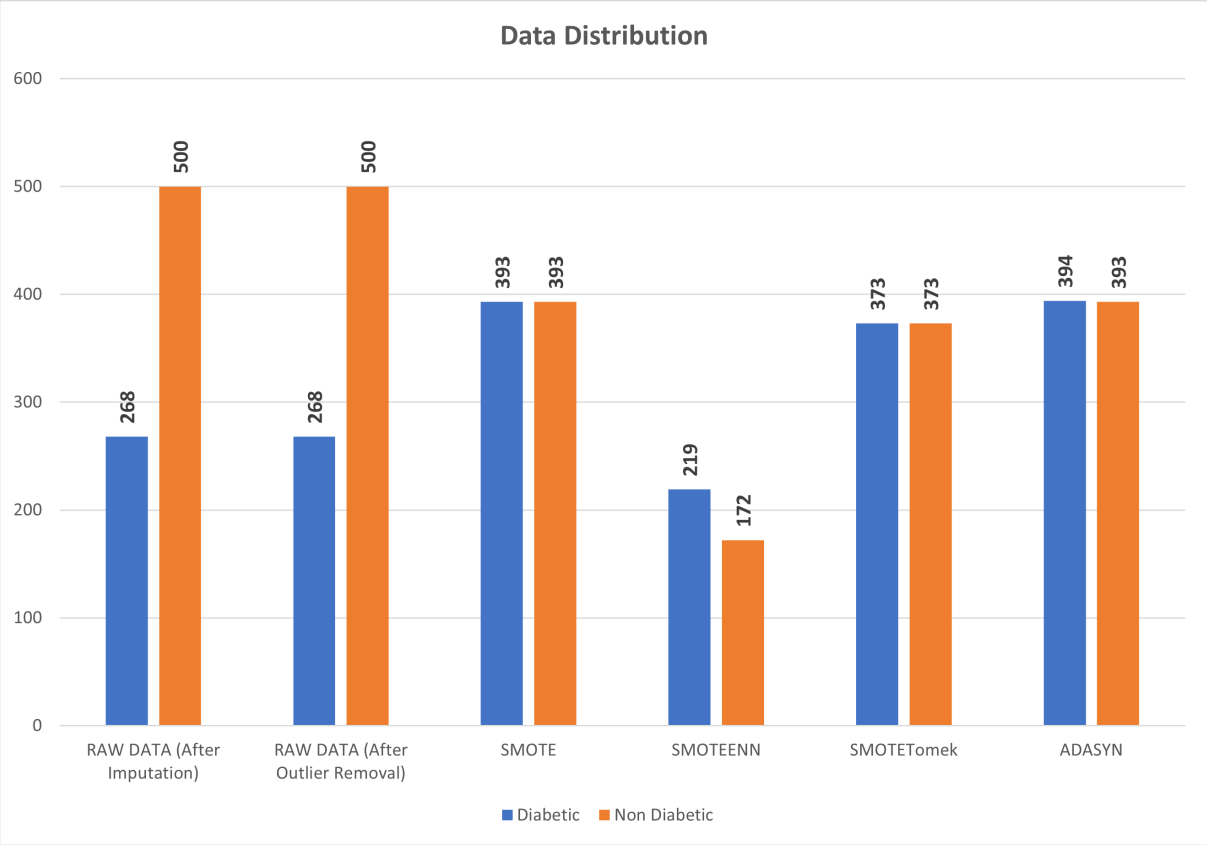


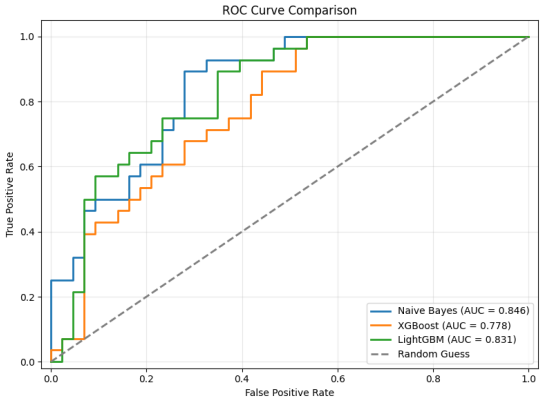
Figure 4.41: Data Distribution for Imputation With outlier without feature selection

Table 4.21: Performance comparison of models with imputation, with outliers, and without feature selection in the PIMA indian dataset

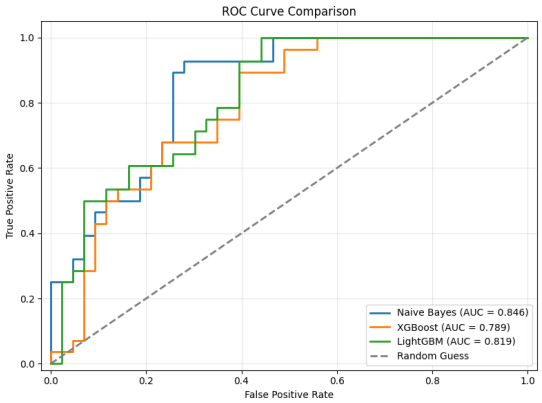
Resampling Technique	ML Model	Accuracy	Precision	Recall	F1-Score	AUC-ROC	MCC
RAW DATA	Naive Bayes	0.7922	0.6667	0.6383	0.6522	0.8721	0.5044
	XGBoost	0.7922	0.6415	0.7234	0.6800	0.8304	0.5291
	LightGBM	0.8052	**0.6809**	0.6809	0.6809	0.8405	0.5407
SMOTE	Naive Bayes	0.7922	0.6316	0.7660	0.6923	**0.8821**	0.5433
	XGBoost	0.7468	0.5690	0.7021	0.6286	0.8352	0.4452
	LightGBM	**0.8117**	0.6800	0.7234	0.7010	0.8445	**0.5644**
SMOTEENN	Naive Bayes	0.7597	0.5694	**0.8723**	0.6891	0.8737	0.5377
	XGBoost	0.7857	0.6129	0.8085	0.6972	0.8624	0.5486
	LightGBM	0.7922	0.6230	0.8085	0.7037	0.8660	0.5589
SMOTETomek	Naive Bayes	0.7857	0.6207	0.7660	0.6857	0.8801	0.5325
	XGBoost	0.7727	0.6034	0.7447	0.6667	0.8212	0.5034
	LightGBM	0.7597	0.5862	0.7234	0.6476	0.8386	0.4743
ADASYN	Naive Bayes	0.7792	0.6066	0.7872	0.6852	**0.8821**	0.5300
	XGBoost	0.7987	0.6379	0.7872	**0.7048**	0.8383	0.5616
	LightGBM	0.7727	0.6000	0.7660	0.6729	0.8324	0.5115

Table 4.21 demonstrates that no specific model showed the maximum results for most of the evaluation metrics. Here, LightGBM displayed the highest precision with raw data, and maximum accuracy and MCC under SMOTE. Similarly, Naive Bayes revealed the

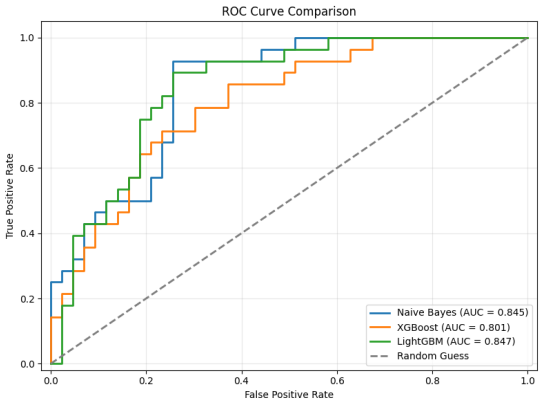
highest recall with SMOTEENN and the maximum AUC-ROC under ADASYN. While XGBoost gave the highest F1-score only under ADASYN. On the other hand, XGBoost showed the lowest accuracy, precision, F1-score and MCC after SMOTE resampling, it also gave the least recall with raw data besides the lowest AUC-ROC after SMOTETomek.



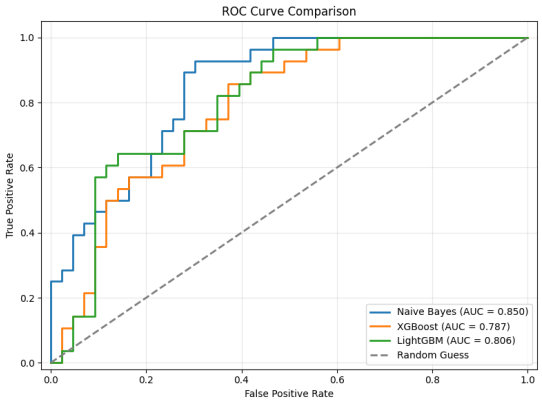
(a) RAW(No Resampling techniques)



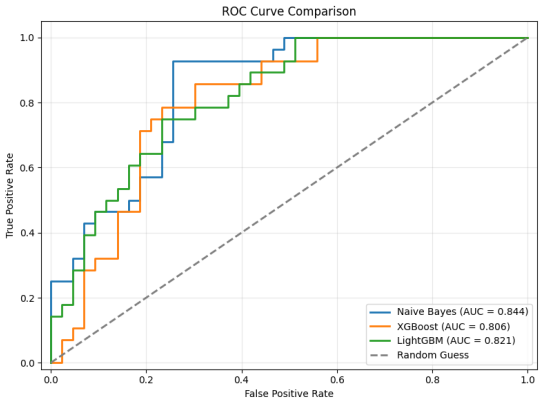
(b) SMOTE



(c) SMOTEENN



(d) ADASYN



(e) SMOTETomek

Figure 4.42: ROC curves demonstrating model performance on the PIMA indian dataset for post-split data resampling Imputation With outlier without feature selection.

Elimination With outlier without feature selection

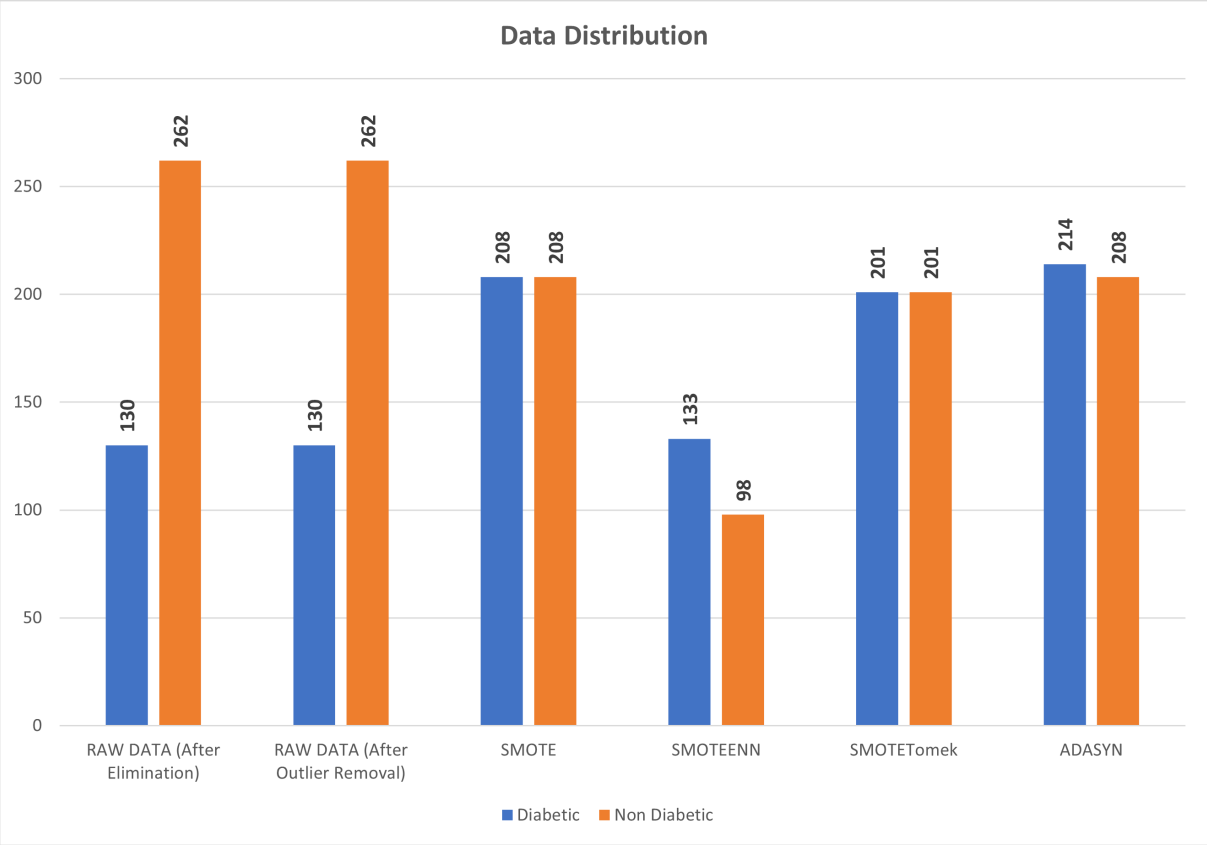


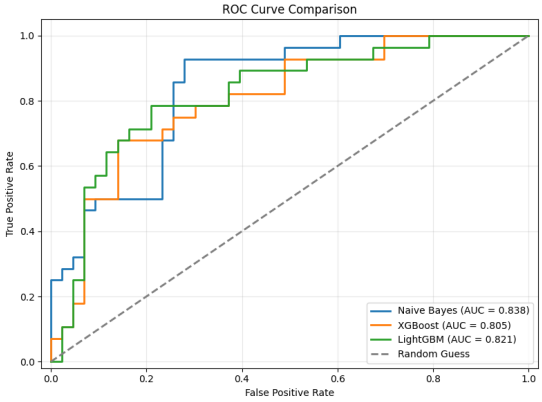
Figure 4.43: Data Distribution for Elimination With outlier without feature selection

Table 4.22: Performance comparison of models with elimination, with outliers, and without feature selection in the PIMA indian dataset

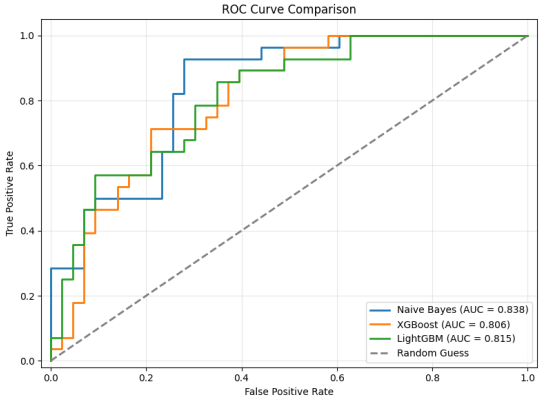
Resampling Technique	ML Model	Accuracy	Precision	Recall	F1-Score	AUC-ROC	MCC
RAW DATA	Naive Bayes	0.8228	0.72	0.72	0.72	0.8911	0.5904
	XGBoost	0.7975	0.7143	0.6	0.6522	0.8689	0.5147
	LightGBM	0.7848	0.7222	0.52	0.6047	0.8793	0.4739
SMOTE	Naive Bayes	0.8101	0.6923	0.72	0.7059	0.8904	0.566
	XGBoost	0.8228	0.72	0.72	0.72	0.877	0.5904
	LightGBM	0.8228	**0.7619**	0.64	0.6957	0.8689	0.5763
SMOTEENN	Naive Bayes	0.7595	0.5938	**0.76**	0.6667	0.8696	0.492
	XGBoost	0.7722	0.6207	0.72	0.6667	0.8681	0.4982
	LightGBM	0.7595	0.6071	0.68	0.6415	0.8422	0.4631
SMOTETomek	Naive Bayes	0.8101	0.6923	0.72	0.7059	0.8874	0.566
	XGBoost	0.7975	0.6957	0.64	0.6667	0.8756	0.5225
	LightGBM	0.7975	0.6957	0.64	0.6667	0.8733	0.5225
ADASYN	Naive Bayes	0.7975	0.6552	**0.76**	0.7037	**0.8963**	0.5546
	XGBoost	0.8228	0.72	0.72	0.72	0.8807	0.5904
	LightGBM	**0.8354**	0.7308	**0.76**	**0.7451**	0.6239	**0.6239**

Table 4.22 illustrates that LightGBM showed the highest performance in accuracy, recall, F1-score and MCC value under ADASYN beside giving the maximum precision under SMOTE. Furthermore, Naive Bayes exhibited the highest recall with SMOTEENN, and

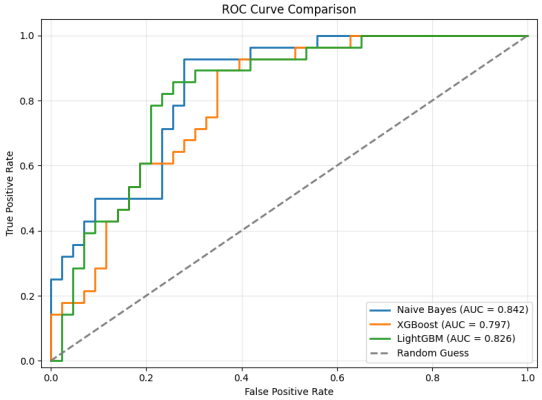
the maximum recall and AUC-ROC together under ADASYN. Conversely, Naive bayes gave the lowest accuracy and precision after SMOTEENN resampling beside showing the lowest recall with raw data. LightGBM also gave the lowest F1-score with raw data, the lowest accuracy and MCC under SMOTEENN, and the lowest AUC-ROC for ADASYN resampling.



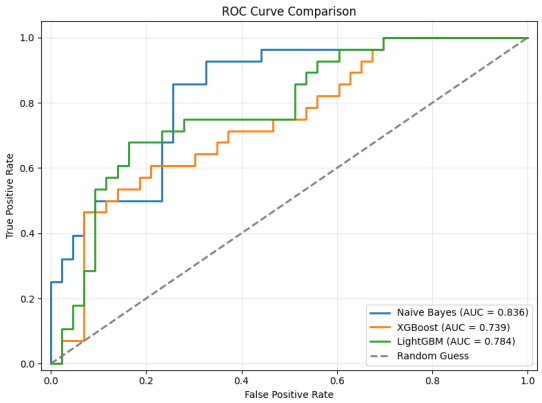
(a) RAW(No Resampling techniques)



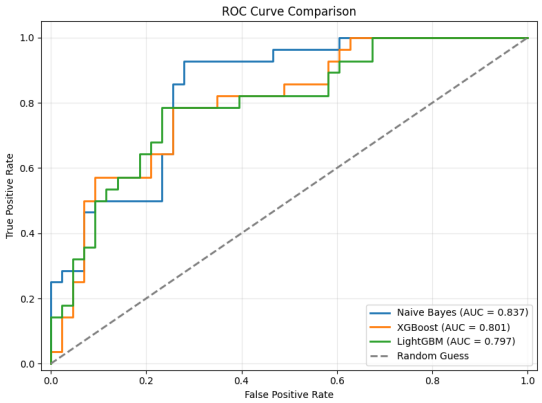
(b) SMOTE



(c) SMOTEENN



(d) ADASYN



(e) SMOTETomek

Figure 4.44: ROC curves demonstrating model performance on the PIMA indian dataset for post-split data resampling Elimination With outlier without feature selection.

Imputation With outlier with feature selection

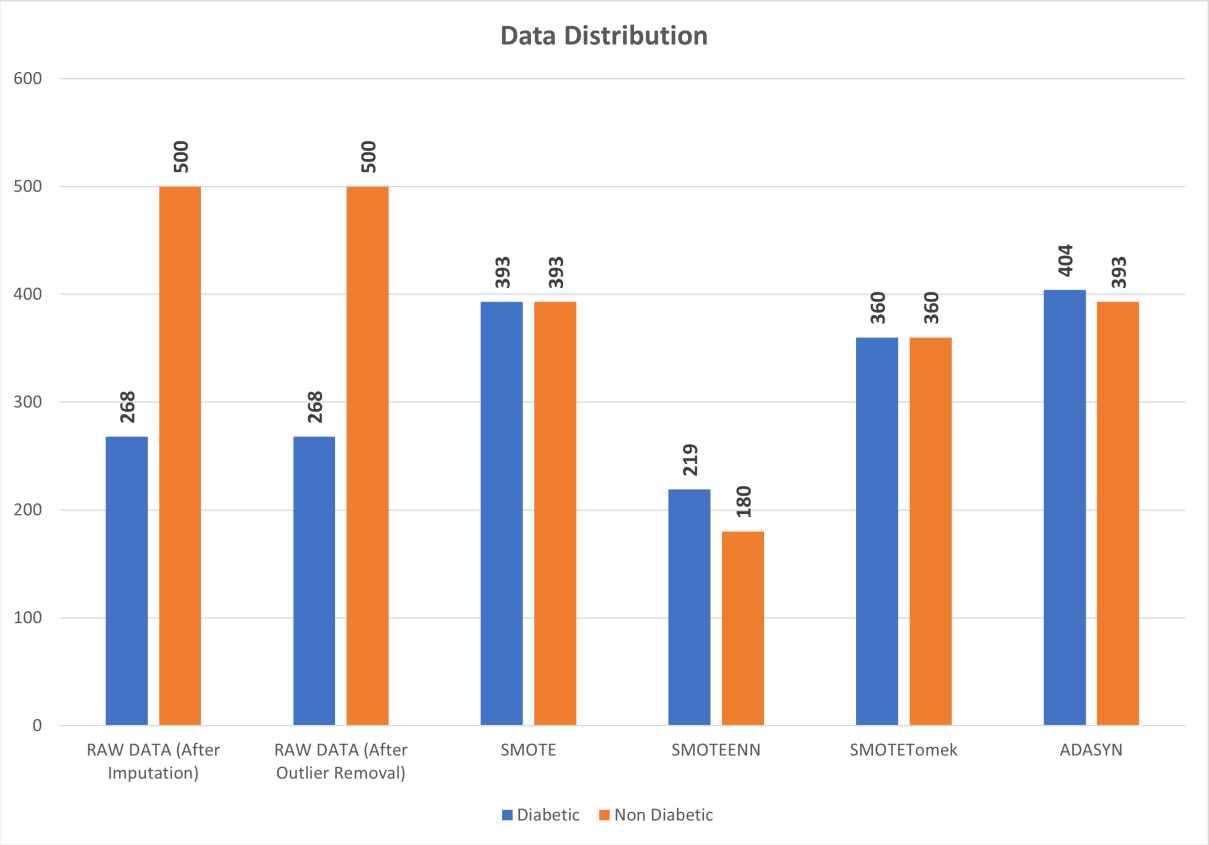


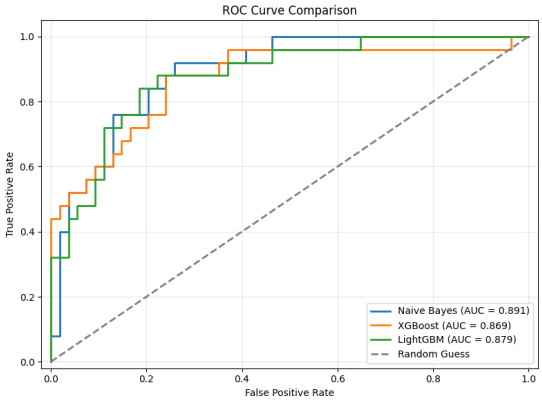
Figure 4.45: Data Distribution for Imputation With outlier with feature selection

Table 4.23: Performance comparison of models with imputation, with outliers, and with feature selection in the PIMA indian dataset

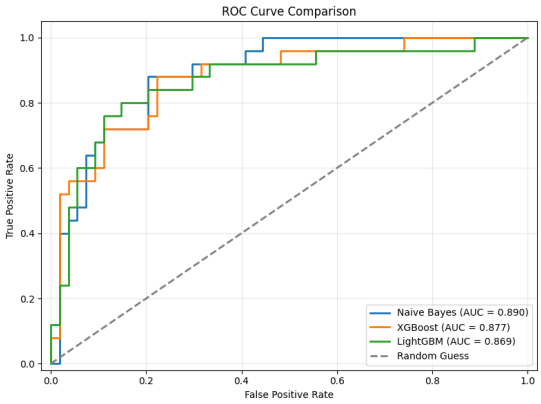
Resampling Technique	ML Model	Accuracy	Precision	Recall	F1-Score	AUC-ROC	MCC
RAW DATA	Naive Bayes	0.7922	**0.6829**	0.5957	0.6364	0.864	0.4941
	XGBoost	0.7727	0.6304	0.617	0.6237	0.8067	0.4609
	LightGBM	0.7532	0.5918	0.617	0.6042	0.8365	0.4252
SMOTE	Naive Bayes	0.7662	0.6	0.7021	0.6471	**0.8696**	0.4772
	XGBoost	0.7727	0.6034	0.7447	0.6667	0.8204	0.5034
	LightGBM	0.7468	0.5769	0.6383	0.6061	0.8381	0.4213
SMOTEENN	Naive Bayes	0.7338	0.5441	0.7872	0.6435	0.8656	0.4614
	XGBoost	**0.7987**	0.629	**0.8298**	0.7156	0.8437	**0.5773**
	LightGBM	0.8117	0.65	**0.8298**	**0.729**	0.8674	0.5982
SMOTETomek	Naive Bayes	0.7662	0.6	0.7021	0.6471	0.8698	0.4772
	XGBoost	0.8052	0.6545	0.766	0.7059	0.8417	0.5655
	LightGBM	**0.7987**	0.6481	0.7447	0.6931	0.8405	0.5473
ADASYN	Naive Bayes	0.7403	0.5556	0.7447	0.6364	0.8682	0.4524
	XGBoost	0.7403	0.5574	0.7234	0.6296	0.8409	0.4435
	LightGBM	0.7532	0.5818	0.6809	0.6275	0.8471	0.4477

Table 4.23 illustrates that XGBoost showed the highest performance in accuracy, recall and MCC value under SMOTEENN. Naive Bayes also gave the maximum precision with raw data and highest AUC-ROC with SMOTE. Similarly, LightGBM displayed the max-

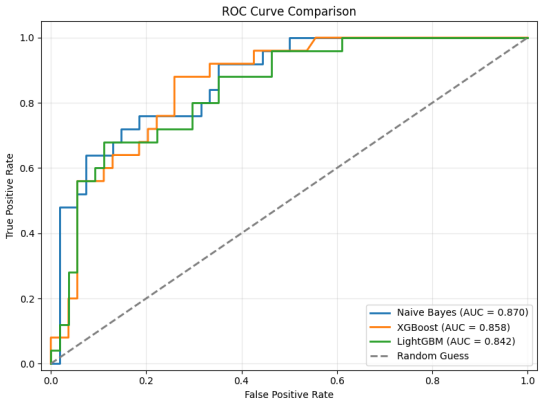
imum recall and F1-score under SMOTEENN, and the highest accuracy after SMOTE-Tomek implementation. In contrast, Naive Bayes showed the least accuracy and precision after SMOTEENN, besides showing the lowest recall with raw data. LightGBM showed the lowest F1-score with raw data, and the least MCC under SMOTE. Lastly, XGBoost revealed the lowest AUC-ROC with raw data.



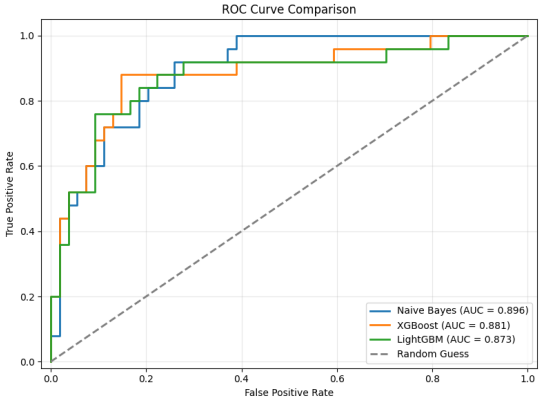
(a) RAW(No Resampling techniques)



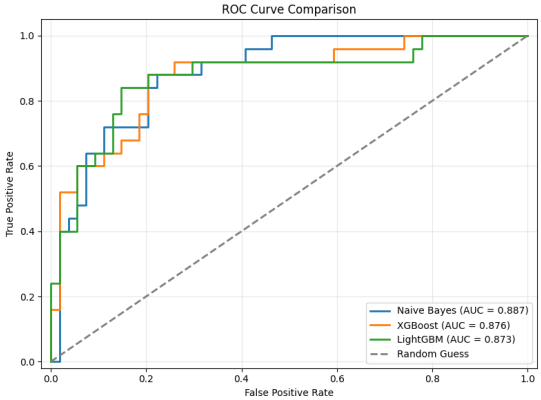
(b) SMOTE



(c) SMOTEENN



(d) ADASYN



(e) SMOTETomek

Figure 4.46: ROC curves demonstrating model performance on the PIMA indian dataset for post-split data resampling Imputation With outlier with feature selection.

Elimination With outlier with feature selection

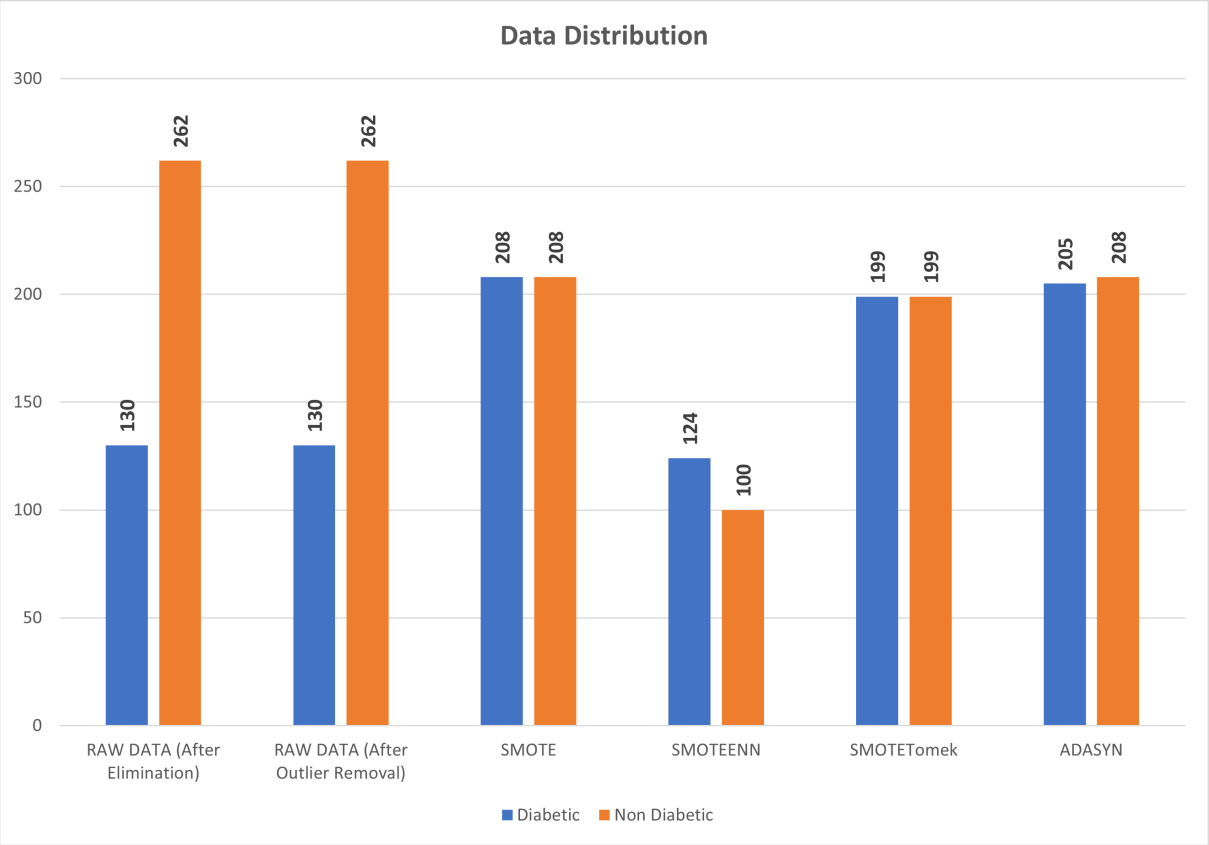


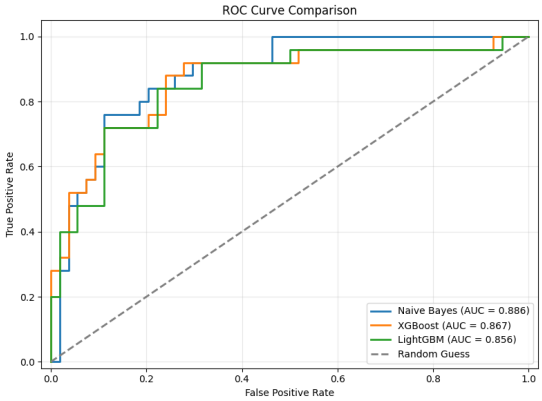
Figure 4.47: Data Distribution for Elimination With outlier with feature selection

Table 4.24: Performance comparison of models with elimination, with outliers, and with feature selection in the PIMA indian dataset

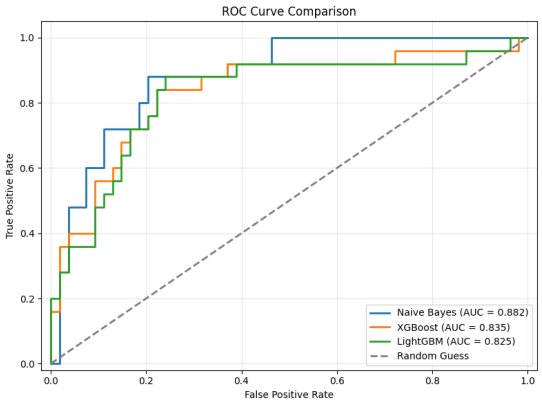
Resampling Technique	ML Model	Accuracy	Precision	Recall	F1-Score	AUC-ROC	MCC
RAW DATA	Naive Bayes	0.8101	0.7273	0.64	0.6809	**0.8859**	0.5488
	XGBoost	**0.8354**	**0.75**	0.72	0.7347	0.8667	**0.6158**
	LightGBM	0.8228	0.72	0.72	0.72	0.8556	0.5904
SMOTE	Naive Bayes	0.8101	0.6923	0.72	**0.7059**	0.8822	0.566
	XGBoost	0.7848	0.6667	0.64	0.6531	0.8348	0.4974
	LightGBM	0.7848	0.6429	0.72	0.6792	0.8252	0.52
SMOTEENN	Naive Bayes	0.7975	0.6552	**0.76**	0.7037	0.8659	0.5546
	XGBoost	0.8101	0.6923	0.72	**0.7059**	0.8659	0.566
	LightGBM	0.7342	0.5769	0.6	0.5882	0.843	0.3922
SMOTETomek	Naive Bayes	0.8101	0.6923	0.72	**0.7059**	0.8807	0.566
	XGBoost	0.7722	0.6207	0.72	0.6667	0.8333	0.4982
	LightGBM	0.7848	0.6333	**0.76**	0.6909	0.8415	0.5331
ADASYN	Naive Bayes	0.7848	0.6429	0.72	0.6792	0.8852	0.52
	XGBoost	0.7975	0.6552	**0.76**	0.7037	0.8185	0.5546
	LightGBM	0.7722	0.6129	**0.76**	0.6786	0.8281	0.5122

Table 4.24 illustrates that XGBoost showed the highest performance in accuracy, precision and MCC value with raw data, beside giving the maximum F1-score under SMOTEE and best recall under ADASYN. Moreover, Naive Bayes exhibited the highest recall with

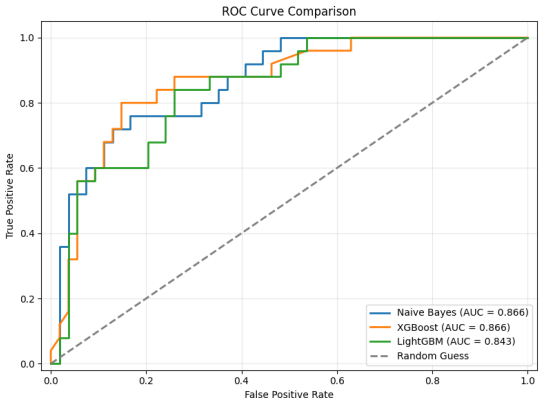
SMOTEENN, and the best F1-score under both SMOTE and SMOTETomek. LightGBM showed the maximum recall after SMOTETomek resampling. LightGBM mostly showed the lowest performance result under SMOTEEn across all the evaluation metrics exceptt AUC-ROC. The least AUC-ROC was demonstrated by XGBoost after ADASYN.



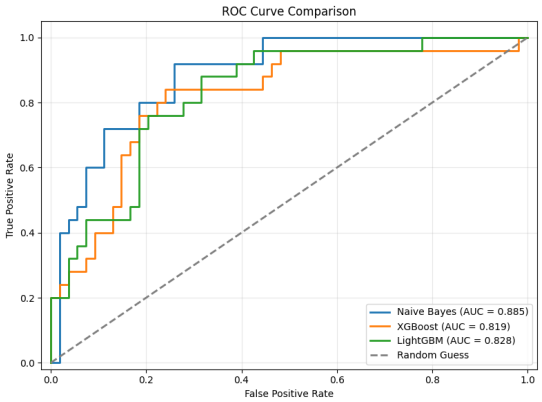
(a) RAW(No Resampling techniques)



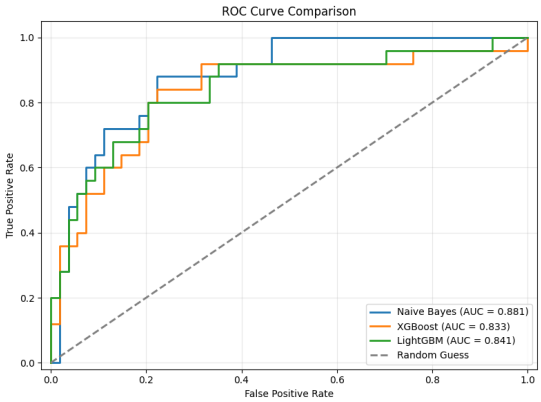
(b) SMOTE



(c) SMOTEENN



(d) ADASYN



(e) SMOTETomek

Figure 4.48: ROC curves demonstrating model performance on the PIMA indian dataset for post-split data resampling Elimination With outlier with feature selection.

Looking at the data distribution for the post-split PIMA indian dataset across all eight circumstances whether missing values were handled through imputation or elimination, outliers were removed or kept, and with or without feature selection the visuals consistently highlight the original class imbalance found in the raw data. More importantly,

they effectively show how employing the four resampling methods (SMOTE, SMOTEENN, SMOTETomek, and ADASYN) successfully levels out this imbalance.

When evaluating the ROC curves across the different scenarios for this dataset, a consistent performance trend is clear for the Naive Bayes, XGBoost, and LightGBM models. Interestingly, in these post-split setups, the raw data actually shows consistently higher AUC-ROC scores than the resampling methods, which refers to an improved True Positive Rate (Recall). Also, the models incorporating the resampling techniques generally follow a similar pattern to one another. On the other hand, SMOTEENN and ADASYN generally fall a bit short, showing ROC curves that are still to some degree lower than the rest across the various setups.

4.2.2 BRFSS-2015 Prediction Result

The results obtained from applying resampling methods after the train and test split of the BRFSS-2015 dataset in four different circumstances are illustrated here.

Outlier removed without feature selection

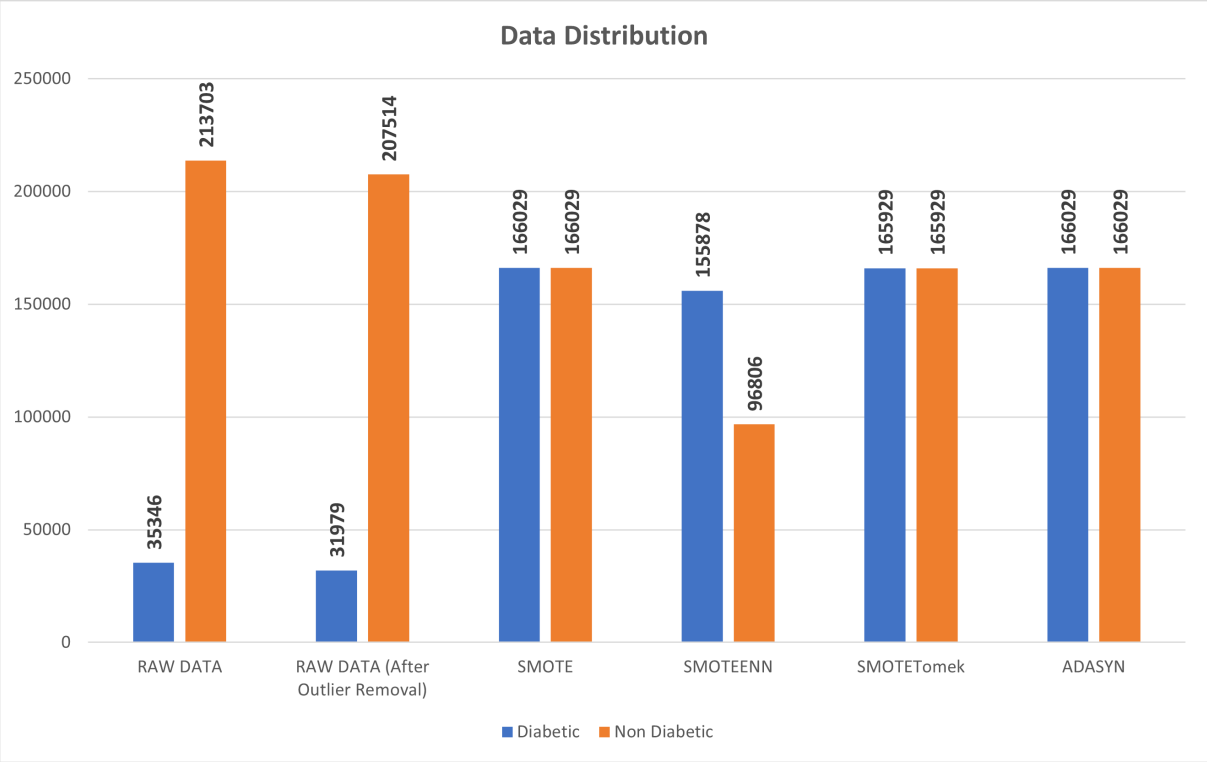
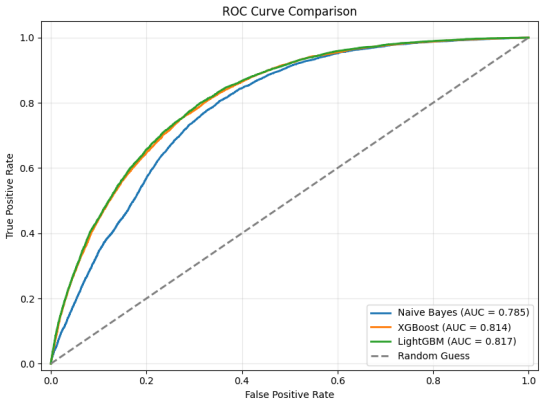


Figure 4.49: Data Distribution for Outlier removed without feature selection

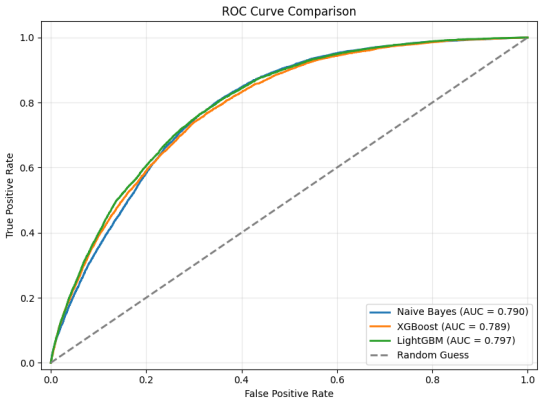
Table 4.25: Performance comparison of models after outlier removal and without feature selection in the BRFSS dataset

Resampling Technique	ML Model	Accuracy	Precision	Recall	F1-Score	AUC-ROC	MCC
RAW DATA	Naive Bayes	0.7911	0.3451	0.4445	0.3886	0.7849	0.2681
	XGBoost	0.8550	0.5482	0.1663	0.2552	0.8138	0.2438
	LightGBM	**0.8556**	**0.5599**	0.1546	0.2423	**0.8168**	0.2389
SMOTE	Naive Bayes	0.7285	0.3160	0.7031	0.4361	0.7898	0.3299
	XGBoost	0.7517	0.3285	0.6347	0.4329	0.7890	0.3201
	LightGBM	0.7577	0.3362	0.6394	0.4407	0.7970	0.3302
SMOTEENN	Naive Bayes	0.6994	0.2974	0.7435	0.4249	0.7778	0.3207
	XGBoost	0.6396	0.2731	0.8507	0.4135	0.8040	0.3239
	LightGBM	0.6457	0.2768	**0.8513**	0.4178	0.8075	0.3296
SMOTETomek	Naive Bayes	0.7285	0.3160	0.7028	0.4360	0.7898	0.3298
	XGBoost	0.7520	0.3282	0.6314	0.4319	0.7881	0.3187
	LightGBM	0.7572	0.3361	0.6424	**0.4413**	0.7881	**0.3312**
ADASYN	Naive Bayes	0.7158	0.3083	0.7264	0.4329	0.7881	0.3286
	XGBoost	0.7330	0.3121	0.6545	0.4227	0.7799	0.3084
	LightGBM	0.7498	0.3278	0.6427	0.4341	0.7915	0.3221

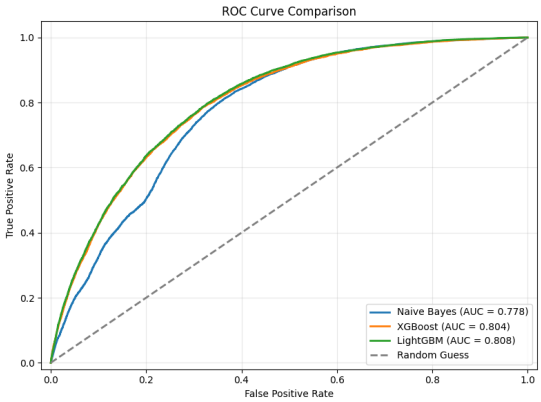
Table 4.25 demonstrates that the highest performance in accuracy, precision and AUC-ROC attained without feature selection after removing the outliers is by the LightGBM models with raw data besides the maximum recall with SMOTEENN and highest F1-score with SMOTETomek is shown by the same model. In contrast, LightGBM also showed the least outcomes for recall, F1-score and MCC with the raw data. Additionally, under SMOTEENN resampling Naive Bayes displayed least AUC-ROC, XGBoost illustrated lowest precision and LightGBM showed the minimum accuracy.



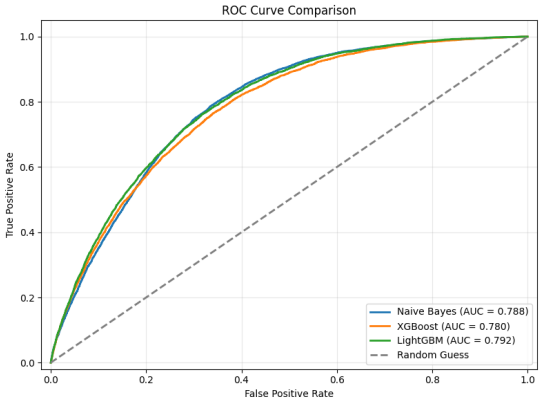
(a) RAW(No Resampling techniques)



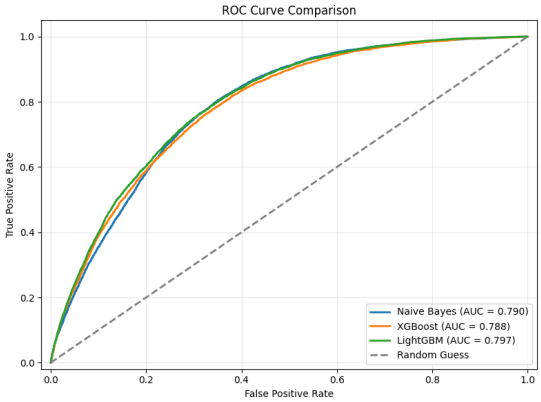
(b) SMOTE



(c) SMOTEENN



(d) ADASYN



(e) SMOTETomek

Figure 4.50: ROC curves demonstrating model performance on the BRFSS dataset for post-split data resampling Outlier removed without feature selection.

Outlier removed with feature selection

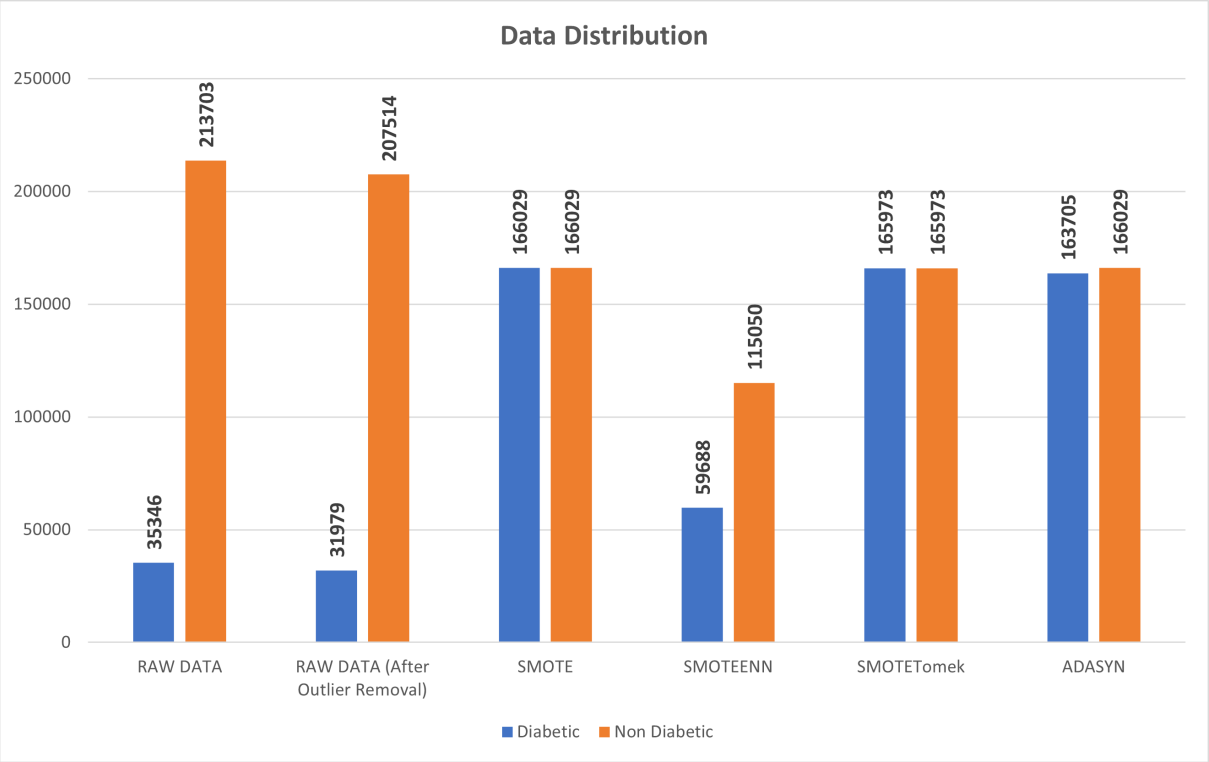


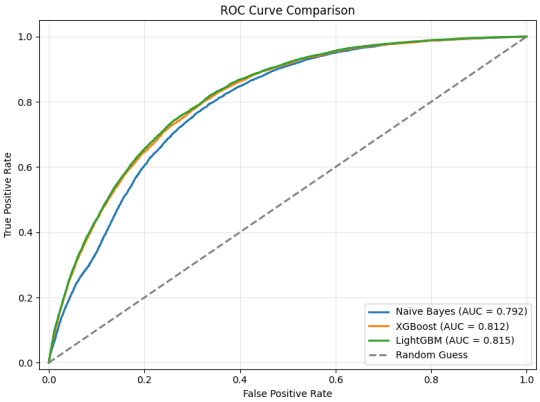
Figure 4.51: Data Distribution for Outlier removed with feature selection

Table 4.26: Performance comparison of models after outlier removal and with feature selection in the BRFSS dataset

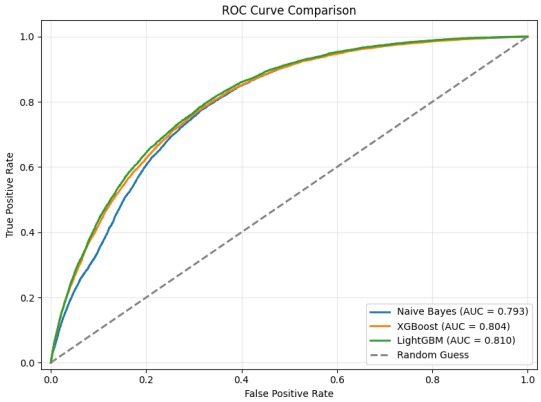
Resampling Technique	ML Model	Accuracy	Precision	Recall	F1-Score	AUC-ROC	MCC
RAW DATA	Naive Bayes	0.8051	0.3672	0.4219	0.3926	0.7922	0.2781
	XGBoost	0.8542	0.5418	0.1538	0.2396	0.8117	0.2317
	LightGBM	**0.8552**	**0.5585**	0.1454	0.2308	**0.8153**	0.2309
SMOTE	Naive Bayes	0.7351	0.3223	0.7016	0.4417	0.7930	0.3368
	XGBoost	0.7194	0.3146	0.7457	0.4425	0.8036	0.3432
	LightGBM	0.7101	0.3099	0.7672	0.4415	0.8099	0.3451
SMOTEENN	Naive Bayes	0.7612	0.3343	0.6045	0.4305	0.7873	0.3156
	XGBoost	0.7819	0.3557	0.5677	0.4374	0.7875	0.3239
	LightGBM	0.7853	0.3618	0.5735	**0.4437**	0.7947	0.3320
SMOTETomek	Naive Bayes	0.7351	0.3223	0.7016	0.4417	0.7930	0.3368
	XGBoost	0.7184	0.3134	0.7443	0.4411	0.8036	0.3413
	LightGBM	0.7104	0.3104	0.7692	0.4423	0.8101	**0.3464**
ADASYN	Naive Bayes	0.7191	0.3127	0.7356	0.4389	0.7925	0.3374
	XGBoost	0.6960	0.2987	0.7689	0.4303	0.7977	0.3312
	LightGBM	0.6874	0.2967	**0.7980**	0.4326	0.8060	0.3388

Table 4.26 signifies that the highest performance in accuracy, precision and AUC-ROC attained with feature selection after removing the outliers is by the LightGBM models with raw data besides the maximum recall with ADASYN, highest F1-score with SMOTEENN and maximum MCC value with SMOTETomek is shown by the same model. In contrast, LightGBM also showed the least outcomes for recall, F1-score and MCC with the raw

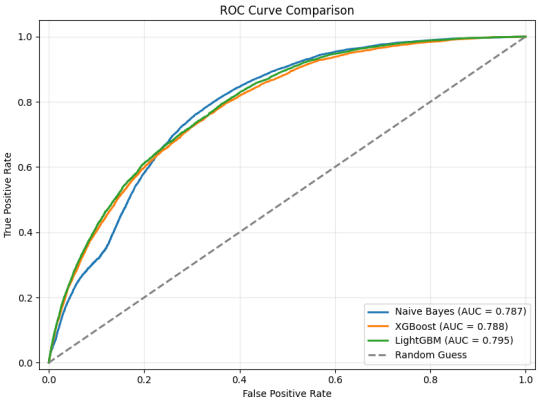
data and lowest accuracy and precision with ADASYN. Additionally, under SMOTEENN resampling Naive Bayes displayed least AUC-ROC.



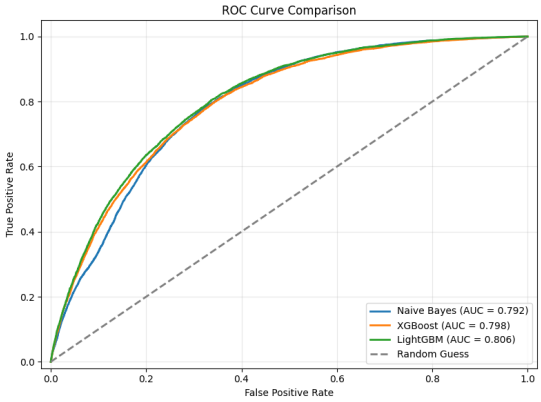
(a) RAW(No Resampling techniques)



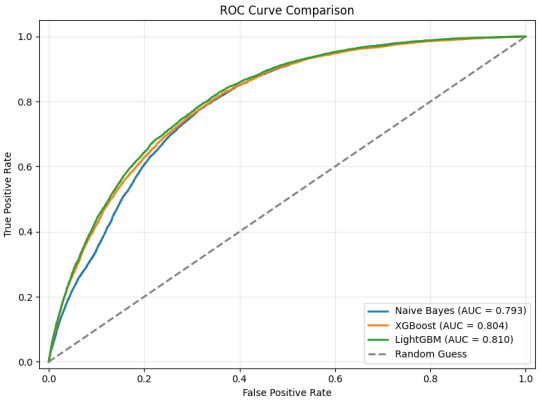
(b) SMOTE



(c) SMOTEENN



(d) ADASYN



(e) SMOTETomek

Figure 4.52: ROC curves demonstrating model performance on the BRFSS dataset for post-split data resampling Outlier removed with feature selection.

With outlier without feature selection

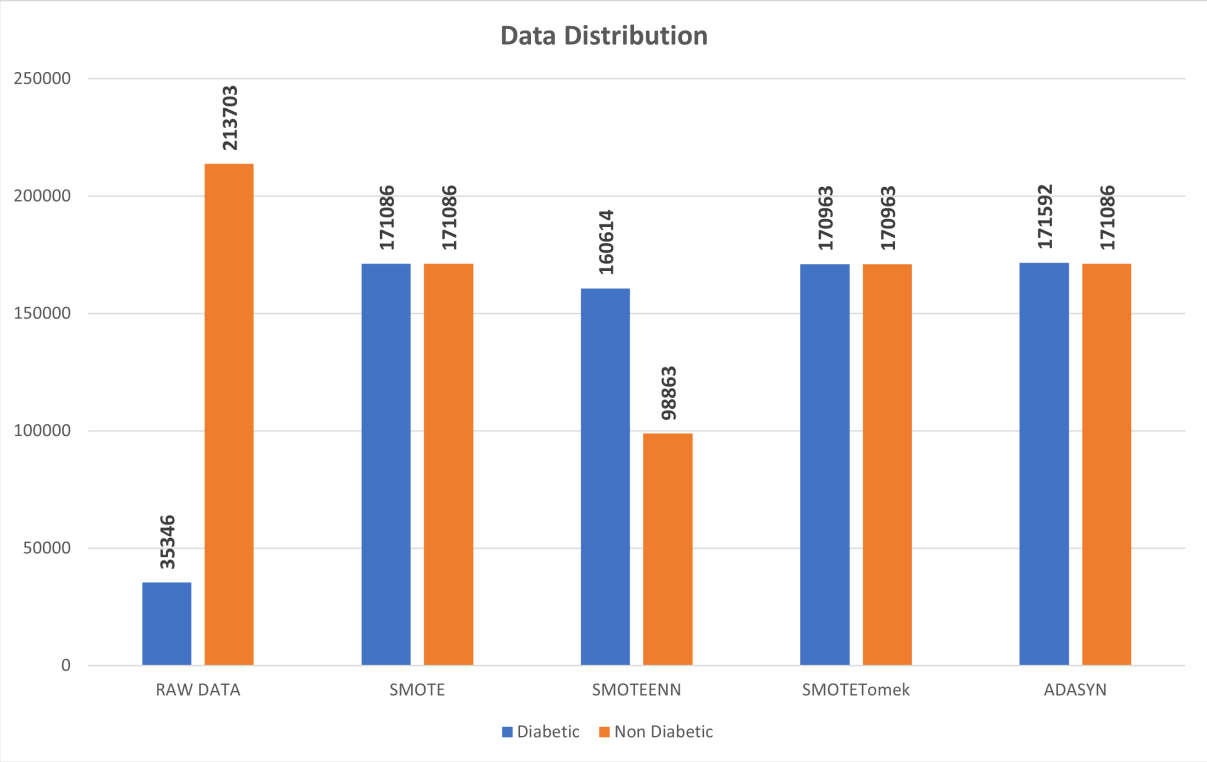


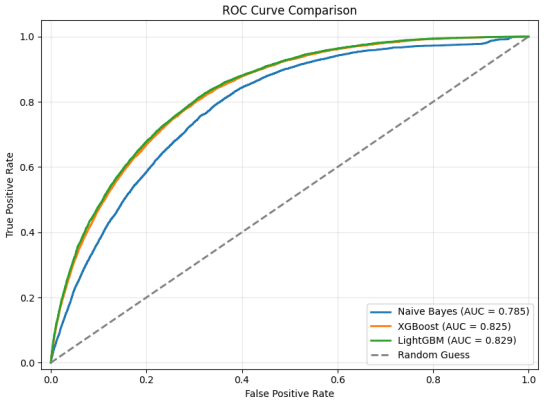
Figure 4.53: Data Distribution for With outlier without feature selection

Table 4.27: Performance comparison of models with outliers and without feature selection in the BRFSS dataset

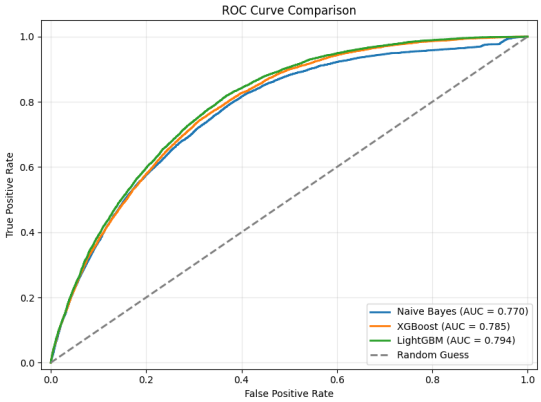
Resampling Technique	ML Model	Accuracy	Precision	Recall	F1-Score	AUC-ROC	MCC
RAW DATA	Naive Bayes	0.7701	0.3619	0.5720	0.4433	0.7849	0.3204
	XGBoost	0.8502	0.5909	0.2085	0.3083	0.8251	0.2875
	LightGBM	**0.8517**	**0.6112**	0.2011	0.3027	**0.8294**	0.2901
SMOTE	Naive Bayes	0.6535	0.2878	0.7900	0.4218	0.7697	0.3084
	XGBoost	0.7583	0.3498	0.5944	0.4404	0.7847	0.3164
	LightGBM	0.7588	0.3547	0.6194	0.4511	0.7939	0.3307
SMOTEENN	Naive Bayes	0.6561	0.2899	0.7926	0.4245	0.7725	0.3124
	XGBoost	0.6658	0.3021	0.8308	0.4431	0.8091	0.3436
	LightGBM	0.6623	0.3016	**0.8444**	0.4445	**0.8145**	**0.3480**
SMOTETomek	Naive Bayes	0.6535	0.2878	0.7899	0.4219	0.7697	0.3084
	XGBoost	0.7583	0.3507	0.5996	0.4426	0.7846	0.3192
	LightGBM	0.7602	0.3567	0.6206	**0.4530**	0.7937	0.3332
ADASYN	Naive Bayes	0.6255	0.2742	0.8139	0.4102	0.7626	0.2963
	XGBoost	0.7477	0.3400	0.6133	0.4375	0.7768	0.3127
	LightGBM	0.7506	0.3456	0.6256	0.4453	0.7873	0.3233

Table 4.27 indicates the highest performance in accuracy, precision and AUC-ROC attained without feature selection with the outliers is by the LightGBM models with raw data besides the maximum recall and F1-score with SMOTEENN and maximum MCC value with SMOTETomek is shown by the same model. In contrast, LightGBM also showed the least outcomes for recall and F1-score along with lowest MCC by XGBoost

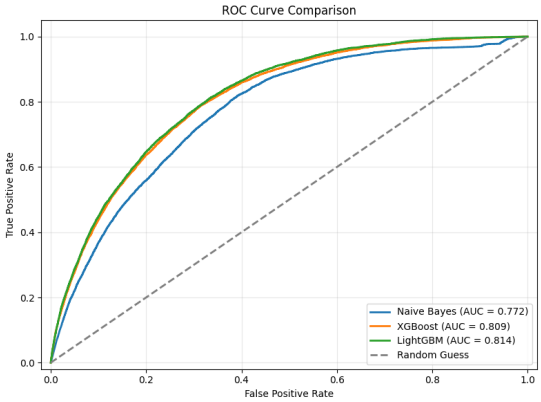
with the raw data. Additionally, LightGBM also illustrated the lowest accuracy and precision, and least AUC-ROC by Naive Bayes with ADASYN.



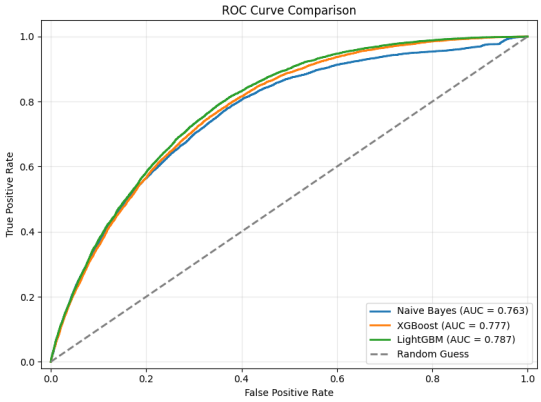
(a) RAW(No Resampling techniques)



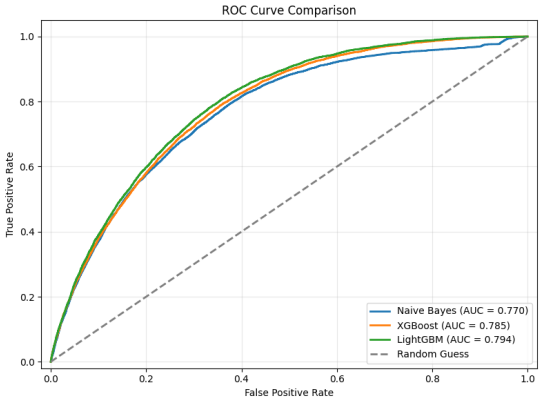
(b) SMOTE



(c) SMOTEENN



(d) ADASYN



(e) SMOTETomek

Figure 4.54: ROC curves demonstrating model performance on the BRFSS dataset for post-split data resampling With outlier without feature selection.

With outlier with feature selection

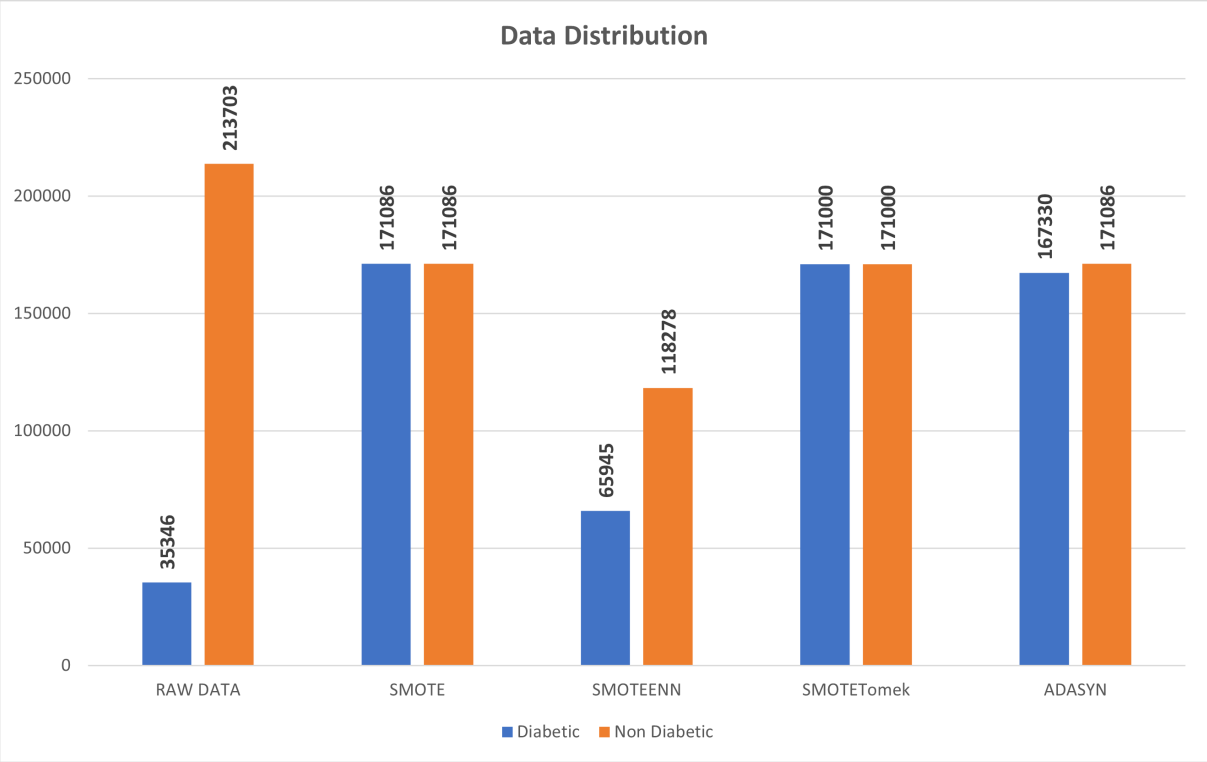


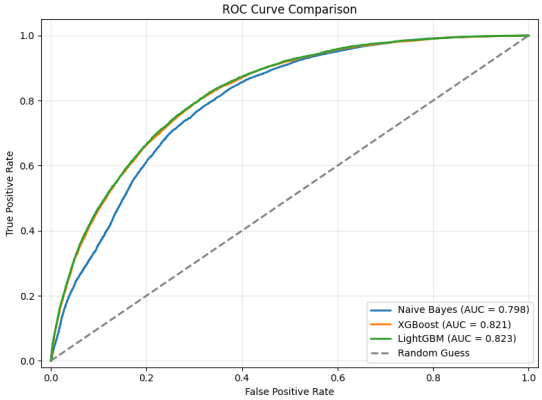
Figure 4.55: Data Distribution for With outlier with feature selection

Table 4.28: Performance comparison of models with outliers and with feature selection in the BRFSS dataset

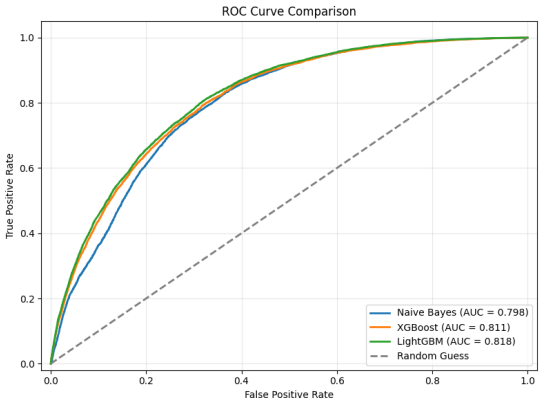
Resampling Technique	ML Model	Accuracy	Precision	Recall	F1-Score	AUC-ROC	MCC
RAW DATA	Naive Bayes	0.8008	0.3906	0.4365	0.4123	0.7975	0.3204
	XGBoost	0.8494	0.5933	0.1867	0.2841	0.8206	0.2722
	LightGBM	**0.8504**	**0.6135**	0.1761	0.2737	**0.8229**	0.2714
SMOTE	Naive Bayes	0.7440	0.3497	0.6975	0.4658	0.7984	0.3542
	XGBoost	0.7230	0.3362	0.7505	0.4644	0.8111	0.3582
	LightGBM	0.7152	0.3331	0.7779	**0.4664**	0.8178	**0.3646**
SMOTEENN	Naive Bayes	0.7630	0.3581	0.6070	0.4504	0.7911	0.3296
	XGBoost	0.7862	0.3872	0.5770	0.4634	0.7905	0.3467
	LightGBM	0.7831	0.3841	0.5892	0.4650	0.7955	0.3486
SMOTETomek	Naive Bayes	0.7439	0.3496	0.6979	0.4659	0.7984	0.3543
	XGBoost	0.7245	0.3377	0.7506	0.4658	0.8114	0.3600
	LightGBM	0.7164	0.3336	0.7740	0.4662	0.8180	0.3638
ADASYN	Naive Bayes	0.7298	0.3389	0.7241	0.4617	0.7977	0.3517
	XGBoost	0.7050	0.3235	0.7729	0.4561	0.8061	0.3508
	LightGBM	0.6941	0.3195	**0.8064**	0.4576	0.8155	0.3580

Table 4.28 indicates the highest performance in accuracy, precision and AUC-ROC attained without feature selection with the outliers is by the LightGBM models with raw data besides the maximum F1-score and MCC with SMOTE and maximum recall value with ADASYN is shown by the same model. In contrast, LightGBM also showed the least outcomes for recall, F1-score and MCC with the same raw data along with lowest

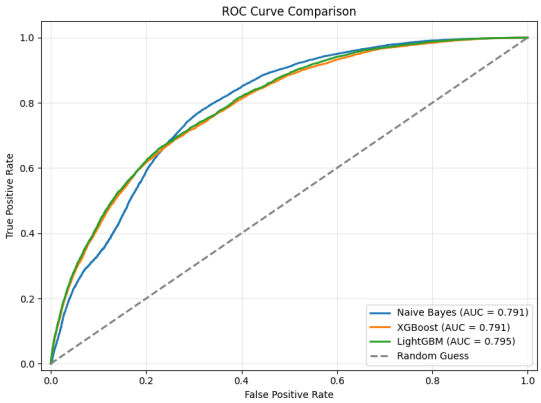
accuracy and precision by LightGBM with ADAYSN. Additionally, XGBoost illustrated the lowest AUC-ROC with SMOTEENN.



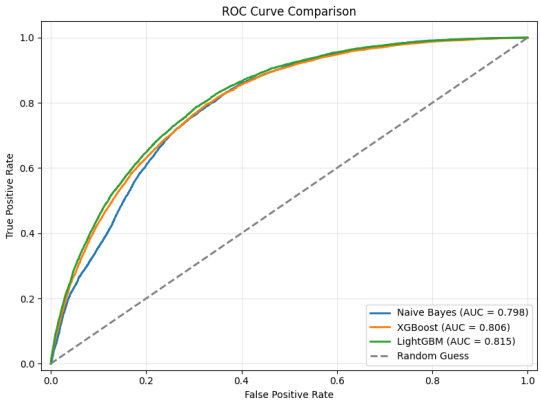
(a) RAW(No Resampling techniques)



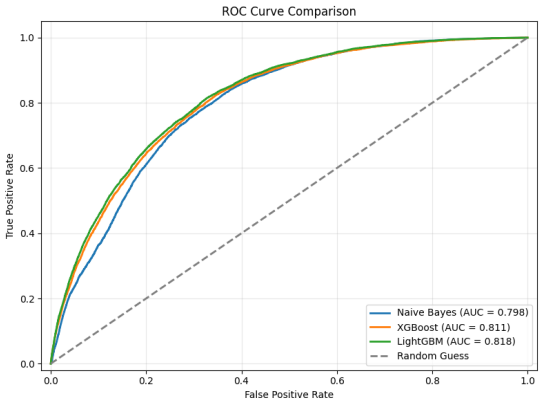
(b) SMOTE



(c) SMOTEENN



(d) ADASYN



(e) SMOTETomek

Figure 4.56: ROC curves demonstrating model performance on the BRFSS dataset for post-split data resampling With outlier with feature selection.

Looking at the data distribution for the post-split BRFSS-2015 dataset across all four circumstances whether outliers were removed or kept, and with or without feature selection the visuals consistently highlight the original class imbalance found in the raw data. More importantly, they effectively show how employing the four resampling methods (SMOTE, SMOTEENN, SMOTETomek, and ADASYN) successfully levels out this imbalance.

When evaluating the ROC curves across the different scenarios for this dataset, a consistent performance trend is clear for the Naive Bayes, XGBoost, and LightGBM models. Interestingly, in these post-split setups, the raw data actually shows higher AUC-ROC scores than the resampling methods, which refers to an improved True Positive Rate (Recall). Also, the models incorporating SMOTE, SMOTEENN, and SMOTETomek deliver a closely similar pattern to one another across the board. On the other hand, ADASYN generally falls a bit short, showing ROC curves that are still to some degree lower than the rest.

4.2.3 DiaHealth Prediction Result

Outlier removed without feature selection

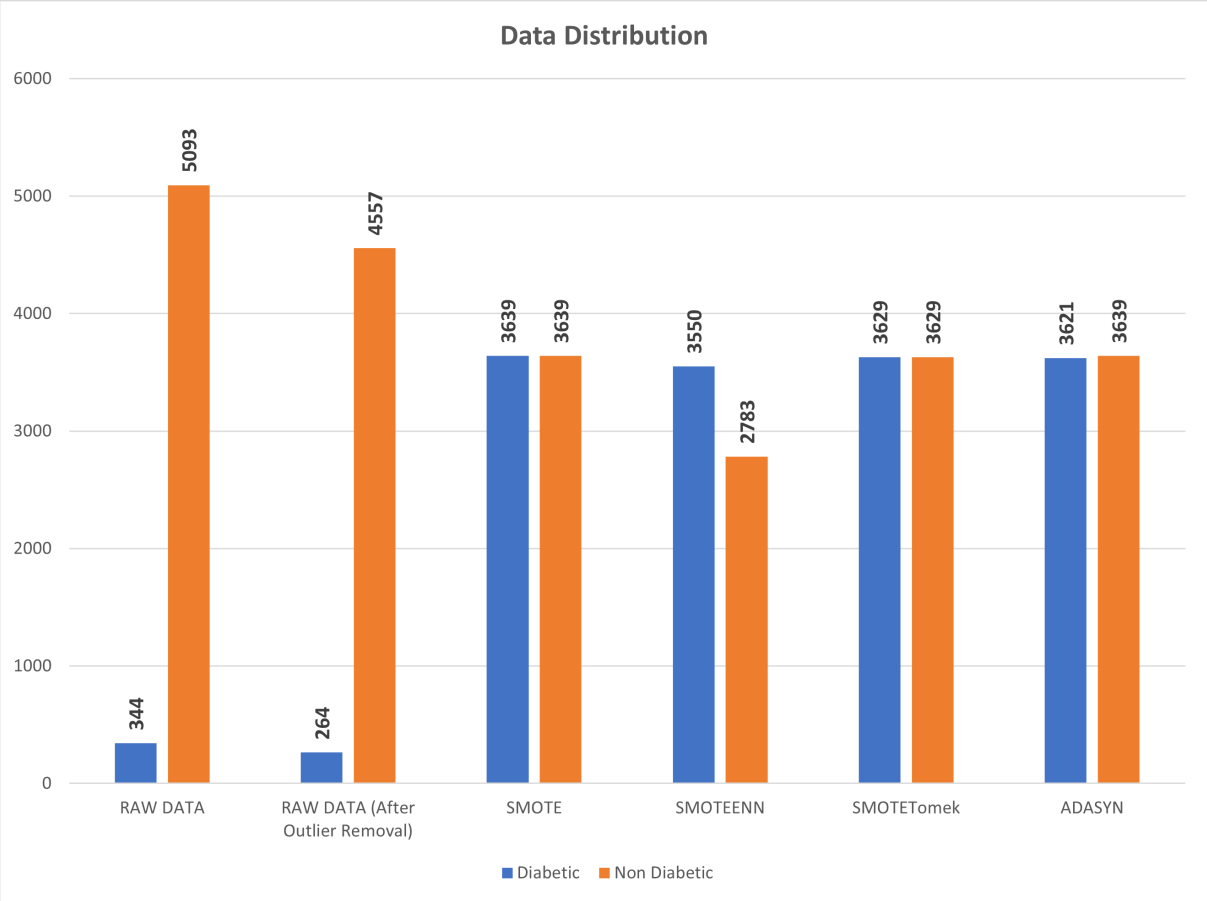
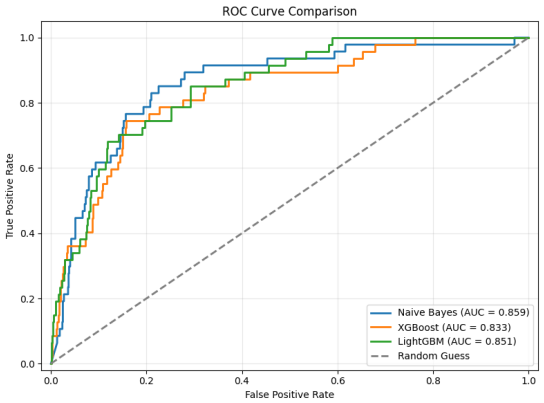


Figure 4.57: Data Distribution for Outlier removed without feature selection

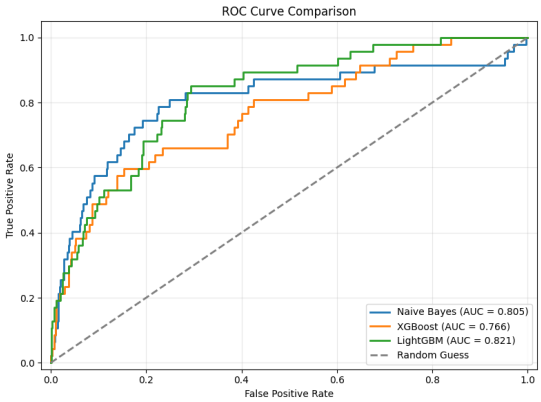
Table 4.29: Performance comparison of models with outliers removed and without feature selection in the Diahealth dataset

Resampling Technique	ML Model	Accuracy	Precision	Recall	F1-Score	AUC-ROC	MCC
RAW DATA	Naive Bayes	0.8964	0.2569	0.5957	**0.359**	**0.859**	**0.3451**
	XGBoost	0.942	0.3043	0.1489	0.2	0.8328	0.1856
	LightGBM	**0.9503**	**0.4737**	0.1915	0.2727	0.8514	0.2798
SMOTE	Naive Bayes	0.3637	0.0645	**0.8936**	0.1203	0.8052	0.1058
	XGBoost	0.9264	0.2778	0.3191	0.297	0.7664	0.2591
	LightGBM	0.9295	0.2857	0.2979	0.2917	0.8205	0.2547
SMOTEENN	Naive Bayes	0.3658	0.0647	**0.8936**	0.1207	0.8112	0.1066
	XGBoost	0.887	0.2075	0.4681	0.2876	0.8181	0.2592
	LightGBM	0.8933	0.2308	0.5106	0.3179	0.835	0.294
SMOTETomek	Naive Bayes	0.3637	0.0645	**0.8936**	0.1203	0.8049	0.1058
	XGBoost	0.9244	0.2593	0.2979	0.2772	0.8145	0.2382
	LightGBM	0.9285	0.2885	0.3191	0.303	0.8315	0.2658
ADASYN	Naive Bayes	0.3565	0.0638	**0.8936**	0.1191	0.8053	0.1029
	XGBoost	0.9326	0.3333	0.383	0.3564	0.8196	0.3219
	LightGBM	0.9275	0.2909	0.3404	0.3137	0.8336	0.2766

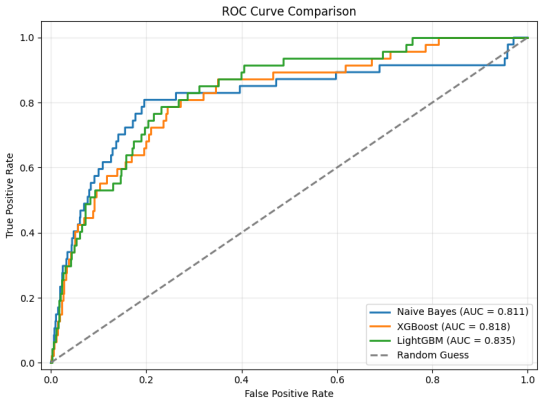
Table 4.29 illustrates that the highest performance in F1-score, AUC-ROC and MCC were attained without feature selection after removing the outliers is by Naive Bayes and highest accuracy and precision by LightGBM both with raw data. Naive Bayes under SMOTE, SMOTEENN, SMOTETomek and ADASYN also showed the highest and same recall throughout. In contrast, Naive Bayes showed the lowest results in accuracy, precision, F1-score and MCC under ADASYN resamplings along with lowest AUC-ROC with SMOTETomek. Whereas, XGBoost gave the lowest recall with raw data.



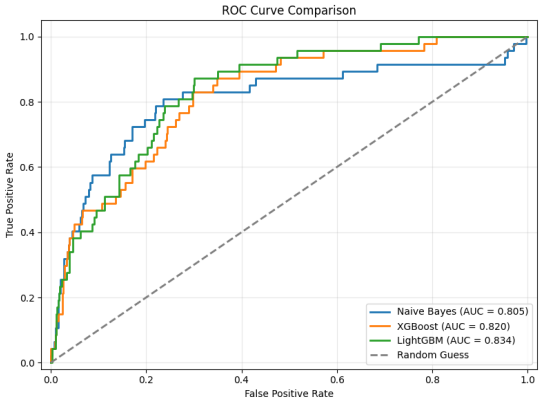
(a) RAW(No Resampling techniques)



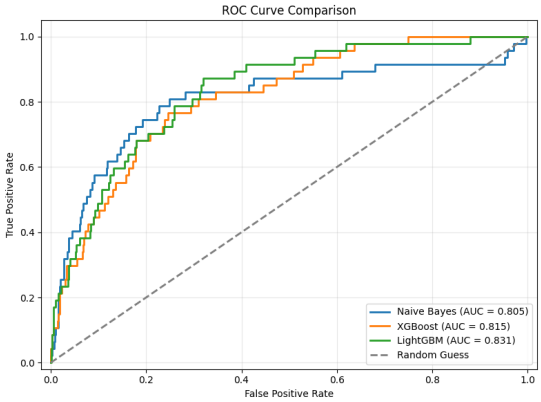
(b) SMOTE



(c) SMOTEENN



(d) ADASYN



(e) SMOTETomek

Figure 4.58: ROC curves demonstrating model performance on the DiaHealth dataset for post-split data resampling Outlier removed without feature selection.

Outlier removed with feature selection

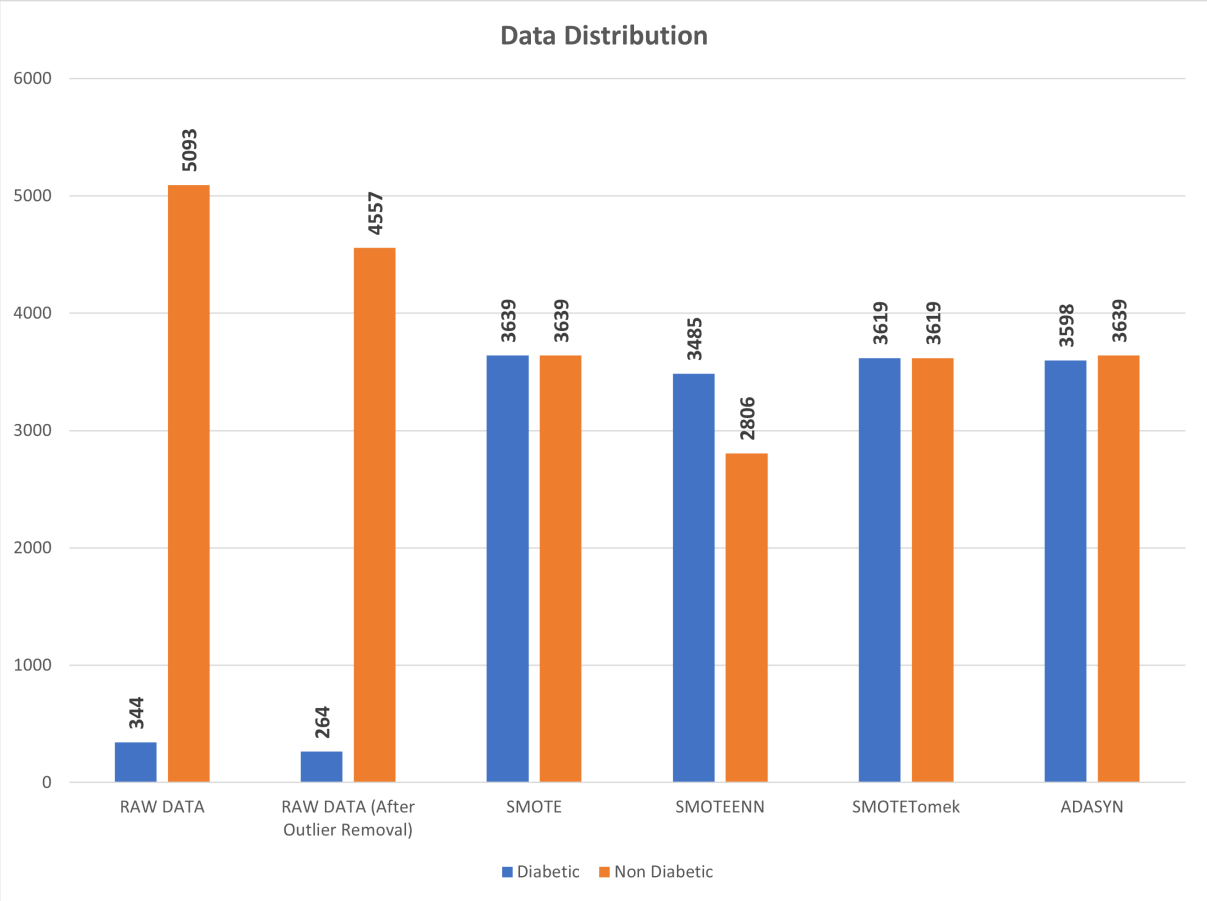


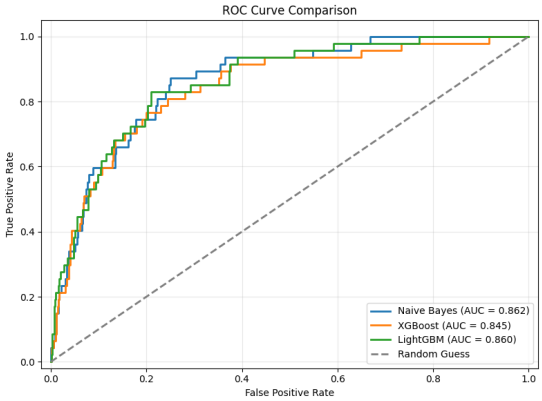
Figure 4.59: Data Distribution for Outlier removed with feature selection

Table 4.30: Performance comparison of models with outliers removed and with feature selection in the Diahealth dataset

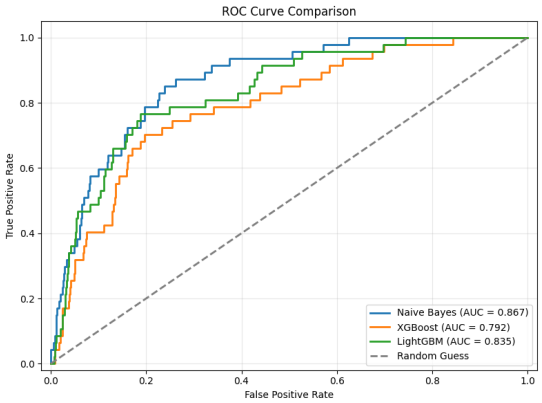
Resampling Technique	ML Model	Accuracy	Precision	Recall	F1-Score	AUC-ROC	MCC
RAW DATA	Naive Bayes	0.8943	0.2523	0.5957	**0.3544**	0.8625	**0.3409**
	XGBoost	0.9451	0.375	0.1915	0.2535	0.8449	0.2421
	LightGBM	**0.9492**	**0.4545**	0.2128	0.2899	0.8602	0.288
SMOTE	Naive Bayes	0.829	0.1828	0.7234	0.2918	0.8675	0.3044
	XGBoost	0.8808	0.1792	0.4043	0.2484	0.7915	0.213
	LightGBM	0.8725	0.1984	0.5319	0.289	0.8353	0.2695
SMOTEENN	Naive Bayes	0.7896	0.1638	**0.8085**	0.2724	**0.8731**	0.3008
	XGBoost	0.8456	0.189	0.6596	0.2938	0.826	0.295
	LightGBM	0.8301	0.1803	0.7021	0.287	0.8373	0.2958
SMOTETomek	Naive Bayes	0.828	0.1818	0.7234	0.2906	0.8675	0.3032
	XGBoost	0.8902	0.2294	0.5319	0.3205	0.801	0.2995
	LightGBM	0.8663	0.1894	0.5319	0.2793	0.8269	0.2602
ADASYN	Naive Bayes	0.828	0.1818	0.7234	0.2906	0.867	0.3032
	XGBoost	0.8767	0.1604	0.3617	0.2222	0.7789	0.1822
	LightGBM	0.8642	0.1866	0.5319	0.2762	0.8122	0.2572

Table 4.30 illustrates that the highest performance in accuracy and precision were achieved by LightGBM and in F1-score and MCC were attained by Naive Bayes both with raw

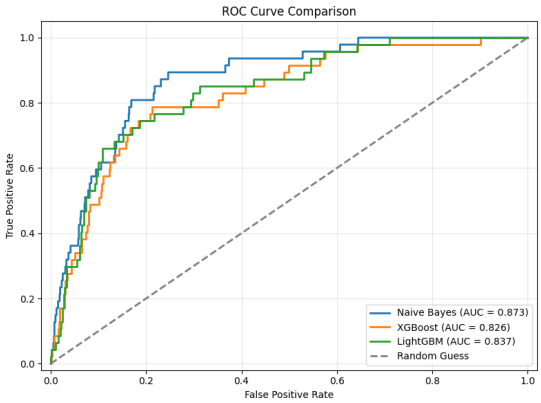
data without feature selection after removing the outliers. Naive Bayes also obtained the highest recall with SMOTEENN. In contrast, XGBoost showed the lowest results in precision, F1-score, AUC-ROC and MCC under ADASYN resamplings along with lowest recall with raw data. Whereas, Naive Bayes gave the least accuracy with SMOTEENN.



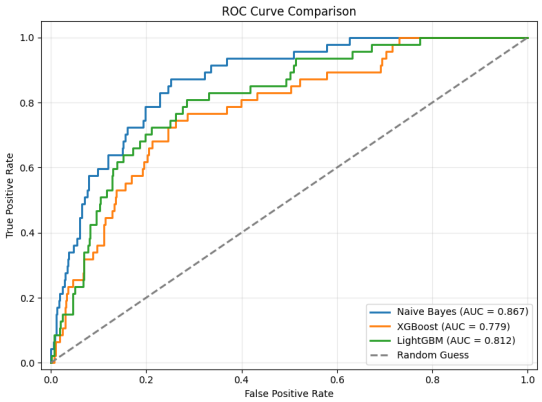
(a) RAW(No Resampling techniques)



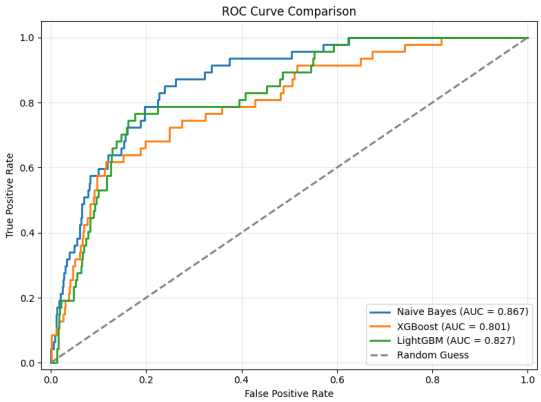
(b) SMOTE



(c) SMOTEENN



(d) ADASYN



(e) SMOTETomek

Figure 4.60: ROC curves demonstrating model performance on the DiaHealth dataset for post-split data resampling Outlier removed with feature selection.

With outlier without feature selection

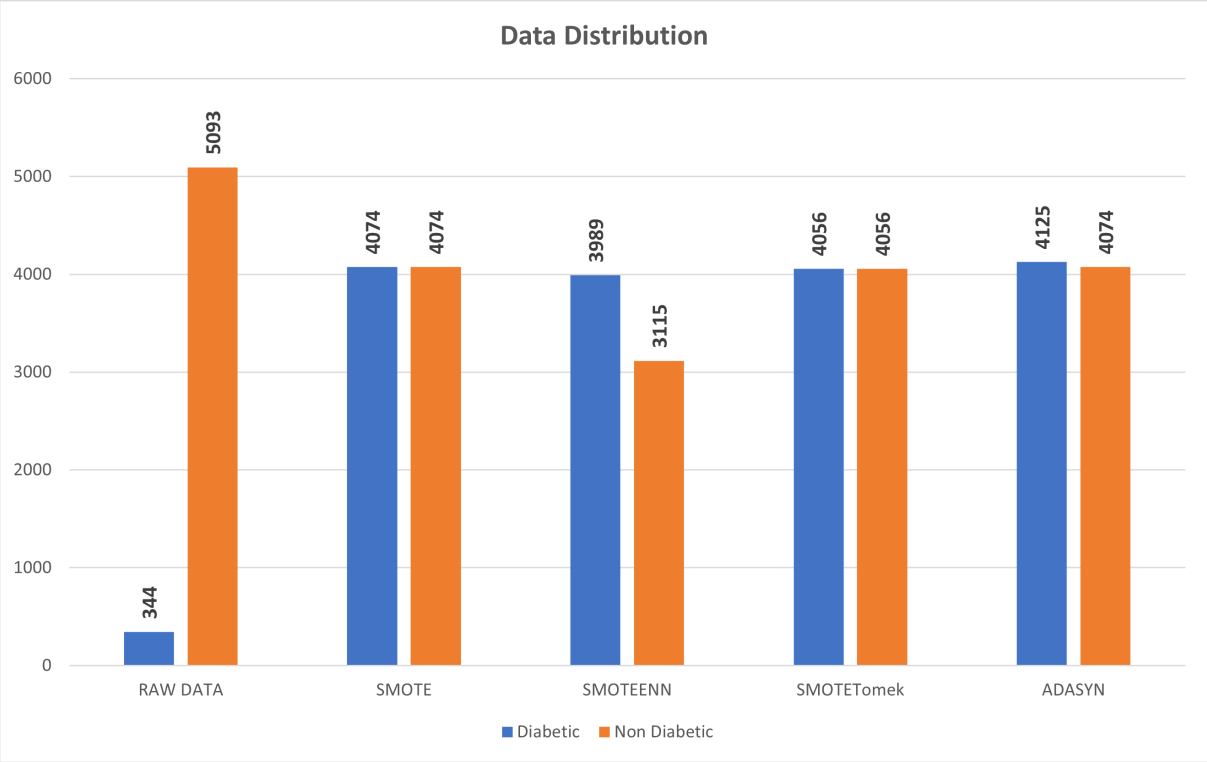


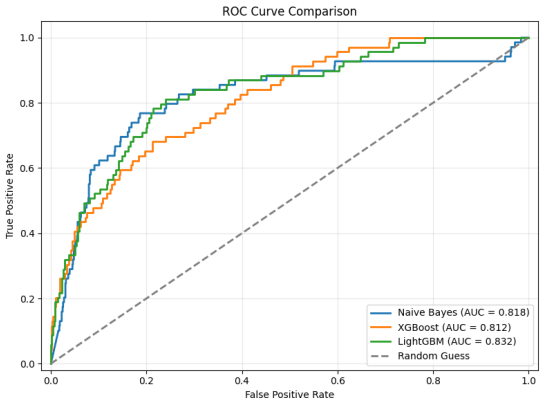
Figure 4.61: Data Distribution for With outlier without feature selection

Table 4.31: Performance comparison of models with outliers and without feature selection in the Diahealth dataset

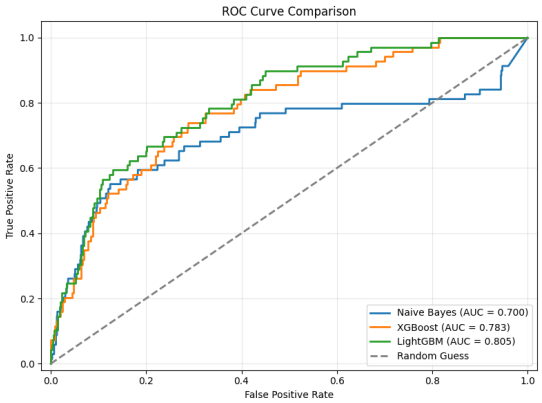
Resampling Technique	ML Model	Accuracy	Precision	Recall	F1-Score	AUC-ROC	MCC
RAW DATA	Naive Bayes	0.8713	0.2739	0.6232	**0.3805**	0.8184	**0.3546**
	XGBoost	0.9347	0.4667	0.2029	0.2828	0.8122	0.2786
	LightGBM	**0.9357**	**0.4815**	0.1884	0.2708	**0.8321**	0.2736
SMOTE	Naive Bayes	0.1792	0.0598	0.8116	0.1114	0.6999	-0.0366
	XGBoost	0.8961	0.225	0.2609	0.2416	0.7833	0.1868
	LightGBM	0.9017	0.2564	0.2899	0.2721	0.8054	0.2201
SMOTEENN	Naive Bayes	0.2472	0.066	**0.8261**	0.1222	0.7278	0.0206
	XGBoost	0.8658	0.2416	0.5217	0.3303	0.8123	0.2912
	LightGBM	0.8548	0.2236	0.5217	0.313	0.8076	0.2739
SMOTETomek	Naive Bayes	0.1792	0.0598	0.8116	0.1114	0.6999	-0.0366
	XGBoost	0.9062	0.2676	0.2754	0.2714	0.7807	0.2214
	LightGBM	0.9072	0.2714	0.2754	0.2734	0.8074	0.2238
ADASYN	Naive Bayes	0.1673	0.0599	0.8261	0.1118	0.697	-0.0376
	XGBoost	0.898	0.2558	0.3188	0.2839	0.7641	0.2313
	LightGBM	0.9026	0.2466	0.2609	0.2535	0.7948	0.2015

Table 4.31 illustrates that the highest performance in accuracy, precision and AUC-ROC were achieved by LightGBM and in F1-score and MCC were attained by Naive Bayes both with raw data without feature selection after removing the outliers. Naive Bayes also obtained the highest recall with SMOTEENN. In contrast, Naive Bayes showed the lowest results in precision, F1-score and MCC under SMOTE resamplings along with

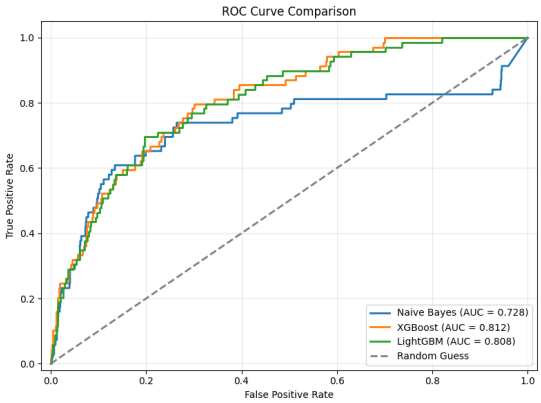
lowest precision and F1-score with SMOTETomek and Lowest accuracy and AUC-ROC with ADASYN.



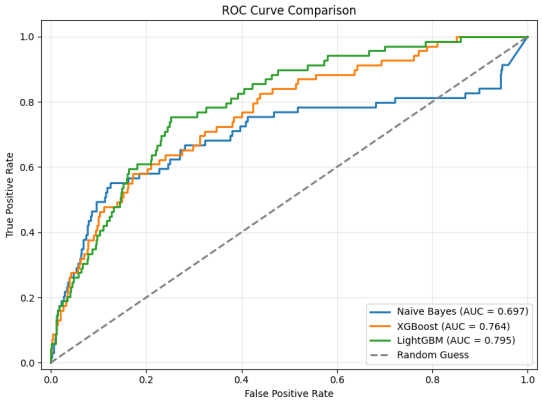
(a) RAW(No Resampling techniques)



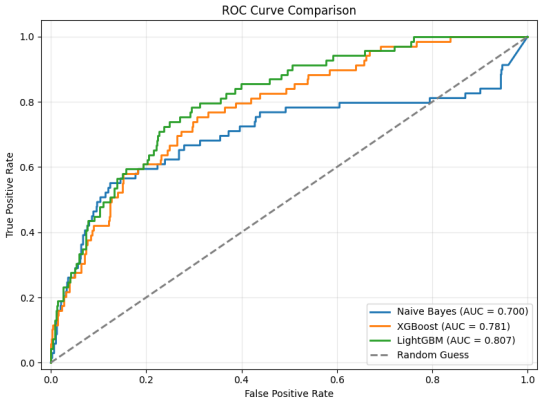
(b) SMOTE



(c) SMOTEENN



(d) ADASYN



(e) SMOTETomek

Figure 4.62: ROC curves demonstrating model performance on the DiaHealth dataset for post-split data resampling With outlier without feature selection.

With outlier with feature selection

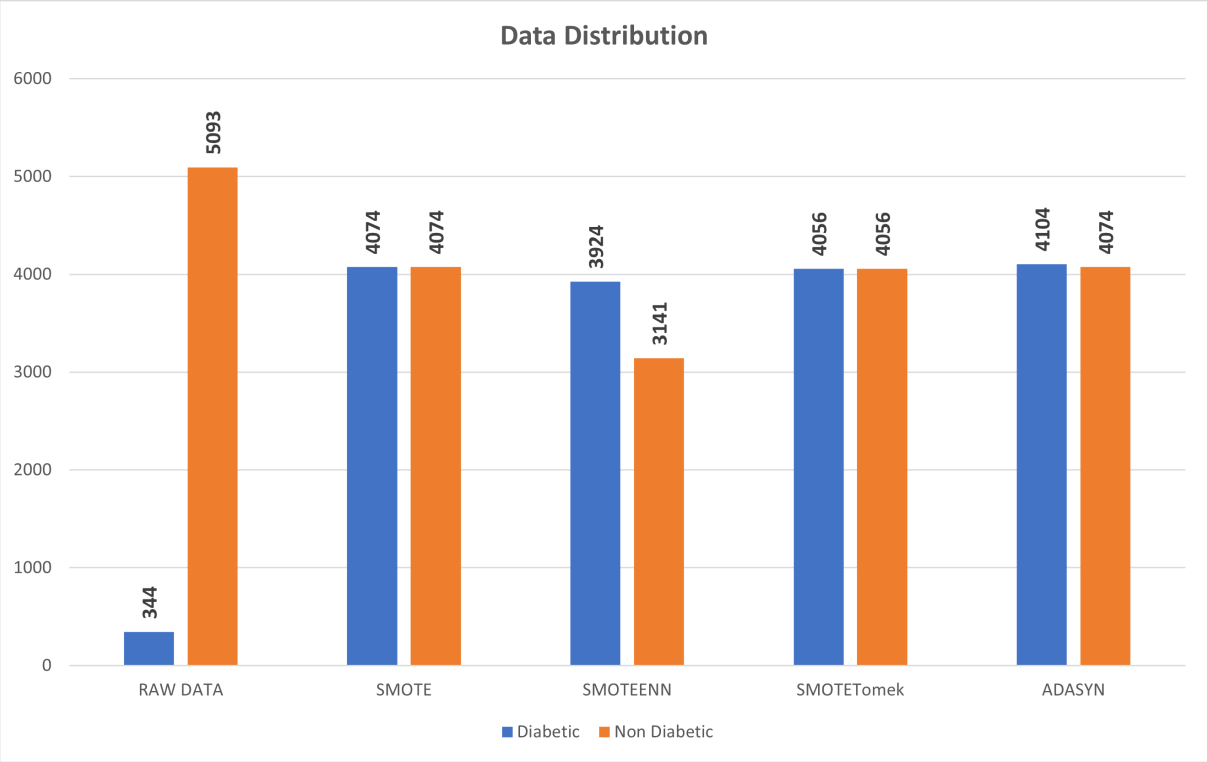


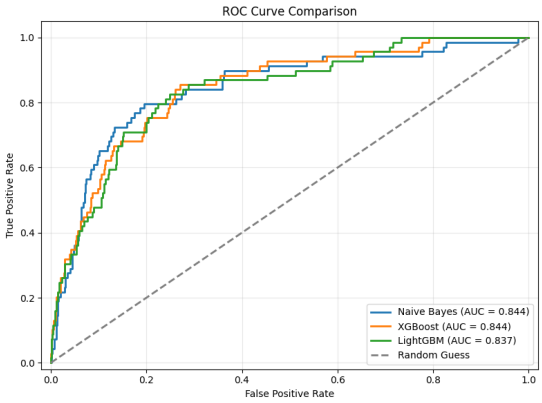
Figure 4.63: Data Distribution for With outlier with feature selection

Table 4.32: Performance comparison of models with outliers and with feature selection in the Diahealth dataset

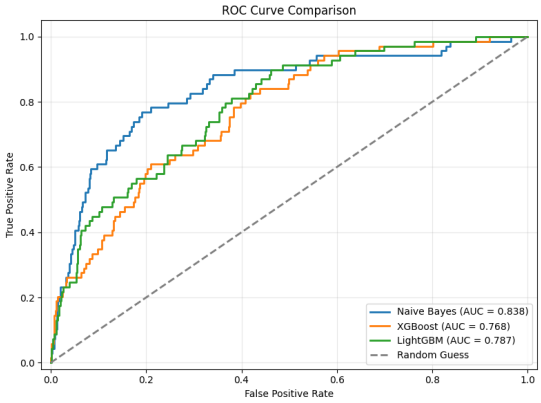
Resampling Technique	ML Model	Accuracy	Precision	Recall	F1-Score	AUC-ROC	MCC
RAW DATA	Naive Bayes	0.8814	0.3	0.6522	**0.411**	0.8441	**0.3882**
	XGBoost	0.9329	0.4444	0.2319	0.3048	**0.8442**	0.2892
	LightGBM	**0.9375**	**0.5172**	0.2174	0.3061	0.8366	0.3081
SMOTE	Naive Bayes	0.7693	0.1862	0.7826	0.3008	0.8382	0.3037
	XGBoost	0.8594	0.1818	0.3478	0.2388	0.7684	0.1805
	LightGBM	0.8511	0.2075	0.4783	0.2895	0.7871	0.2447
SMOTEENN	Naive Bayes	0.7702	0.1953	**0.8406**	0.3169	0.8519	0.3315
	XGBoost	0.8079	0.1602	0.4783	0.24	0.7687	0.1919
	LightGBM	0.8107	0.1754	0.5362	0.2643	0.7813	0.2253
SMOTETomek	Naive Bayes	0.7702	0.1869	0.7826	0.3017	0.8385	0.3046
	XGBoost	0.8621	0.1707	0.3043	0.2188	0.7511	0.1572
	LightGBM	0.8557	0.2067	0.4493	0.2831	0.7835	0.235
ADASYN	Naive Bayes	0.7583	0.1788	0.7826	0.2911	0.8388	0.2935
	XGBoost	0.8548	0.1704	0.3333	0.2255	0.7594	0.1652
	LightGBM	0.8557	0.2143	0.4783	0.296	0.784	0.2514

Table 4.32 illustrates that the highest performance in accuracy and precision were achieved by LightGBM, in F1-score and MCC were attained by Naive Bayes, and in AUC-ROC by XGBoost all with raw data without feature selection after removing the outliers. In contrast, XGBoost showed the lowest results in F1-score, AUC-ROC and MCC under SMOTETomek resamplings along with lowest precision with SMOTEENN. Whereas,

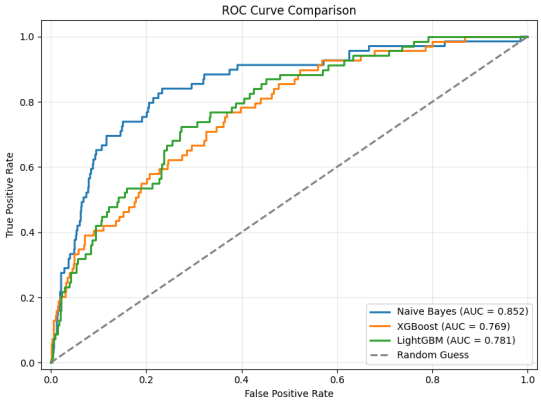
Naive Bayes gave the least accuracy with ADASYN and LightGBM displayed least recall with raw data.



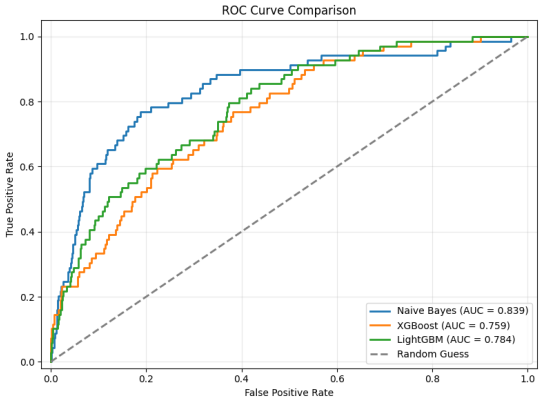
(a) RAW(No Resampling techniques)



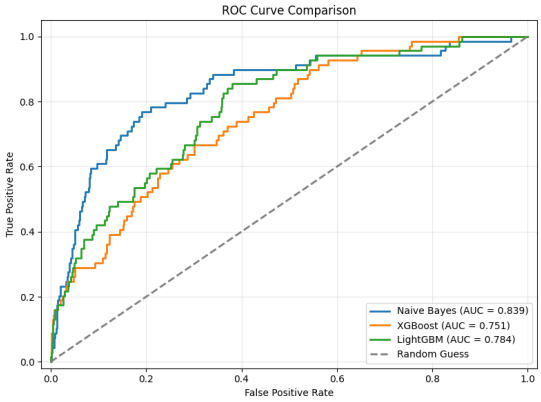
(b) SMOTE



(c) SMOTEENN



(d) ADASYN



(e) SMOTETomek

Figure 4.64: ROC curves demonstrating model performance on the DiaHealth dataset for post-split data resampling With outlier with feature selection.

Looking at the data distribution for the post-split DiaHealth dataset across all four circumstances whether outliers were removed or kept, and with or without feature selection the visuals consistently highlight the original class imbalance found in the raw data. More importantly, they effectively show how employing the four resampling methods (SMOTE, SMOTEENN, SMOTETomek, and ADASYN) successfully levels out this imbalance.

When evaluating the ROC curves across the different scenarios for this dataset, a consistent performance trend is clear for the Naive Bayes, XGBoost, and LightGBM

models. Interestingly, in these post-split setups, the raw data actually shows higher AUC-ROC scores than the resampling methods, which refers to an improved True Positive Rate (Recall). Also, the models incorporating SMOTE, SMOTEENN, and SMOTETomek deliver a closely similar pattern to one another across the board. On the other hand, ADASYN generally falls a bit short, showing ROC curves that are still to some degree lower than the rest.

4.3 Comparative analysis: Post-split VS Pre-split

To identify which configuration gives the best result, we have conducted several experimental setups for three datasets (Pima Indian, BRFSS-15, Diahealth) based on varying components (Imputation/Elimination, with/without outliers, with/without feature selection). Additionally, we applied four robust resampling techniques and used three machine learning models to predict the output.

In order to answer our research question, we executed these experiments across two separate pipelines. One is the pre-split resampling pipeline and the other is the post-split resampling pipeline. From these separate pipelines, we have extracted the best results of the top-performing models alongside their optimal resampling techniques. The inflation comparison of both pipelines is shown in Table 4.33, Table 4.34, and Table 4.35. Below, these three comparison tables are provided to clearly illustrate the performance differences and the exact amount of metric inflation between the pipelines.

After comparing all three dataset results, it is clear that the pre-split resampling pipeline produces accuracy scores approximately (8.76%–17.52%) higher than those of the post-split pipeline across all datasets and configurations (PIDD: ≈17.52%, BRFSS-2015: ≈11.23%, Diahealth: ≈8.76%). For other metrics, the inflation is even more pronounced: precision inflation reaches up to 72.92% in the Diahealth dataset, and MCC inflation reaches up to 64.29%. However, in medical machine learning, achieving the highest possible numerical result is not the only priority. We should raise the question: does a higher score actually translate to a safer and more reliable clinical model, or does it merely indicate algorithmic memorization? Because real-world diagnosis requires absolute algorithmic integrity. Clinically, these inflated metric values might not be able to diagnose actual real-life patients. The result's viability in actual healthcare scenarios, and the criteria for selecting a genuinely robust pipeline will be discussed in detail in Section-5.

Table 4.33: Pima Indian Diabetes Dataset comparison: Pre-split vs. Post-split

	Pre-Split							Post-Split							Inflation (Pre–Post)				
Configuration	**Resampling**	**Model**	**Acc**	**Prec**	**Rec**	**F1**	**MCC**	**Resampling**	**Model**	**Acc**	**Prec**	**Rec**	**F1**	**MCC**	**Acc**	**Prec**	**Rec**	**F1**	**MCC**
Imputation, Outlier removed, no feature selection	SMOTEENN	LightGBM	0.9906	0.9839	1.0000	0.9919	0.9808	SMOTEENN	Naive Bayes	0.7862	0.6522	0.8654	0.7438	0.5832	0.2044	0.3317	0.1346	0.2481	0.3976
Imputation, Outlier removed, with feature selection	SMOTEENN	XGBoost	0.9633	0.9825	0.9492	0.9655	0.9270	SMOTEENN	Naive Bayes	0.7793	0.6515	0.8269	0.7288	0.5582	0.1840	0.3310	0.1223	0.2367	0.3688
Imputation, with outliers, no feature selection	SMOTEENN	LightGBM	0.9391	0.9130	0.9844	0.9474	0.8789	SMOTEENN	XGBoost	0.7857	0.6129	0.8085	0.6972	0.5486	0.1534	0.3001	0.1759	0.2502	0.3303
Imputation, with outliers, with feature selection	SMOTEENN	XGBoost	0.9732	1.0000	0.9516	0.9752	0.9475	SMOTEENN	XGBoost	0.8052	0.6393	0.8298	0.7222	0.5877	0.1680	0.3607	0.1218	0.2530	0.3598
Elimination, Outlier removed, no feature selection	SMOTEENN	XGBoost	0.9516	0.9211	1.0000	0.9589	0.9048	SMOTETomek	XGBoost	0.7746	0.7000	0.7500	0.7241	0.5350	0.1770	0.2211	0.2500	0.2348	0.3698
Elimination, Outlier removed, with feature selection	SMOTEENN	XGBoost	0.9677	0.9487	1.0000	0.9737	0.9342	ADASYN	LightGBM	0.7606	0.7037	0.6786	0.6909	0.4958	0.2071	0.2450	0.3214	0.2828	0.4384
Elimination, with outliers, no feature selection	SMOTEENN	LightGBM	0.9841	0.9730	1.0000	0.9863	0.9680	ADASYN	LightGBM	0.8354	0.7308	0.7600	0.7451	0.6239	0.1487	0.2422	0.2400	0.2412	0.3441
Elimination, with outliers, with feature selection	SMOTEENN	LightGBM	0.9688	0.9487	1.0000	0.9737	0.9373	SMOTEENN	XGBoost	0.8101	0.6923	0.7200	0.7059	0.5660	0.1587	0.2564	0.2800	0.2678	0.3713
Avg. Inflation															**0.1752**	**0.2985**	**0.2058**	**0.2518**	**0.3725**

Table 4.34: BRFSS-2015 comparison: Pre-split vs. Post-split

	Pre-Split								Post-Split								Inflation (Pre−Post)					
Configuration	**Resampling**	**Model**	**Acc**	**Prec**	**Rec**	**F1**	**AUC**	**MCC**	**Resampling**	**Model**	**Acc**	**Prec**	**Rec**	**F1**	**AUC**	**MCC**	**Acc**	**Prec**	**Rec**	**F1**	**AUC**	**MCC**
Outlier removed, no feature selection	SMOTEENN	LightGBM	0.8633	0.8804	0.8999	0.8901	0.9385	0.7098	SMOTETomek	LightGBM	0.7572	0.3361	0.6424	0.4413	0.7969	0.3312	0.1061	0.5443	0.2575	0.4488	0.1416	0.3786
Outlier removed, with feature selection	SMOTEENN	XGBoost	0.8947	0.8605	0.8096	0.8343	0.9449	0.7579	SMOTEENN	LightGBM	0.7853	0.3618	0.5735	0.4437	0.7947	0.3320	0.1094	0.4987	0.2361	0.3906	0.1502	0.4259
With outliers, no feature selection	SMOTEENN	LightGBM	0.8673	0.8832	0.9043	0.8937	0.9407	0.7177	SMOTETomek	LightGBM	0.7602	0.3567	0.6206	0.4530	0.7937	0.3332	0.1071	0.5265	0.2837	0.4407	0.1470	0.3845
With outliers, with feature selection	SMOTEENN	XGBoost	0.9095	0.8840	0.8476	0.8654	0.9622	0.7977	SMOTEENN	LightGBM	0.7831	0.3841	0.5892	0.4650	0.7955	0.3486	0.1264	0.4999	0.2584	0.4004	0.1667	0.4491
Avg. Inflation																	**0.1123**	**0.5174**	**0.2589**	**0.4201**	**0.1514**	**0.4095**

Table 4.35: Diahealth comparison: Pre-split vs. Post-split

	Pre-Split								Post-Split								Inflation (Pre−Post)					
Configuration	**Resampling**	**Model**	**Acc**	**Prec**	**Rec**	**F1**	**AUC**	**MCC**	**Resampling**	**Model**	**Acc**	**Prec**	**Rec**	**F1**	**AUC**	**MCC**	**Acc**	**Prec**	**Rec**	**F1**	**AUC**	**MCC**
Outlier removed, no feature selection	SMOTEENN	LightGBM	0.9694	0.9697	0.9751	0.9724	0.9948	0.9381	SMOTEENN	LightGBM	0.8933	0.2308	0.5106	0.3179	0.8350	0.2940	0.0761	0.7389	0.4645	0.6545	0.1598	0.6441
Outlier removed, with feature selection	SMOTEENN	XGBoost	0.9636	0.9502	0.9865	0.9680	0.9863	0.9268	SMOTETomek	XGBoost	0.8902	0.2294	0.5319	0.3205	0.8010	0.2995	0.0734	0.7208	0.4546	0.6475	0.1853	0.6273
With outliers, no feature selection	SMOTEENN	XGBoost	0.9679	0.9676	0.9762	0.9719	0.9948	0.9344	SMOTEENN	XGBoost	0.8658	0.2416	0.5217	0.3303	0.8123	0.2912	0.1021	0.7260	0.4545	0.6416	0.1825	0.6432
With outliers, with feature selection	SMOTEENN	XGBoost	0.9546	0.9452	0.9743	0.9596	0.9879	0.9085	ADASYN	LightGBM	0.8557	0.2143	0.4783	0.2960	0.7840	0.2514	0.0989	0.7309	0.4960	0.6636	0.2039	0.6571
Avg. Inflation																	**0.0876**	**0.7292**	**0.4674**	**0.6518**	**0.1829**	**0.6429**

4.4 Metric Comparison and Clinical Utility

Metric comparison is the most important thing to understand if the model is working or not. A model will only be useful when the evaluation of the model will be done correctly. Our previous sections already emphasized on the critical concerns of data leakage, this section focuses on how choosing the wrong evaluation metric can hide a model's actual clinical flaws.

4.4.1 Accuracy vs. Clinical Metrics

In standard machine learning, accuracy is a common choice as an evaluation metric. However, in healthcare datasets like BRFSS-2015 or Diahealth, which are highly imbalanced with a significantly lower number of diabetes positive patients compared to healthy persons, relying on accuracy can create a dangerous illusion. Choosing accuracy as a primary metric for these types of skewed datasets can lead to serious issues.

By looking at our post-split results, we can clearly see how high accuracy masks poor clinical performance. For example, in the BRFSS-2015 post-split pipeline, the model achieved an accuracy of 0.73. To an untrained eye, a 0.73 success rate might look acceptable. But when we look at the F1-Score for the exact same model, it drops down to just 0.35, and the MCC falls to a very low score of 0.27. A score below 0.5 demonstrates weak predictive power and highlights the model's inability to reliably distinguish between the majority and minority classes [80].

We see the exact same pattern in the Diahealth dataset. The post-split Accuracy shows a decent value of 0.74, but the F1-Score is only 0.39 and the MCC is just 0.25. Even in the Pima Indian Diabetes Dataset (PIDD), an 0.80 Accuracy hides a much lower F1-Score of 0.69. What does this mean in real life? It means the model is getting a high Accuracy score simply by guessing that most people are healthy (the majority class). The model fails heavily when trying to correctly identify the actual diabetic patients (the minority class), which is reflected in the terrible F1 and MCC scores. If a hospital were to deploy a model based purely on its 0.74 Accuracy, it would drastically fail to diagnose real diabetic patients, proving that Accuracy is a highly misleading metric for this domain.

4.4.2 Recommended Metric Protocol

Based on the large performance differences found in our experiments, this study suggests that accuracy should not be used as the main metric in diabetes prediction research. The evaluation process needs to use better assessment standards because they will evaluate model performance through methods that provide reliable and clinically significant results.

Primary Metrics (MCC and F1-Score): The Matthews Correlation Coefficient (MCC) should be the main metric for model selection in this study area. The Matthews Correlation Coefficient (MCC) provides more accurate results because it considers all four components of the confusion matrix which include true positives and true negatives and false positives and false negatives. This makes it much harder for a model to get a high MCC score if it is doing poorly on the minority diabetic class. Along with MCC, the F1-score should also be given high importance, because it shows the balance between precision and recall.

Secondary Clinical Metric (Recall/Sensitivity): In medical prediction, false negatives are a serious concern. Predicting that a diabetic patient is healthy can be much more

harmful than predicting that a healthy person may need further testing. For this reason, recall should always be checked carefully. A useful diabetes prediction model should be able to identify as many true positive diabetic cases as possible.

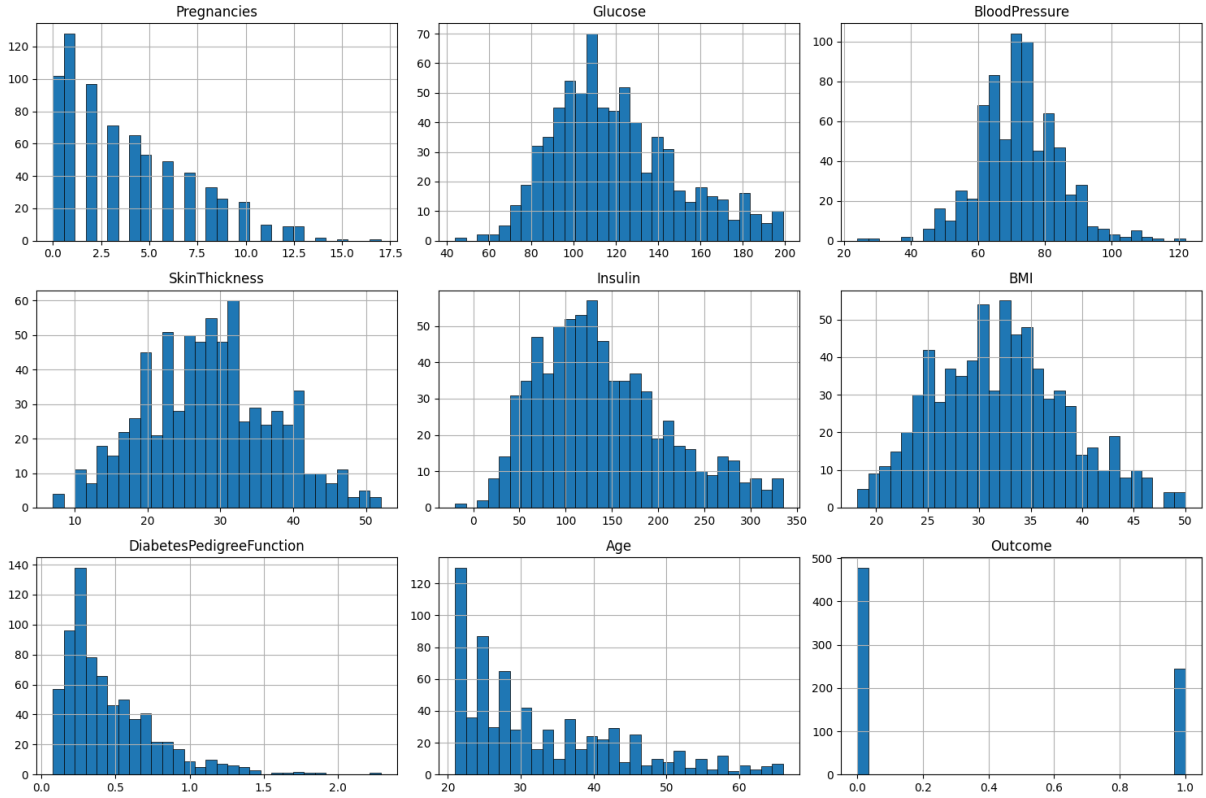
Secondary Global Metric (AUC-ROC): The AUC-ROC would be used as a secondary metric to measure the model's overall ability for class separation across different decision thresholds. It gives us a broader picture of classification performance, but it should not be treated as the only decision-making metric.

Accuracy for Reference Only: Accuracy may still be reported, but only for comparison with earlier published studies. It should not be used as the main basis for model selection, threshold setting, or clinical judgment. The findings of this study show that accuracy is not only one of the most inflated metrics under pre-split conditions, but also one of the least helpful for understanding how a model will behave on real imbalanced medical data. Relying too much on accuracy may lead researchers to choose a model that looks strong in numbers but is actually weaker in detecting diabetic patients.

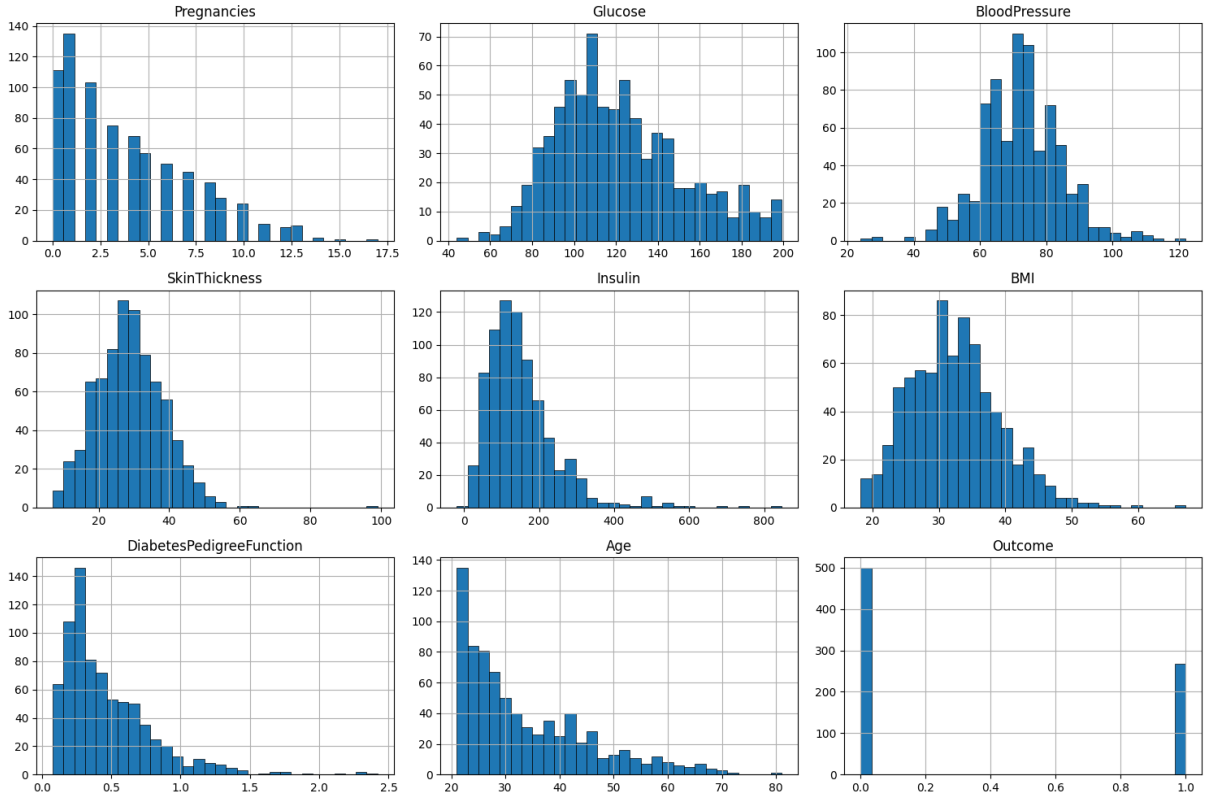
The study recommends that upcoming diabetes prediction studies should first focus on measuring MCC and F1-score because these two metrics hold primary importance for their research. The statistical methods enable researchers to select models which demonstrate strong statistical performance and prove better for actual medical usage.

4.5 Impact of Resampling on Data Distribution

To better understand the effect of resampling placement, the class distributions of the raw dataset, the resampled training data, and the test data were compared for both the pre-split and post-split pipelines. This comparison is important because, in an imbalanced medical dataset, the reliability of model evaluation depends not only on model performance but also on whether the test set still reflects the original data distribution. The two pipelines in this study differ mainly because the post-split method keeps the original test set distribution intact, while the pre-split method alters it before testing.

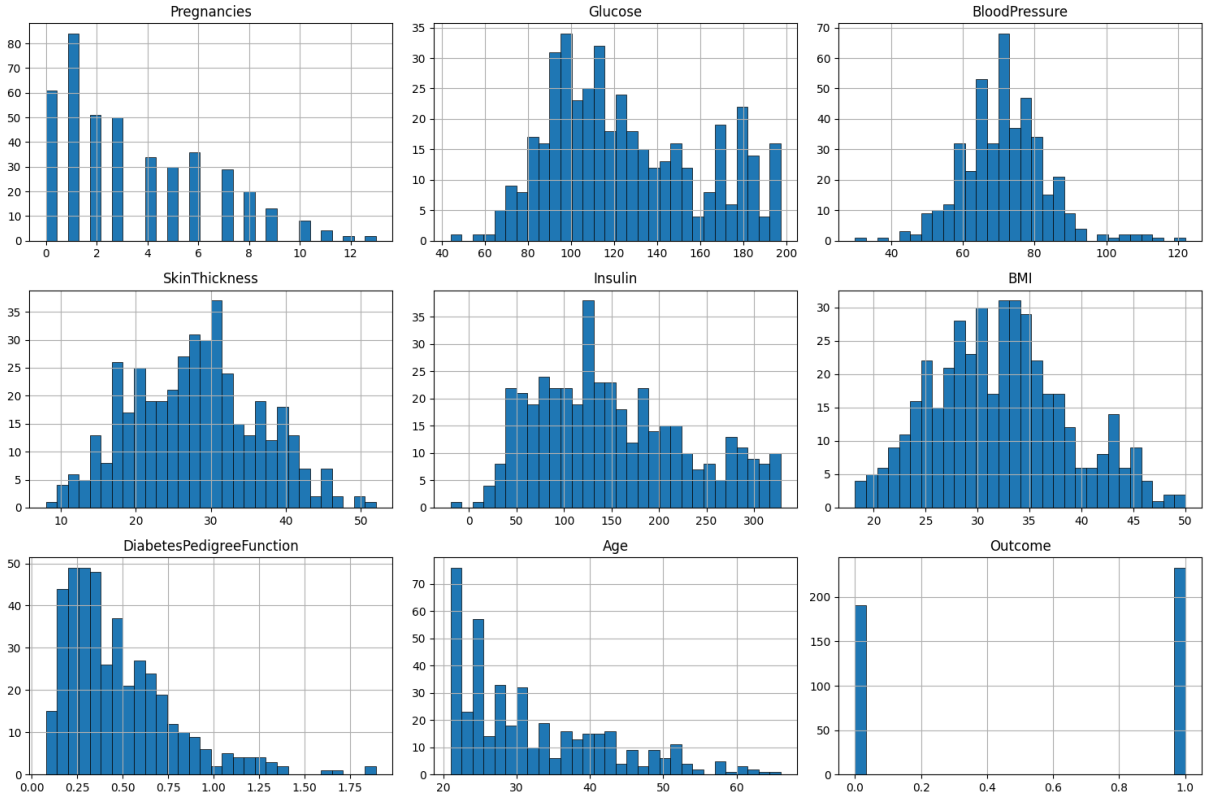


(a) Raw Class Distribution

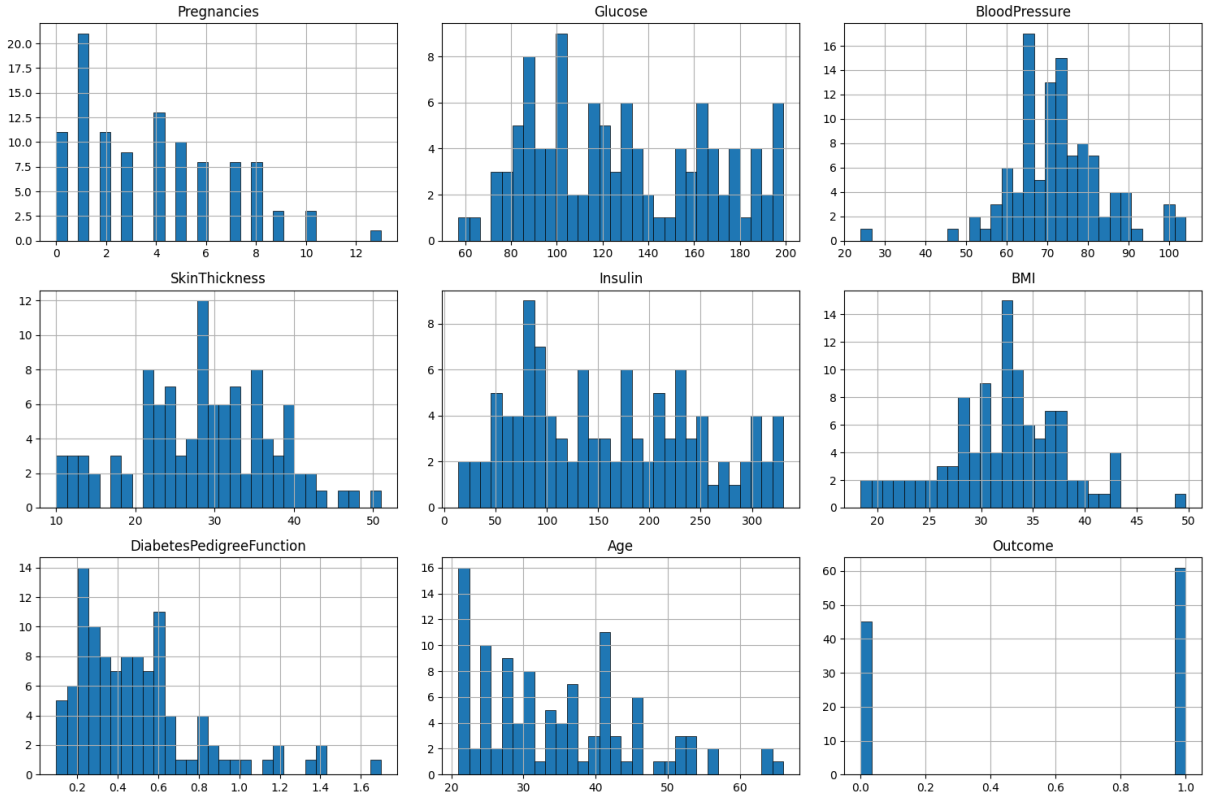


(b) After Imputation Outlier Removal

Figure 4.65: Histograms demonstrating the feature distributions of the PIMA indian dataset (Part 1): Raw distribution and after outlier removal.

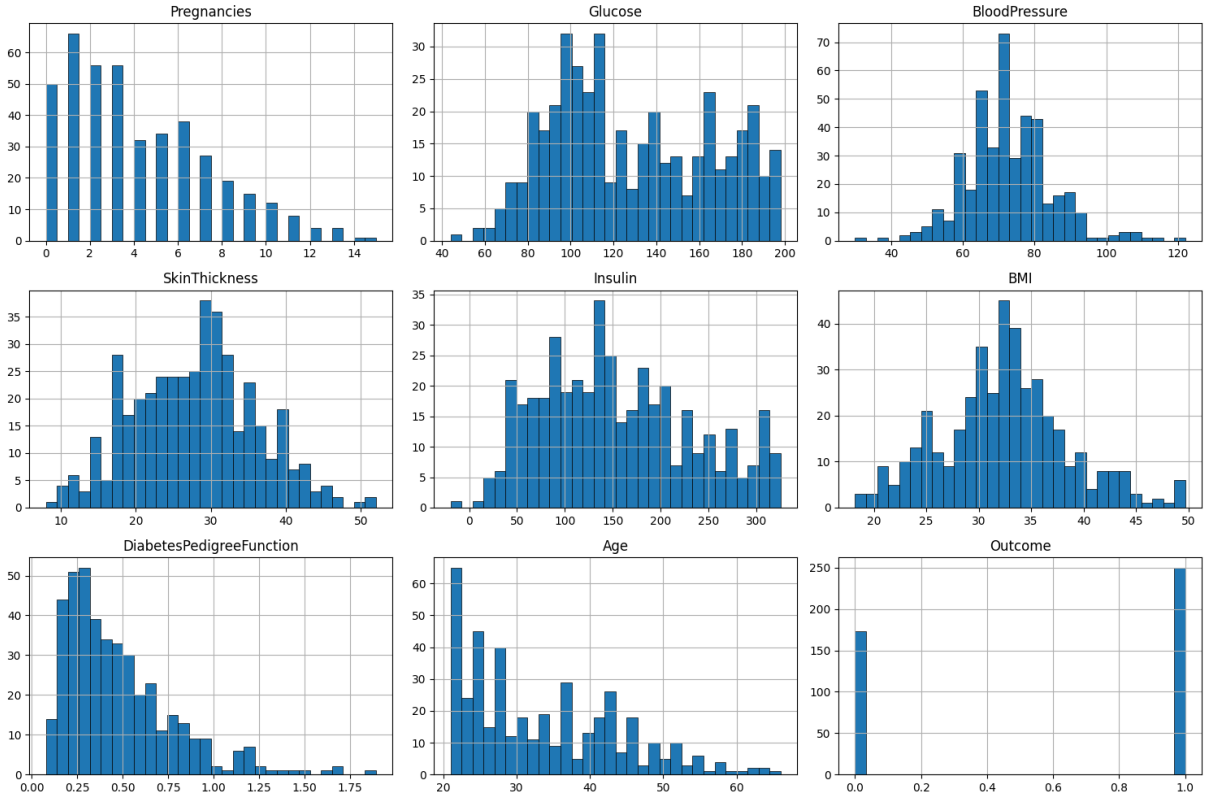


(a) Pre-Imputation Outlier Removed (w/o FS, SMOTEENN Train)

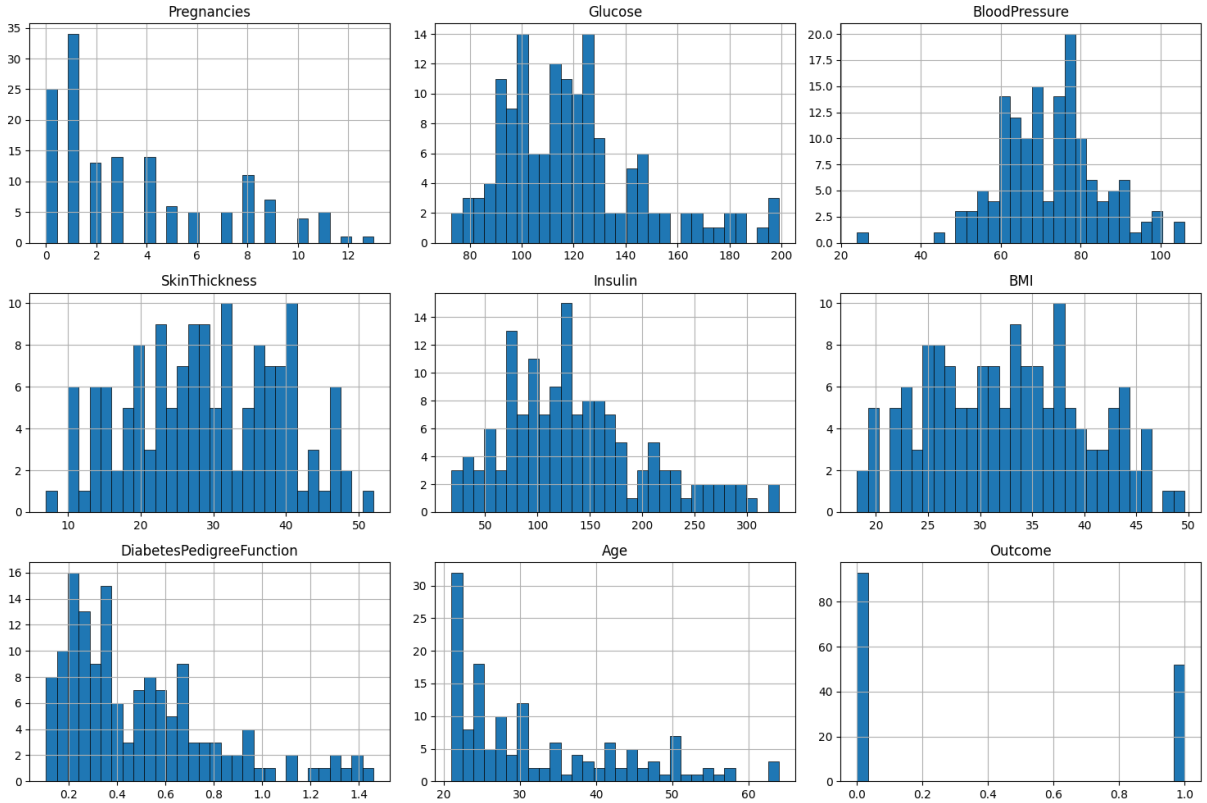


(b) Pre-Imputation Outlier Removed (w/o FS, SMOTEENN Test)

Figure 4.66: Histograms demonstrating the feature distributions of the PIMA indian dataset (Part 2): Pre-imputation data after applying SMOTEENN for training and testing.



(a) Post-Imputation Outlier Removed (w/o FS, SMOTEENN Train)



(b) Post-Imputation Outlier Removed (w/o FS, Test)

Figure 4.67: Histograms demonstrating the feature distributions of the PIMA indian dataset (Part 3): Post-imputation modeling with SMOTEENN for training and testing data.

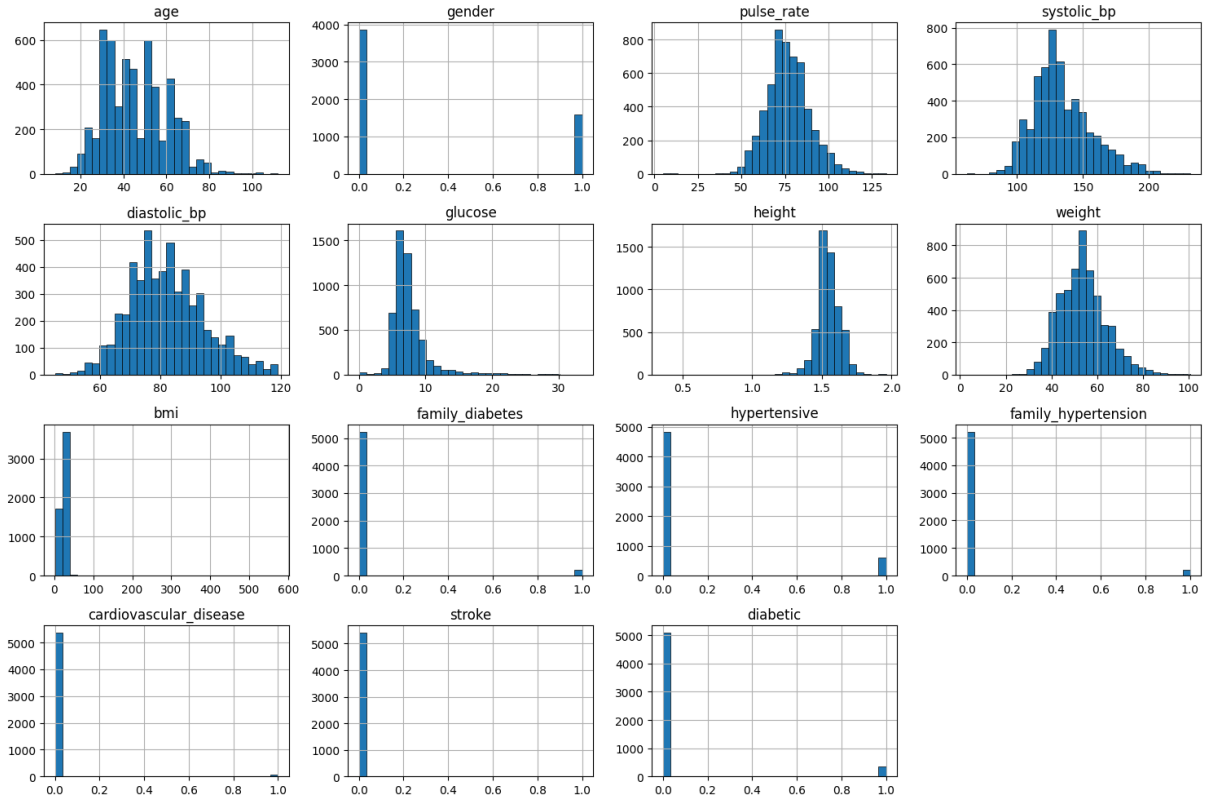
For PIMA indian dataset, the raw dataset was originally imbalanced, with the non-diabetic class larger than the diabetic class. In the post-split pipeline, the selected resampling technique was applied only to the training set, which made the training distribution more balanced, while the test set remained as it is. This is methodologically appropriate because the model learns from better-represented training data but is still evaluated on data that reflects the original problem. In contrast, in the pre-split pipeline, the class distribution was altered before the train-test split. Due to the changes made in the original dataset, the training and test sets become more balanced. Thus making assessment procedures easier than they should be. Another important observation from the distribution graphs is that the post-split pipeline preserves the natural class imbalance ratio on the test set, whereas the pre-split pipeline alters it. This visual difference is significant because it shows that the post-split method keeps evaluation closer to the original data distribution, while the pre-split method changes the evaluation setting itself.

How the Pre-Split Method Artificially Inflates Evaluation Metrics

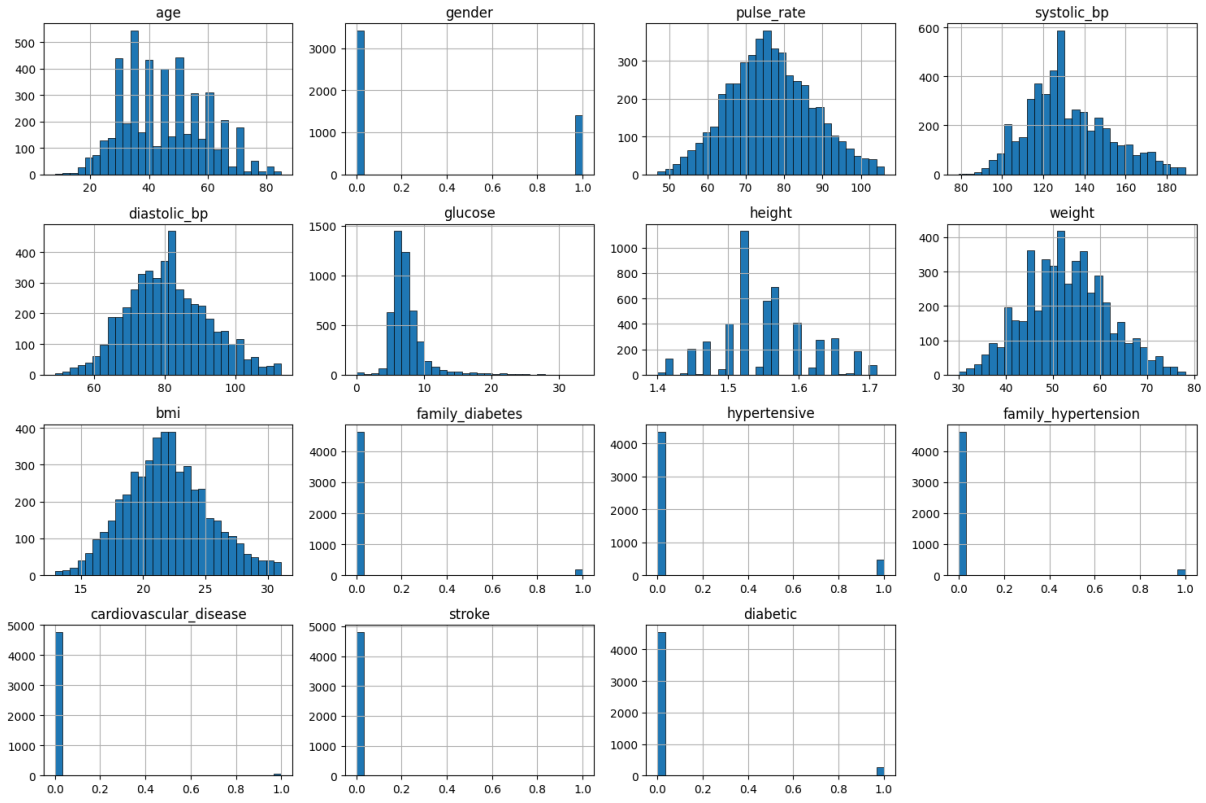
By balancing the test set, the pre-split method changes the math behind standard evaluation metrics. This creates a false sense of high performance based on how True Positives (TP) and False Positives (FP) work:

- **The Post-Split Reality (Lower, but Accurate Metrics):** In the natural imbalanced environment, healthy patients outnumber diabetic patients. Because the healthy group is so large, a small error rate creates a massive number of False Positives. This reduces the precision score, which depends on the ratio of True Positives to all positive guesses, and lowers the F1-Score. The model faces a real-world scenario, finding more diabetic patients (True Positives) means making more false predictions (False Positives). The scores look lower here, but they give an honest picture of how the model handles a real patient population.
- **The Pre-Split Illusion (Higher, but Misleading Metrics):** When data is balanced before splitting, the test set loses its natural imbalance. The number of diabetic patients increases, and the number of healthy patients decreases. When tested in this altered setup, the model produces fewer False Positives, while True Positives become more common. Without a large number of False Positives bringing the scores down, the Precision, F1-Score, and overall Accuracy increase. This makes the model better, but it does not perform better in the real world.

The pre-split pipeline achieves better results because of the evaluation environment, which has been manipulated to create false positive results that do not reflect actual performance results. Therefore, the graphs not only illustrate a technical difference in preprocessing; they also provide visual support for why the post-split workflow is methodologically more reliable.

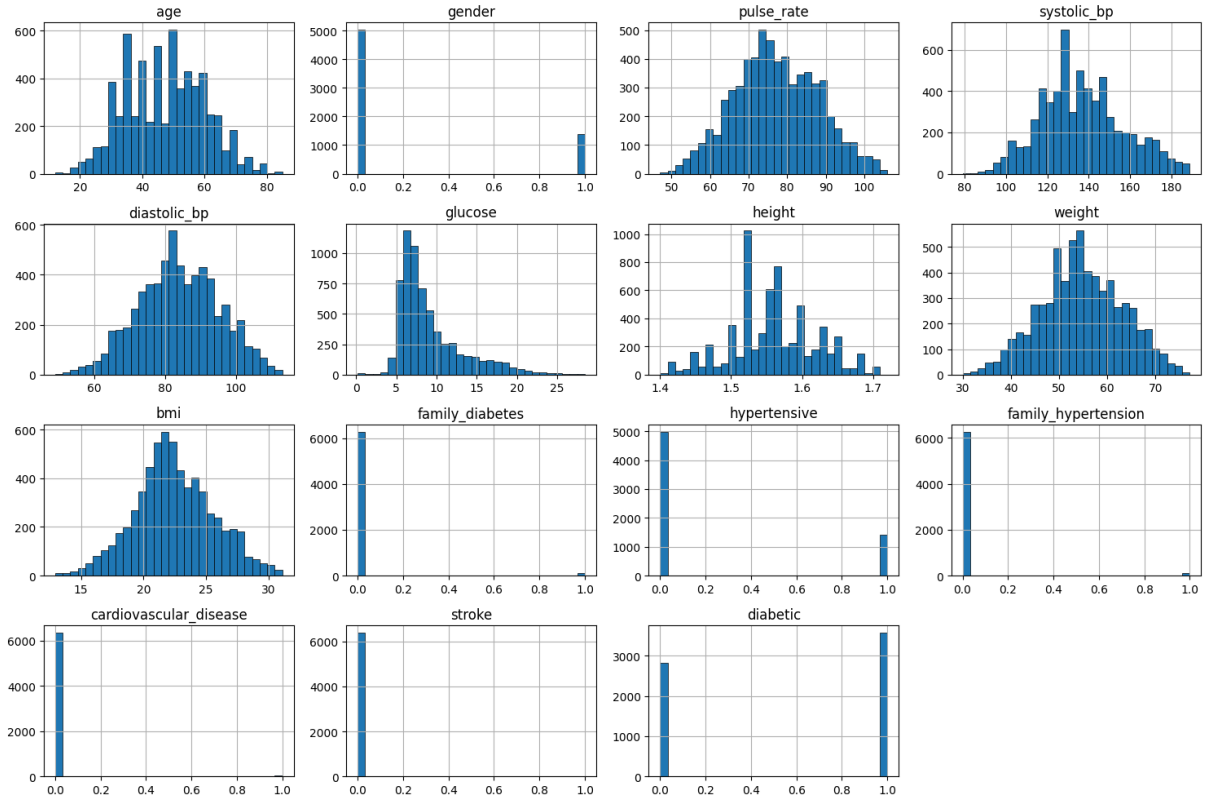


(a) Raw Class Distribution

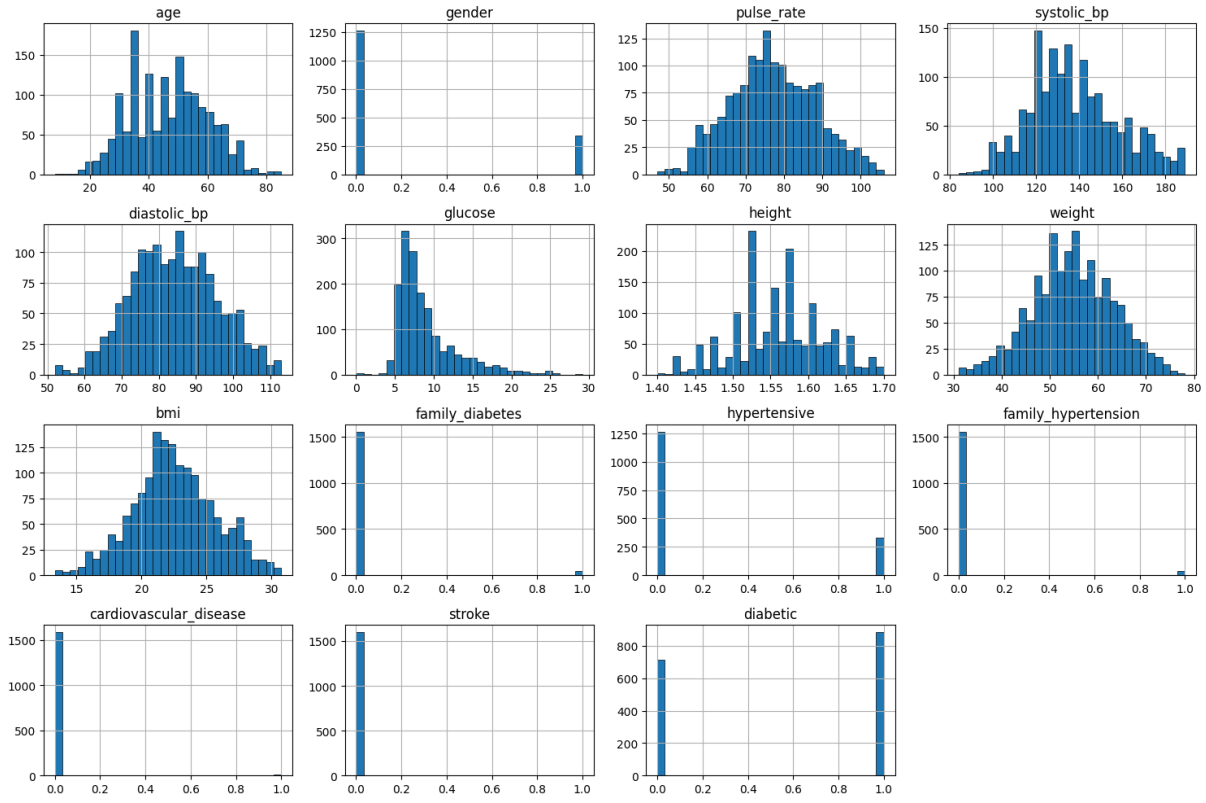


(b) After Outlier Removal

Figure 4.68: Histograms demonstrating the feature distributions of the DiaHealth dataset (Part 1): Raw distribution and after outlier removal.

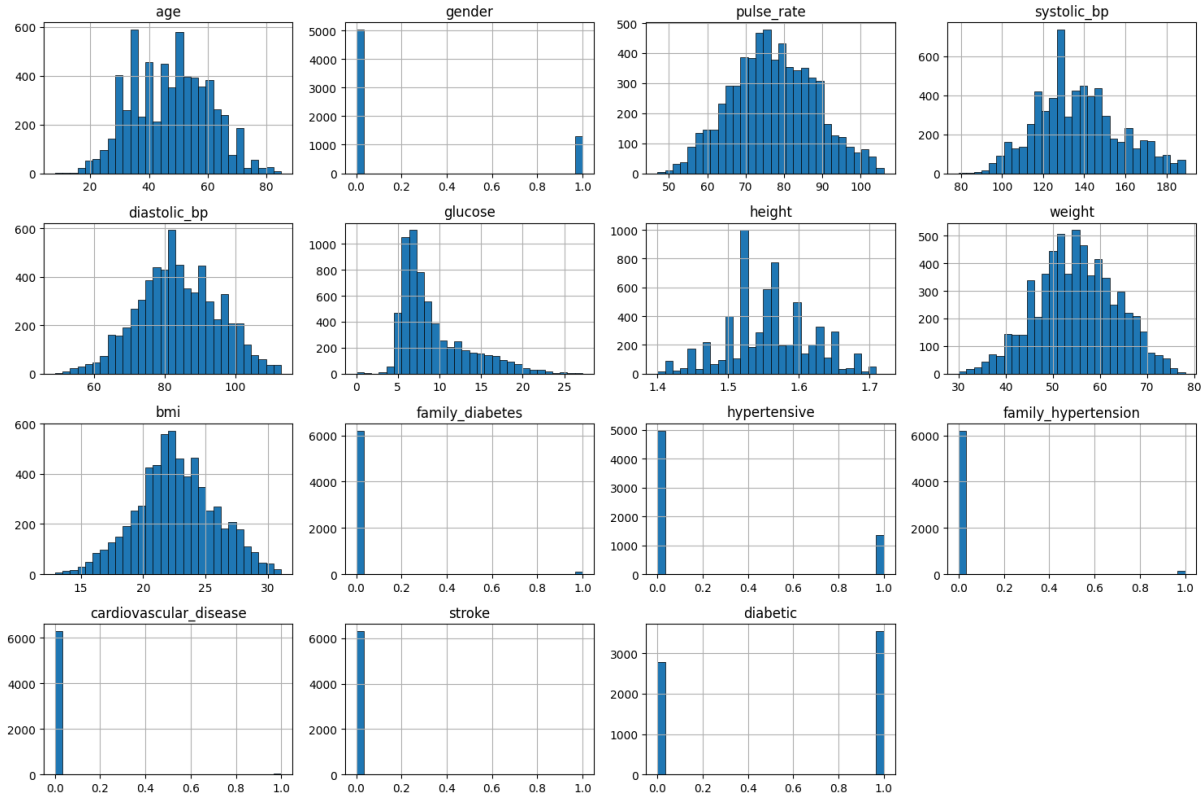


(a) Pre-Split Outlier Removed (w/o FS, SMOTEENN Train)

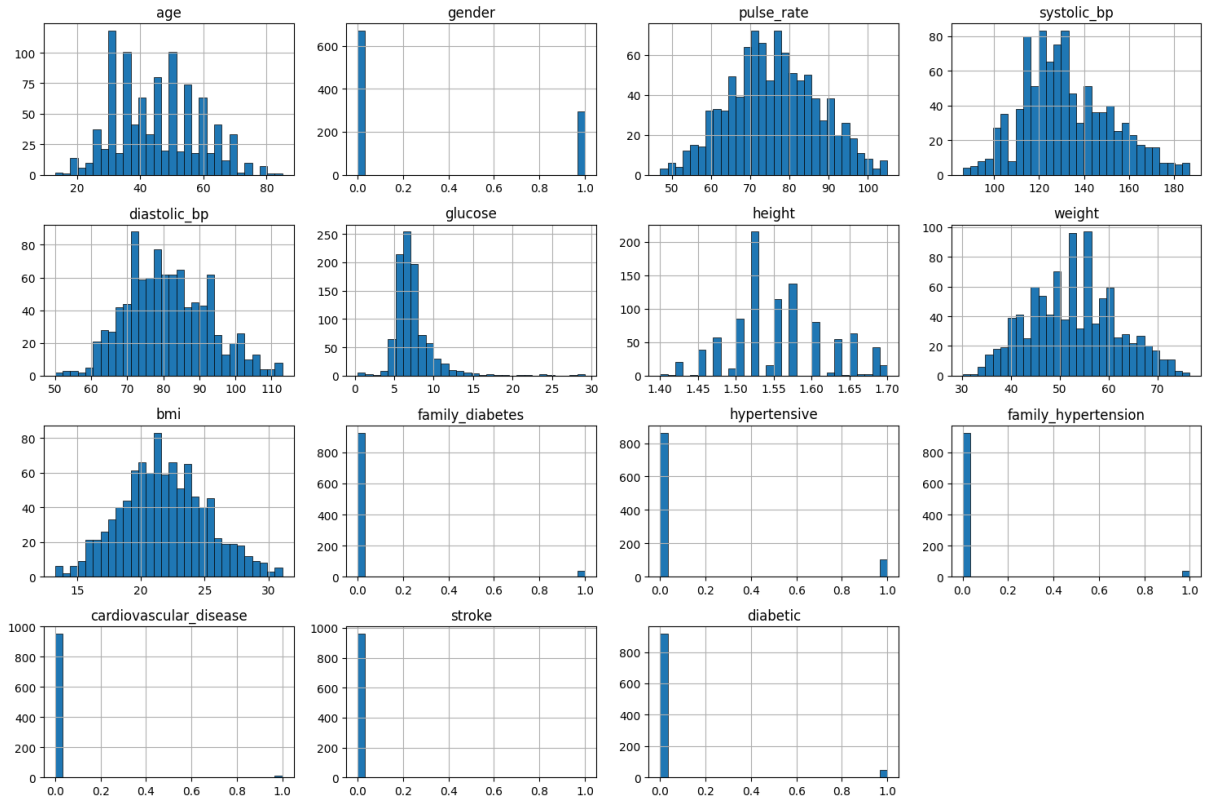


(b) Pre-Split Outlier Removed (w/o FS, SMOTEENN Test)

Figure 4.69: Histograms demonstrating the feature distributions of the DiaHealth dataset (Part 2): Pre-split data after applying SMOTEENN for training and testing.



(a) Post-Split Outlier Removed (w/o FS, SMOTEENN Train)



(b) Post-Split Outlier Removed (w/o FS, Test)

Figure 4.70: Histograms demonstrating the feature distributions of the DiaHealth dataset (Part 3): Post-split modeling with SMOTEENN for training data and preserved testing data.

A similar pattern was observed for the DiaHealth dataset. The raw dataset was also imbalanced, with the majority class clearly larger than the minority diabetic class. In the post-split pipeline, resampling was applied only to the training set, so the training distribution became more balanced while the test set preserved the original imbalance. This keeps evaluation closer to a real-world scenario, where diabetic cases remain less frequent than non-diabetic cases. However, in the pre-split pipeline, the balancing step affected the data before splitting, which again caused the test set distribution to move away from the raw dataset. As a result, the test set became less representative of the original DiaHealth population. From a research perspective, this distribution shift is highly important. The main issue is not only that pre-split resampling may improve performance artificially, but also that it changes the evaluation environment itself. A model tested on an already modified class distribution is not being judged under the same level of difficulty as a model tested on the natural distribution. In other words, the pre-split pipeline reduces the realism of the test set, while the post-split pipeline preserves it. This strengthens the methodological conclusion of this study: post-split resampling is more valid because it improves class balance only in the training data and leaves the test data untouched, allowing a fairer and more realistic assessment of model behavior.

4.6 SHAP-Based Feature Importance Analysis

To answer RQ-3, we used SHAP to understand which features contributed most to the model decisions, especially for the minority class (diabetes positive). Since this study compares pre-split resampling and post-split resampling, SHAP also helps us understand whether the models trained under the wrong and correct pipeline are actually learning the same clinical pattern or not. To put it another way, this section is not only about identifying important features, but also about checking whether the important features are stable, clinically meaningful, and methodologically trustworthy or not.

Across all three datasets, one common pattern was observed. The pre-split pipelines usually showed stronger and cleaner SHAP patterns, but these explanations were also more likely to be inflated because the resampling was done before the train-test split. On the other hand, the post-split pipelines often produced less extreme feature importance patterns, but those patterns were more realistic and methodologically more acceptable.

However, the degree of clinical usefulness was not the same for all datasets. In PIMA indian dataset, the post-split SHAP results were much more clinically aligned. In Diahealth, even the post-split explanations remained somewhat unstable and relied on several indirect predictors. In BRFSS-2015, the post-split models produced a broader public-health risk pattern rather than a direct biomarker-based explanation, which is still acceptable considering the nature of the dataset.

4.6.1 PIMA indian dataset SHAP Analysis

For the PIMA indian dataset, we have analyzed a total of sixteen combinations including eight pre-split pipelines and eight post-split pipelines which were mentioned in the earlier sections. The purpose of this analysis was to see whether the same disease-related features remained important across different preprocessing choices, and more importantly, whether the wrong pre-split pipeline and the correct post-split pipeline were actually learning diabetes in the same way. Also, if the predictions of the models align with clinical predictors or not.

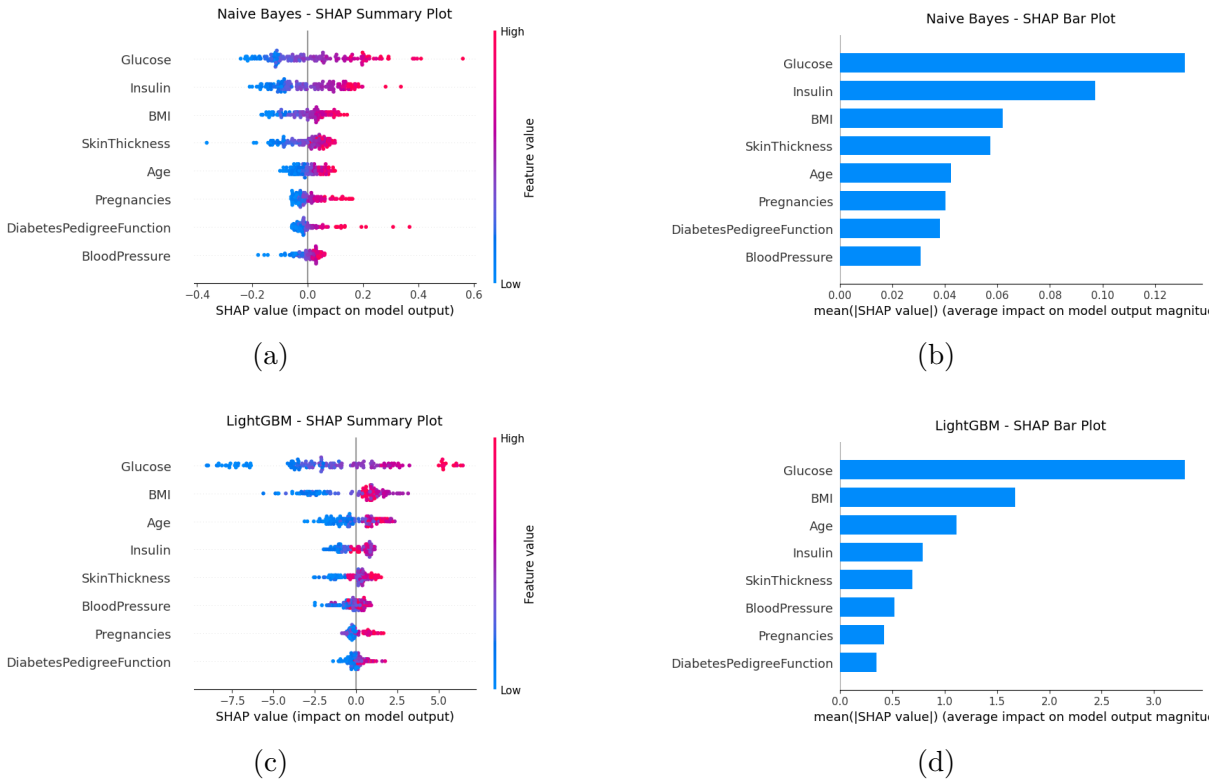


Figure 4.71: Representative SHAP summary plots illustrating feature importance across the entire population of the PIMA indian dataset: (a) pre-split beeswarm plot, (b) pre-split bar plot, (c) post-split beeswarm plot, and (d) post-split bar plot.

From the SHAP graphs, a very clear pattern can be seen. In the post-split pipelines, the models mostly relied on Glucose as the main feature, and then gave importance to other clinically meaningful variables such as Age, BMI, and in some cases DiabetesPedigreeFunction. This is quite consistent with clinical understanding, because glucose level is one of the most direct indicators of diabetes, while age, obesity-related factors, and family history-related risk are also well-known contributors.

In contrast, the pre-split pipelines were much more centered around Glucose and Insulin, with Insulin repeatedly becoming the second most dominant feature. This does not mean that Insulin is irrelevant, but its contribution appeared unusually strong and too consistent in the pre-split setting, which raises concern.

Table 4.36: Dominant feature comparison across the eight matched pre/post pairs (PIDD)

Configuration	**Post-split dominant features**	**Pre-split dominant features**
Imputation, outlier removed, no feature selection	Glucose > BMI > Age > DPF	Glucose > Insulin > BMI > Age
Imputation, outlier removed, feature selection	Insulin > Age > Glucose > BMI	Glucose > Insulin > BMI > Age
Imputation, with outlier, no feature selection	Glucose > BMI > Age > DPF	Glucose > Insulin > Age > BMI
Imputation, with outlier, feature selection	Glucose > BMI > Age > Insulin	Glucose > Age > Insulin > BMI
Elimination, outlier removed, no feature selection	Glucose > Age > BMI > SkinThickness	Glucose > Age > Insulin > SkinThickness
Elimination, outlier removed, feature selection	Glucose > Age > Insulin > BMI	Glucose > Insulin > Age > SkinThickness
Elimination, with outlier, no feature selection	Glucose > Age > DPF > BMI	Glucose > Insulin > BMI > Age
Elimination, with outlier, feature selection	Glucose > Age > Insulin > BMI	Glucose > Insulin > SkinThickness > Age

The feature stability analysis supports the same conclusion. The post-split configuration showed consistent dominant features which remained unchanged during both imputation and elimination settings. The models developed diabetes risk assessment capabilities which extended beyond their primary pattern because they learned multiple risk factors which included glycemic burden and age-related and obesity-related and family history-related risks. In the pre-split configuration, the rankings below Glucose were less stable and repeatedly clustered around Insulin. In some settings BMI was third, in others Age was third, and in one case SkinThickness moved unusually high. This suggests that the explanation in the pre-split setting is more sensitive to the leakage-prone workflow than to a stable underlying disease pattern.

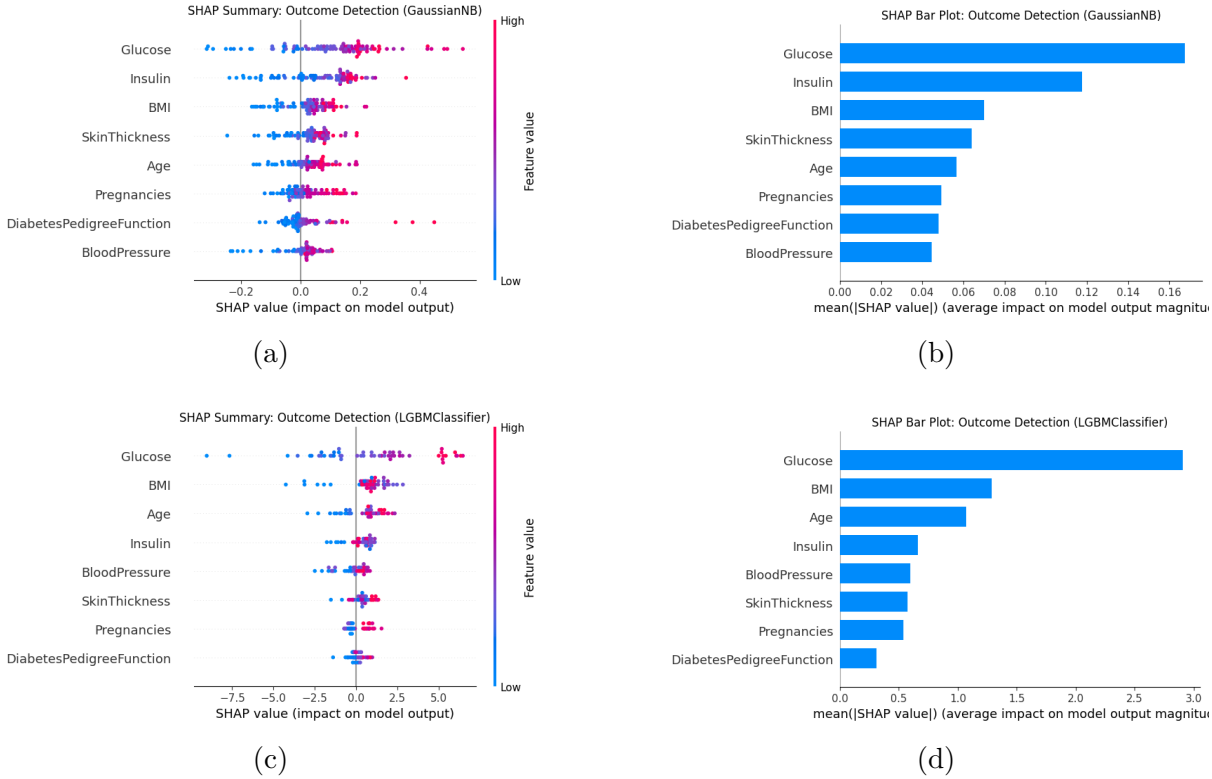


Figure 4.72: Representative SHAP summary plots illustrating feature importance specifically for the diabetic minority class in the PIMA indian dataset: (a) pre-split beeswarm plot, (b) pre-split bar plot, (c) post-split beeswarm plot, and (d) post-split bar plot.

The minority-class SHAP analysis makes this even clearer. Since the diabetic class is the main class of interest in this study, this part is especially important. In the post-split minority-class SHAP plots, Glucose was the top feature in 7 out of 8 configurations, while Age appeared in the top three in all 8 configurations. The remaining importance was shared among BMI, Insulin, and sometimes DiabetesPedigreeFunction. This means that when the corrected post-split models identified diabetic cases, they mostly did so using a combination of glycemic level, age-related vulnerability, and metabolic/body-composition risk, which is clinically understandable. This is also why the most suitable figure for PIMA indian dataset should include both an overall post-split SHAP bar plot and a minority-class post-split SHAP bar plot, ideally taken from one of the clearer corrected settings where Glucose, BMI, Age, and DiabetesPedigreeFunction appear in a clinically sensible order. The pre-split minority-class SHAP plots told a somewhat different story. Glucose remained important, but Insulin appeared in the top three in 7 out of the 8 configurations and was very often the second strongest feature. This matters because minority-class oversampling performed before the split can make the diabetic class look more separable than

it really is. As a result, the features that help create that artificial separability may also receive inflated SHAP importance. So, the problem is not only that pre-split resampling inflates performance metrics; it also changes the explanation pattern for diabetic cases. From a clinical perspective, the post-split PIMA indian dataset SHAP results are clearly more acceptable. In this dataset, it is expected that Glucose will be the strongest feature. It is also reasonable for BMI and Age to remain important, because obesity and increasing age are major diabetes risk factors. In some no-feature-selection settings, DiabetesPedigreeFunction also contributes meaningfully, which is again clinically reasonable because family-history-related risk has a known role in diabetes. The pre-split explanations are not entirely clinically wrong, because Insulin is still related to diabetes physiology. However, the issue is not the presence of Insulin itself, but the degree of dominance given to it. In PIMA indian dataset, Insulin is a difficult variable because it is strongly affected by missingness and preprocessing. So when a pipeline repeatedly elevates it to second position across nearly all settings, especially together with near-perfect performance, that pattern becomes methodologically suspicious. Overall, the PIMA indian dataset SHAP analysis shows that the post-split corrected pipeline provides the more reliable explanation set. Both pipeline families agree that Glucose is the central feature, which confirms that the main diabetes signal is preserved. However, beyond that shared point, the pre-split models consistently overemphasize Insulin, produce more inflated performance, and show less stable explanations. The post-split models, on the other hand, keep Glucose as the dominant feature while distributing importance more plausibly across Age, BMI, and sometimes DiabetesPedigreeFunction. Therefore, for PIMA indian dataset, the SHAP results support the conclusion that the post-split explanations should be treated as the main interpretation, while the pre-split explanations should be discussed as inflated and methodologically compromised.

4.6.2 Diahealth SHAP Analysis

For the Diahealth dataset, 8 configurations were analyzed, including 4 pre-split and 4 post-split pipelines. These configurations covered the four combinations of outlier removed/with outlier and with/without feature selection. Unlike PIMA indian dataset, there were no separate imputation and elimination branches for this dataset. So, the main purpose here was to examine whether the models learned a stable and clinically meaningful diabetes pattern under the corrected workflow, and whether the pre-split setup changed the feature importance in a misleading way.

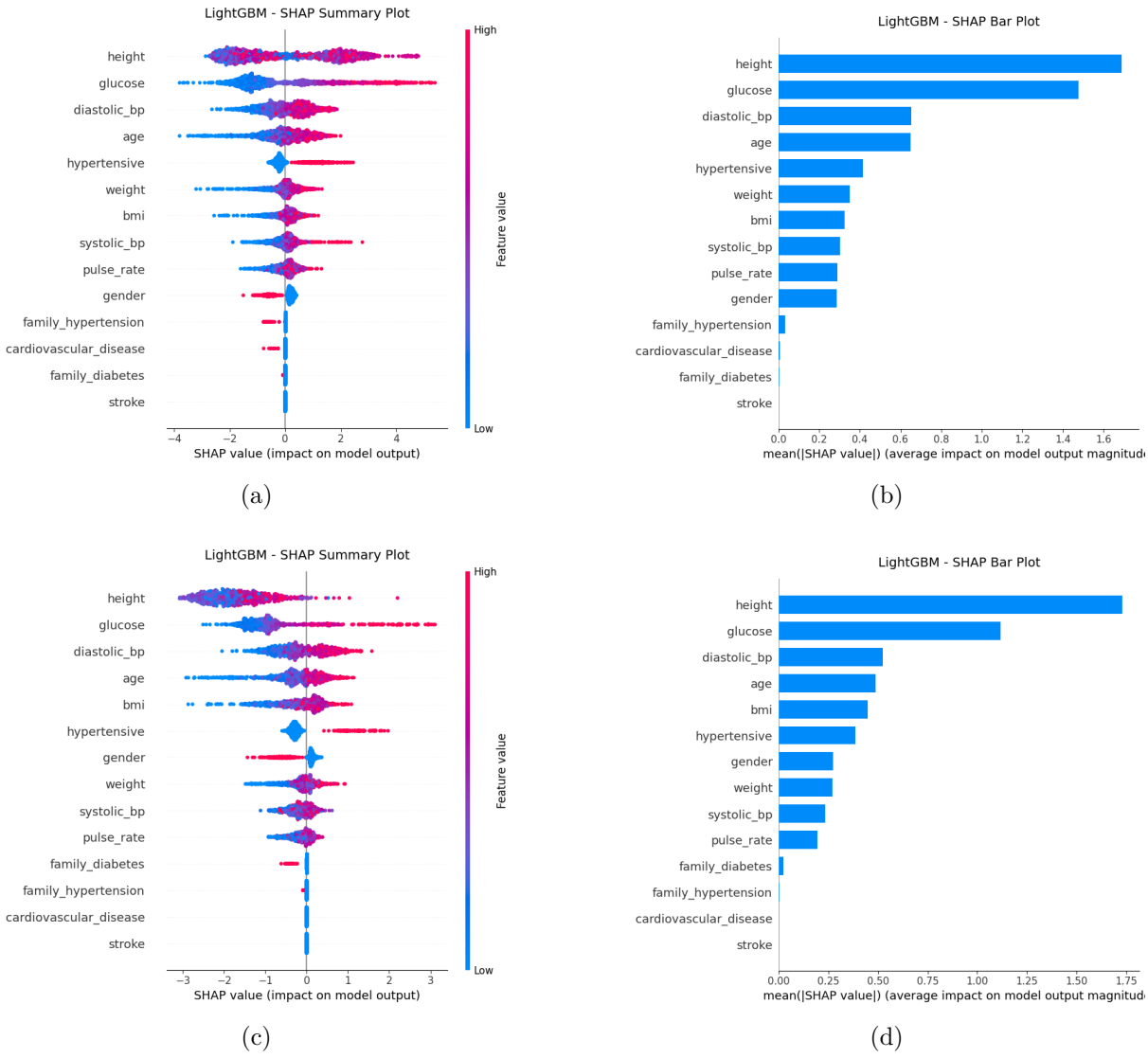


Figure 4.73: Representative SHAP summary plots illustrating feature importance across the entire population of the Diahealth dataset: (a) pre-split beeswarm plot, (b) pre-split bar plot, (c) post-split beeswarm plot, and (d) post-split bar plot.

The SHAP results for Diahealth looked quite different from PIDD. In PIMA indian dataset, the post-split models showed a clear and clinically expected glucose-centered pattern. In Diahealth, however, the SHAP plots suggested that the models were learning from a mixture of direct and indirect predictors. In several settings, features such as height, diastolic blood pressure, weight, and the binary hypertensive variable became highly important. Some of these features can be related to diabetes risk in an indirect way, but they are not the kind of primary diabetes indicators we would normally expect to dominate the explanation. This makes the Diahealth SHAP findings more cautious to interpret.

Table 4.37: Dominant feature comparison across the four matched pre/post pairs (Diahealth)

Configuration	**Post-split dominant features**	**Pre-split dominant features**
Outlier removed, no feature selection	height > glucose > age > diastolic_bp > bmi	height > glucose > diastolic_bp > age > hypertensive
Outlier removed, feature selection	glucose > age > diastolic_bp > weight > bmi	glucose > age > diastolic_bp > weight > systolic_bp
With outlier, no feature selection	glucose > height > age > systolic_bp > diastolic_bp	height > glucose > age > systolic_bp > diastolic_bp
With outlier, feature selection	glucose > age > hypertensive > weight > systolic_bp	glucose > age > diastolic_bp > weight > systolic_bp

When the pre and post-pipelines were compared, the post-split models generally looked more glucose-centered, while the pre-split models more often gave unusually high importance to height or blood-pressure-related variables. In the outlier removed without feature selection setting, both pipelines placed height very high, which already suggests that the dataset itself may contain some noisy or weak signal. In the feature selection settings, Glucose became more dominant, especially in the post-split models, but blood pressure and weight still remained strong contributors. So, compared to PIMA indian dataset, the Diahealth models appear less stable and less disease-specific.

The feature stability analysis supports that view. In the post-split overall SHAP plots, glucose and age were the most stable features, appearing in the top three across all four configurations. But after these two, the ranking changed noticeably from one setting to another. In some cases height was near the top, in others, diastolic blood pressure became strong, and in one setting the hypertensive variable also became important. This suggests that even the corrected post-split models were not learning one clean and consistent diabetes-specific pattern. Instead, they were moving between glycemic, anthropometric, and blood-pressure-related explanations depending on the preprocessing setting.

The pre-split Diahealth configurations were even less convincing. Although glucose remained important in every case, the repeated dominance of height is particularly difficult to justify clinically. Height can interact indirectly with body composition or other health measures, but it is not usually expected to dominate diabetes prediction in the same way as glucose, BMI, age, or stronger obesity-related variables. So when SHAP repeatedly places height at or near the top, especially under a leakage-prone pipeline, it suggests that the model may be learning proxy relationships, dataset-specific structure, or noise, rather than a clean disease mechanism.

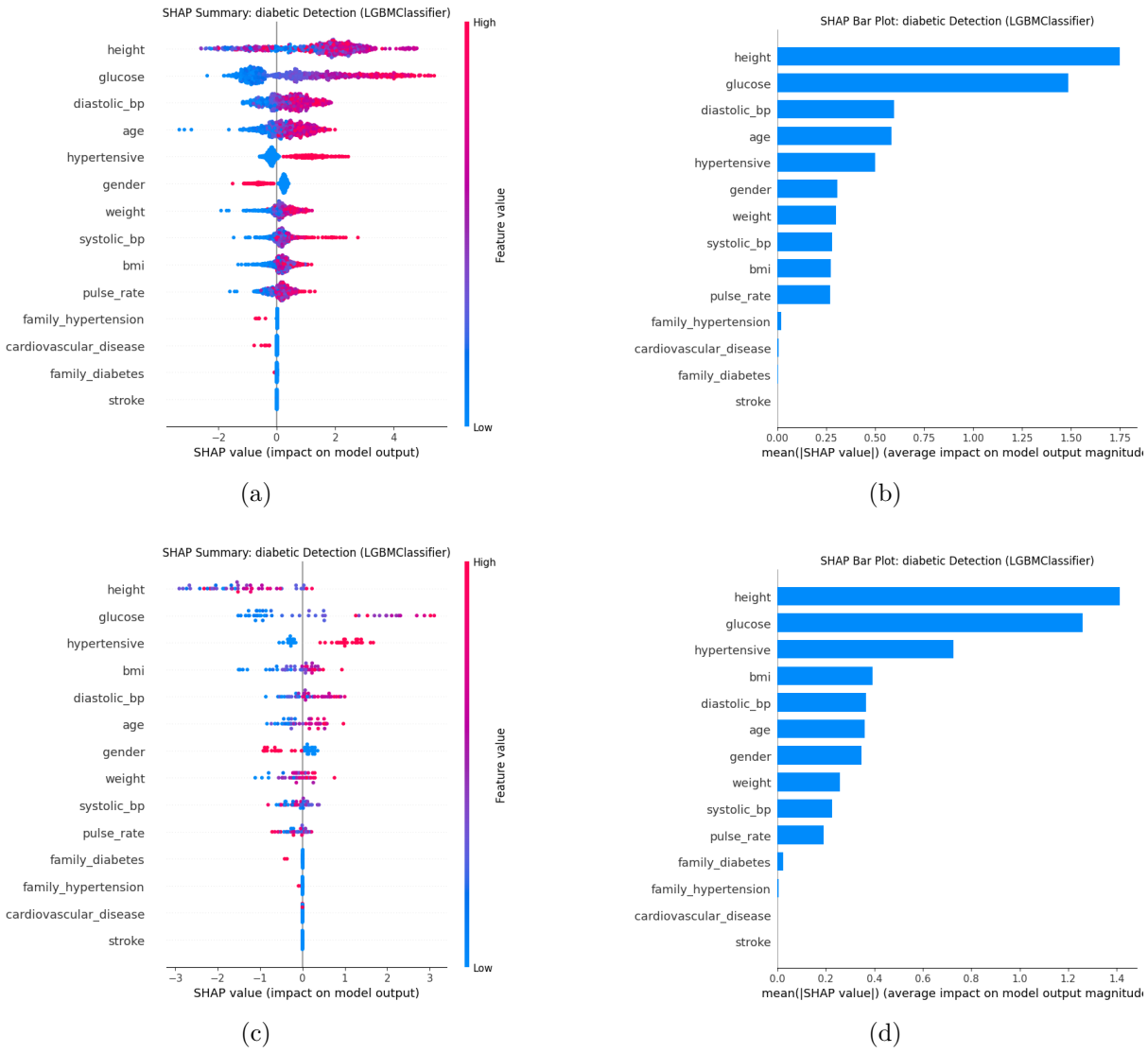


Figure 4.74: Representative SHAP summary plots illustrating feature importance specifically for the diabetic minority class in the Diahealth dataset: (a) pre-split beeswarm plot, (b) pre-split bar plot, (c) post-split beeswarm plot, and (d) post-split bar plot.

The minority-class SHAP analysis gives a slightly better picture for the corrected Diahealth pipeline. In the post-split minority-class plots, glucose was the top feature in all four configurations, which is more reassuring than what was seen in the overall plots. This means that when the corrected models identified diabetic cases, they at least relied first on the most clinically meaningful variable available in the dataset. However, the supporting features still remained mixed, including height, hypertensive status, weight, age, and diastolic blood pressure. So even though the minority-class explanation is better than the overall explanation, it still depends on a combination of direct and indirect signals. Because of that, the most suitable figure for Diahealth is a single post-split minority-class SHAP bar plot, since this figure best supports the key finding that the corrected model restores Glucose to the first position for diabetic cases, even though the dataset still contains several proxy-like signals.

The pre-split minority-class SHAP plots were again less clinically convincing. Glucose remained important, but height became the top feature in two of the four configurations, and diastolic blood pressure entered the top three in three of the four configurations. This means that the pre-split models were not simply detecting diabetes through glucose and broader metabolic burden. Instead, they were often explaining diabetic predictions

through a less reliable combination such as height + glucose or glucose + diastolic blood pressure, which again suggests that the feature importance was being distorted by the workflow and possibly also by the dataset structure itself.

From a clinical acceptance point of view, the Diahealth results can only be described as partially acceptable. The importance of glucose and age makes sense. The importance of BMI, weight, and hypertension-related features is also understandable, since diabetes often overlaps with obesity and cardiometabolic risk. But the repeated high ranking of height is harder to accept clinically. It may reflect a proxy effect, an interaction with weight or BMI, a measurement pattern, or simply weaker data quality. So, while the Diahealth SHAP results are not meaningless, they do not show the same strong clinical clarity that was seen in PIDD.

Since SHAP explains whatever model it is given, the explanations produced from these pre-split models also become unrealistically clean and strong. In Diahealth, that matters even more because the underlying feature space already seems somewhat noisy.

Therefore, the final interpretation for Diahealth is more cautious than for PIDD. The post-split pipelines should still be treated as the valid basis for interpretation, because they are methodologically more trustworthy and they restore Glucose to the dominant position in the minority-class analysis. However, even these post-split SHAP results remain only moderately clinically acceptable, because the models still rely heavily on indirect variables such as height, blood pressure, weight, and hypertensive status. So, the main story in Diahealth is twofold: first, the pre-split SHAP results are methodologically compromised, and second, the dataset itself seems to have a weaker and noisier diabetes-specific signal, which limits the clarity of the final explanation.

4.6.3 BRFSS-2015 SHAP Analysis

For the BRFSS-2015 dataset, 8 configurations were analyzed, including 4 pre-split and 4 post-split pipelines, covering the four combinations of outlier removed/with outlier and with/without feature selection. As in the previous datasets, the SHAP interpretation was based on the best-performing tree-based model from each configuration, selected mainly using MCC and then checked with AUC-ROC. The main purpose here was to see whether the corrected post-split workflow produced more stable and clinically believable feature importance, and whether the pre-split workflow again inflated the explanations.

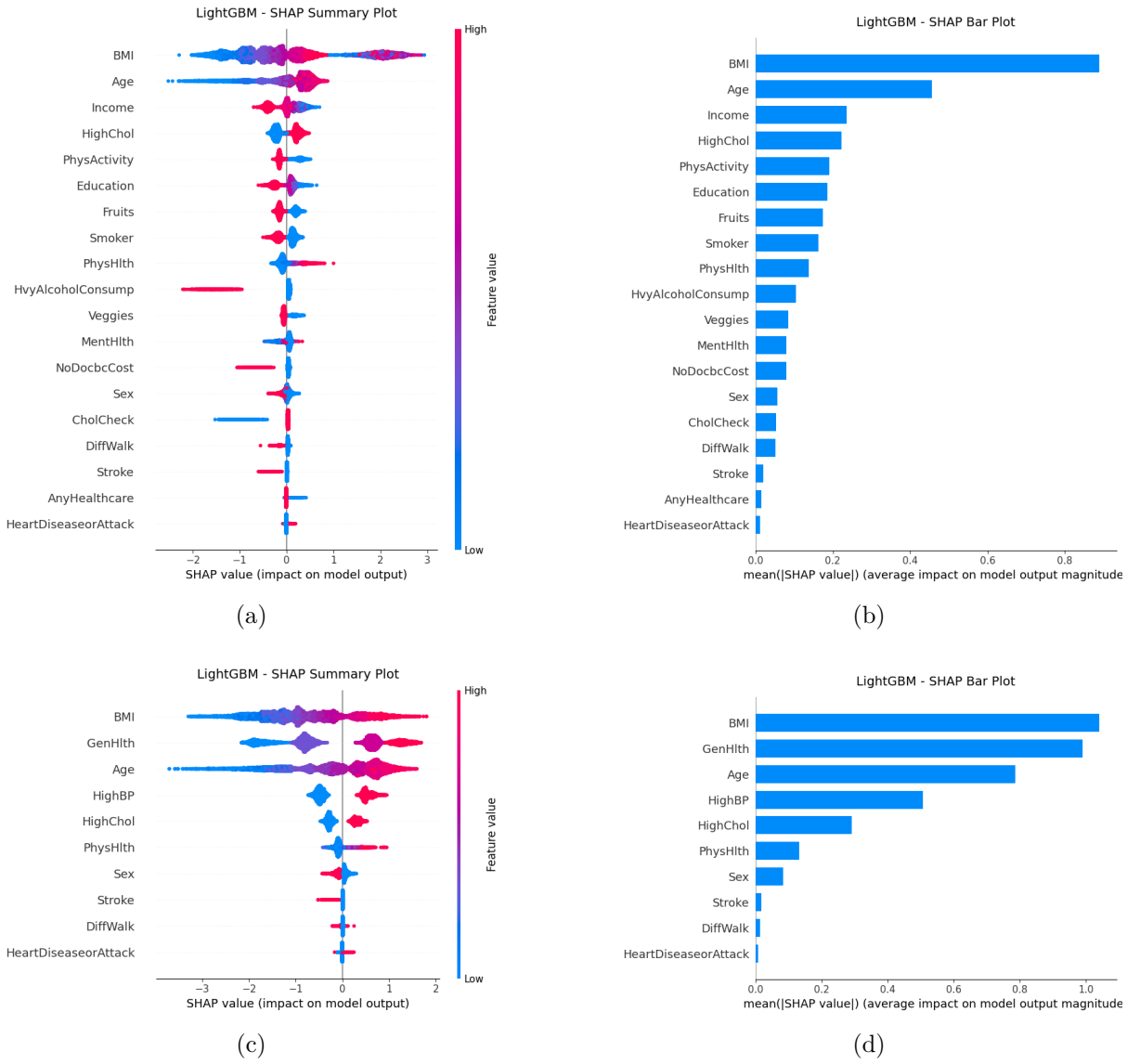


Figure 4.75: Representative SHAP summary plots illustrating feature importance across the entire population of the BRFSS-2015 dataset: (a) pre-split beeswarm plot, (b) pre-split bar plot, (c) post-split beeswarm plot, and (d) post-split bar plot.

The BRFSS-2015 results were different from both PIMA indian dataset and Diahealth. This dataset does not contain direct laboratory diabetes variables like Glucose. Instead, it is built mostly from survey-based and self-reported health features. Because of that, the SHAP explanations naturally reflect a broader public-health diabetes risk pattern rather than a direct disease-biomarker pattern. In the post-split SHAP plots, the dominant features were usually BMI, GenHlth, and Age, followed by HighBP and HighChol. This is quite reasonable for BRFSS, because obesity burden, general health condition, age, blood pressure, and cholesterol are all well-known population-level diabetes risk factors.

Table 4.38: Dominant feature comparison across the four matched pre/post pairs (BRFSS-2015)

Configuration	**Post-split dominant features**	**Pre-split dominant features**
Outlier removed, no feature selection	BMI > GenHlth > Age > HighBP > HighChol	BMI > Age > Income > HighChol > PhysHlth
Outlier removed, feature selection	GenHlth > Age > BMI > HighBP > HighChol	GenHlth > BMI > PhysHlth > Age > HighBP
With outlier, no feature selection	BMI > GenHlth > Age > HighBP > Income	BMI > Age > Income > HighChol > PhysHlth
With outlier, feature selection	GenHlth > BMI > Age > HighBP > HighChol	BMI > GenHlth > PhysHlth > Age > HighBP

When the pre and post results were compared, the post-split models looked more stable and more focused on an expected cardiometabolic risk profile. In the outlier removed without feature selection setting, the post-split model emphasized BMI, GenHlth, Age, HighBP, and HighChol, while the pre-split model still kept BMI high but shifted more importance toward Income and PhysHlth. In the feature-selection settings, both pipelines remained somewhat clinically meaningful, but the post-split models were more balanced and less dependent on burden-related or contextual variables. The same general pattern also remained in the with outlier settings. So overall, the post-split BRFSS pipelines seem to capture a more coherent risk pattern, while the pre-split pipelines become more mixed with indirect or contextual features.

The feature stability analysis also supports this interpretation. In the post-split overall SHAP plots, BMI, GenHlth, and Age stayed within the top three across all four configurations. HighBP and HighChol also appeared repeatedly just below them. This suggests that the corrected BRFSS models were repeatedly learning a fairly stable obesity and cardiometabolic burden profile, which is suitable for this kind of survey dataset. In contrast, the pre-split configurations were less stable. BMI still remained important, and GenHlth was also often high, but the next positions shifted more often toward Income and PhysHlth. These variables are not irrelevant, because diabetes is influenced by social conditions and overall physical burden as well. But when they repeatedly become too dominant, especially under a leakage-prone pipeline, they become less convincing as the main explanation. This is an important difference between BRFSS and the other two datasets. For BRFSS, it is acceptable that the models do not focus on direct biological markers, because the dataset itself is not biomarker-based. Instead, what matters is whether the explanations stay centered on a meaningful population-level diabetes risk structure. In the post-split configurations, that condition is mostly met. In the pre-split configurations, the explanation starts to shift too often toward indirect burden and socioeconomic variables, which weakens interpretability.

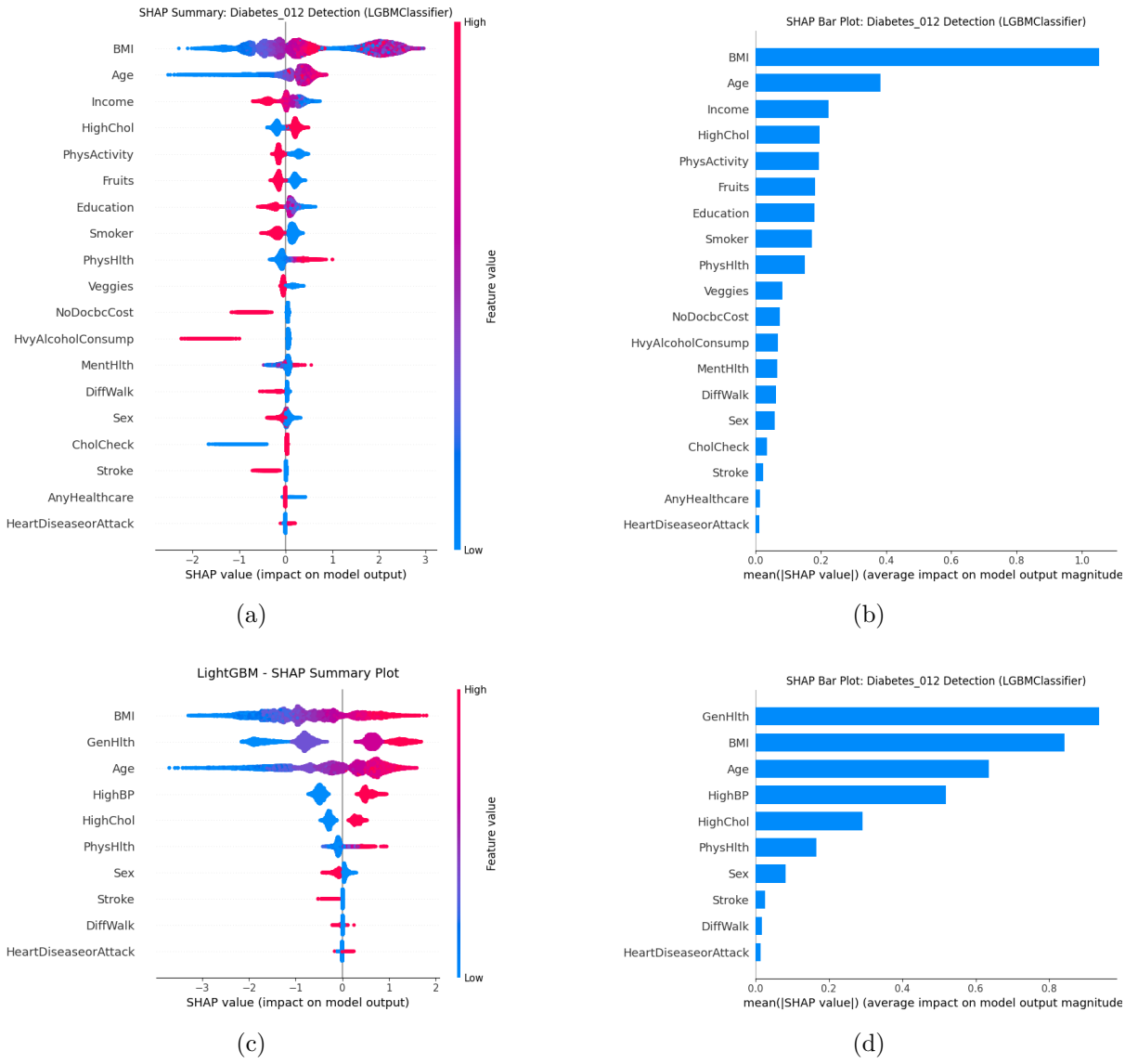


Figure 4.76: Representative SHAP summary plots illustrating feature importance specifically for the diabetic minority class in the BRFSS-2015 dataset: (a) pre-split beeswarm plot, (b) pre-split bar plot, (c) post-split beeswarm plot, and (d) post-split bar plot.

The minority-class SHAP analysis shows the same pattern more clearly. In the post-split minority-class plots, the diabetic class was usually explained by BMI, GenHlth, and Age, followed by HighBP and HighChol. In the feature-selection settings, GenHlth sometimes became the strongest feature, while in the no-feature-selection settings BMI often remained first. This is still clinically and epidemiologically reasonable for BRFSS, because it reflects obesity, poorer health status, age-related vulnerability, and co-existing cardiometabolic risk. In the pre-split minority-class analysis, however, features like Income and PhysHlth entered the upper ranks more frequently. Again, these are meaningful background factors, but they are less convincing as the main drivers compared with BMI, general health, age, blood pressure, and cholesterol burden. Since the main BRFSS story is the stability of the broad risk-factor structure, the most suitable figure here is a single post-split overall SHAP bar plot, which most clearly shows the corrected model centering its explanation on BMI, GenHlth, and Age.

From a clinical and public-health perspective, the BRFSS SHAP results can be considered fairly acceptable, especially in the post-split setting. The main reason is that BRFSS is a survey-based dataset, not a clinical-lab dataset. Therefore, a feature pattern centered

on BMI, self-reported general health, age, blood pressure, and cholesterol is appropriate in this context. At the same time, features such as Income, Education, or NoDocbcCost should be understood more as contextual or social risk correlates, not as direct biological causes of diabetes. This is why the post-split BRFSS explanations are more acceptable than the pre-split ones: they stay closer to the core cardiometabolic profile and do not drift as strongly toward indirect context-driven variables.

As in the previous datasets, this large gap suggests that the pre-split workflow produced overly optimistic results. Because SHAP explains those models directly, the corresponding feature importance also becomes sharper and more confident than it should be. In contrast, the post-split BRFSS explanations are less extreme but more believable, which is exactly what would be expected from a more valid evaluation pipeline.

Therefore, the final interpretation for BRFSS-2015 is that the post-split SHAP explanations should be taken as the main basis for interpretation. Both pre-split and post-split workflows identify meaningful diabetes-related risk variables, but they do not organize them in the same way. The post-split pipelines are more stable and repeatedly center the explanation on BMI, GenHlth, and Age, supported by HighBP and HighChol, which gives a coherent cardiometabolic risk profile. The pre-split pipelines, on the other hand, still find relevant features, but they appear more inflated and more likely to shift importance toward Income and PhysHlth, especially when feature selection is not applied. So, compared with PIMA indian dataset, BRFSS does not produce a clean biomarker-based explanation, but unlike Diahealth, it still produces a fairly reasonable and interpretable population-level risk pattern in the corrected post-split setting.

5 Discussion of Key Findings

5.1 Interpreting the Impact of Pre-Split Resampling (RQ1)

Our comparative analysis shows a large gap between the pre-split and post-split pipelines across all three datasets (PIMA indian,BRFSS-2015,Diahealth). The models that are trained on pre-split data have achieved exceptionally higher scores, almost in all the metric evaluation scores looked perfect. However, as we questioned at the end of section-4.4, we should ask whether these inflated numbers represent a genuinely smart clinical model or simply an algorithm that has memorized the data. Lets take the PIMA Indian dataset as an example. Under the pre-split configuration, one model hit an accuracy of .9906, with an F1-score of 0.9919 and an MCC of 0.9808 which means nearly perfect prediction all the time. In medical diagnostic contexts, those numbers are extraordinary. When we ran the exact same configuration through the unbiased post-split pipeline, the accuracy dropped to 0.7862, the F1-score fell to 0.74, and the MCC went down to 0.58. This huge gap of 15% to 20% across the key evaluation metrics is not a small statistical variance or a rounding error. It gives an idea about what the model has actually learned. If we want to find why this happens, we will have to understand the mechanics of data leakage. When we apply resampling techniques like SMOTE or SMOTEENN before splitting the dataset, we are actually generating synthetic data points based on the whole dataset. And when we split the data, the test set is already not new, because among most of the train set data and test set data are synthetically generated from the same data so theres no unseen instance left. It is heavily contaminated with the synthetic mathematical twins of the patients in the training set. And the tree based models like XGBoost and LightGBM are powerful enough to memorize the synthetic patterns rather than learning the actual biological indicators of diabetes. Due to this kind of intentional or unintentional mistakes, there can be a dilution of finding a perfect model. But this model cannot be deployed in a real life scenario. Because when this kind of model is deployed to diagnose actual patients in the real world, it will fail catastrophically; it has merely memorized synthetic data artifacts rather than learning the genuine clinical indicators of diabetes. So, to ensure clinical safety and reliability, our findings undoubtedly prove that resampling must strictly take place after the train-test split, preserving the test set as a pure, uncontaminated representation of the real world scenario.

5.2 Metric Selection and Clinical Trust (RQ2)

The first major trap for medical machine learning exists through pre-split data leakage. The second major trap exists through the use of the wrong evaluation metric. Our research shows that a model's value and safety depend on the assessment method that evaluates the model. In machine learning, Accuracy is the universal benchmark. But our post-split

results uncovered exactly how dangerous Accuracy can be when applied to imbalanced healthcare datasets like BRFSS-2015 and Diahealth. Because healthy patients massively outnumber diabetic patients, a model can achieve a false 73% or 74% Accuracy simply by blindly guessing that every single patient is healthy. To an untrained eye, this looks like a functional model. Yet, as our results showed, the same models produced terribly low F1-Scores (ranging from 35% to 39%) and MCC scores (around 0.25 to 0.27). Because of the inconsistency in the model's result, the models were failing at their actual job, which is identifying the minority class of real diabetic patients. The medical science field cannot build clinical trust on a model that, to keep its accuracy high, ignores sick patients. To build clinical trust, researchers need to shift how they evaluate single-label diabetes prediction models. Based on our findings, we strongly say that the Matthews Correlation Coefficient (MCC) and the F1-Score become the primary metrics. Because mathematically, MCC requires the model to succeed across all four quadrants of the confusion matrix, it is impossible to cheat. Furthermore, Recall must be carefully monitored as a secondary metric to ensure patient safety, as missing a diabetic diagnosis (a false negative) carries severe real-world consequences.

5.3 What SHAP Reveals About the Corrected Model (RQ3)

This subsection discusses what the SHAP findings reveal about the corrected post-split models and how those findings help answer RQ-3. The main purpose here is not only to restate which features were important, but to interpret what those feature importance patterns tell us about the models reasoning. A particularly important point from this study is that data leakage not only changes performance scores; it can also change what the model appears to learn. For that reason, the SHAP results are useful not only as an explainability tool but also as a way of checking whether the corrected pipeline is learning from clinically sensible information. The discussion can be organized around several key interpretive themes.

Clinical Alignment: In general, the corrected post-split models align with clinical knowledge, although the strength of that alignment is not equally strong across all datasets. For PIMA indian dataset, the corrected post-split SHAP results aligned most clearly with established diabetes knowledge. Glucose consistently appeared as the dominant feature, while Age, BMI, and sometimes DiabetesPedigreeFunction also remained important. This is encouraging because it suggests that once the major leakage problem is reduced, the model begins to rely on a feature pattern that is clinically expected. In other words, the corrected PIMA indian dataset model not only performs more honestly, but it also reasons in a way that is easier to accept medically. For Diahealth, the answer is more cautious. The corrected post-split models did restore Glucose as the most important feature for the minority class, which is an important positive sign. However, the models still relied heavily on variables such as height, blood pressure, weight, and hypertensive status. Some of these are relevant background risk indicators, but their strong dominance makes the learned pattern look more indirect and less disease-specific. So, for Diahealth, SHAP suggests that the corrected model is more trustworthy than the pre-split one, but still not as clinically clean as desired. For BRFSS-2015, the corrected post-split models did not behave like a laboratory-based clinical model, but that is expected because BRFSS is a survey-based public-health dataset. In this case, the repeated importance of BMI, GenHlth, Age, HighBP, and HighChol is still a meaningful and clinically acceptable pat-

tern, although it should be interpreted more as a population-level diabetes risk structure rather than direct disease physiology. So, BRFSS supports the corrected pipeline differently, not through biomarker-level explanation, but through stable and epidemiologically sensible risk-factor reasoning.

Impact of Leakage on Model Reasoning: The SHAP analysis strongly suggests that leakage changes both performance scores and the model's decision logic.This is one of the most important findings of the study. If leakage only increased performance metrics while leaving the learned feature pattern unchanged, then the problem would be limited to evaluation bias. But that is not what was observed. Across the datasets, the pre-split models often showed different feature hierarchies from the post-split models, which means the contaminated pipeline was not just making the model look better; it was also changing which signals the model relied on most. This was clearest in PIMA indian dataset, where the corrected post-split models followed a more balanced Glucose + Age + BMI pattern, while the pre-split models repeatedly overemphasized Insulin. That is a subtle but important point. It means that the wrong workflow may push the model toward a different explanatory structure, not just a higher score. A similar pattern appeared in Diahealth, where the pre-split models more often elevated height and diastolic blood pressure to unusually high positions, while the post-split models at least restored Glucose as the dominant minority-class feature. In BRFSS-2015, the difference appeared through a stronger shift in the pre-split models toward Income and PhysHlth, while the corrected models stayed more centered on BMI, GenHlth, and Age. So, the SHAP results suggest that pre-split resampling does not simply inflate performance; it can also distort the models internal decision logic.

Dataset-Specific SHAP Patterns: The difference in SHAP clarity across the datasets appears to be closely related to the nature and quality of the datasets themselves. In PIMA indian dataset, the data contain several features that are directly connected to diabetes, especially Glucose, and this makes it easier for the corrected models to recover a clinically interpretable pattern. Even with some remaining preprocessing limitations, the post-split PIMA indian dataset models still showed a strong and believable feature hierarchy. In Diahealth, the dataset seems to contain a weaker or noisier diabetes-specific signal. That is why even the corrected post-split models still relied on indirect features such as height or blood pressure-related variables. In this case, SHAP is revealing something important about the dataset, not only about the pipeline. The corrected workflow improves trustworthiness, but it cannot fully solve the problem of limited clinical specificity in the available features. In BRFSS-2015, the explanation is different again. The dataset is not designed around direct clinical biomarkers. Instead, it reflects broader public-health, lifestyle, and self-reported health information. So, the corrected model is expected to learn a risk profile, not a strict disease-mechanism profile. That is why a SHAP pattern centered on BMI, general health, age, blood pressure, and cholesterol is acceptable here, even if it looks less biologically direct than PIMA indian dataset. This dataset-level difference is important because it shows that the value of SHAP depends not only on the model and the pipeline, but also on the type of data being used.

Interpretation of RQ-3: Taken together, the SHAP findings provide a clear answer to RQ-3. Under the corrected post-split pipeline, the models generally show more trustworthy and more clinically acceptable reasoning than under the pre-split pipeline. The degree of clinical clarity varies by dataset, but the same overall pattern remains: the corrected models produce explanations that are less inflated, less extreme, and more believable.

For PIMA indian dataset, the answer is strongest. The corrected model aligns well with expected diabetes-related reasoning, especially through the dominance of Glucose, supported by Age, BMI, and sometimes DiabetesPedigreeFunction. This can be treated as a form of clinical validation of the models reasoning. For Diahealth, the corrected model improves the explanation, especially for the diabetic class, but the final SHAP pattern still needs to be interpreted carefully because the dataset seems to encourage reliance on weaker or indirect predictors. So, the corrected model is better, but not fully satisfying from a clinical perspective. For BRFSS-2015, the corrected model shows a stable and meaningful risk-factor pattern, but one that should be interpreted at the population health level rather than as direct clinical causation. In this sense, the corrected BRFSS model is valid, but its explanation belongs more to epidemiological reasoning than to biomarker-driven reasoning.

Final Interpretation: The most important message from this SHAP analysis is that a corrected pipeline improves not only fairness in evaluation, but also trustworthiness in explanation. The pre-split models often looked stronger, but they were learning from contaminated structure and therefore produced feature importance rankings that were too confident and, in some cases, clinically less believable. The post-split models were less impressive in raw numbers, but they revealed a more honest picture of what the models were actually learning. That is why SHAP was especially valuable in this study. It helped show that the effect of leakage is stronger than score inflation alone. It can reshape the models apparent reasoning, change the ranking of important features, and even create a misleading impression of clinical validity. Therefore, the corrected post-split pipeline should be preferred not only because it gives a more reliable evaluation, but also because it gives a more reliable explanation of how the model reaches its decisions.

6 Threats to Validity

Every scientific research has certain constraints influencing the level of confidence with which we interpret and generalize its results, and recognizing them is a crucial part of robust scientific methodology. This chapter identifies and discusses those limitations within the four categories of validity, named as internal validity, external validity, construct validity and conclusion validity acknowledging the possible factors.

6.1 Internal Validity

Internal validity is about how sure we are that a study actually measures what its supposed to, and that the conclusions really match the data.

6.1.1 Randomness in Train-Test Splits

All experiments were carried out using the 80% of dataset for training and 20% for testing with a fixed random seed, ensuring reproducibility and tying the reported performance metrics to a specific data partition. However, evaluation scores may vary if a different pre-specified random seed value is assigned resulting in a slightly different split. The evaluation metrics of Recall and MCC show sensitivity to test set composition when applied to small datasets which include the Pima Indian Diabetes Dataset that contains 768 samples. The results for PIDD should be interpreted with caution because performance might vary under different random split tests. The larger datasets of BRFSS-2015 which has 249049 instances and DiaHealth which has 5437 instances produce results that show greater stability and consistency.

6.1.2 Pre-processing Choices and Their Influence on Results

Several preprocessing strategies like Imputation (MICE), IQR and a combination of Pearson correlation and Mutual Information were applied during the experiment to handle the missing values in PIDD, to remove the outliers and to select features respectively. Although the selection of preprocessing techniques may vary depending upon the researcher, these are the standard approaches consistently used for a cleaner dataset. Any alternative combination of these preprocessing choices, for cleaning the dataset, such as use of median imputation instead of MICE or selection of a different outlier detection threshold would result in different outputs. To illustrate this sensitivity of results to these choices, parallel experiments including and excluding outliers and feature selection were implemented.

6.1.3 Hyperparameter Configuration

A predefined set of hyperparameters was used to train the XGBoost, LightGBM, and Gaussian Naive Bayes models as a standard practice, while fairness was ensured by main-

taining consistency across all the setups. However, each pipeline configuration was not subjected to extensive hyperparameter tuning like grid search or Bayesian optimization. Though optimizing hyperparameters would improve the post-split performance, reducing the observed inflation gap between pre-split and post-split results, our primary objective was to exhibit the presence and extent of the inflation rather than to identify the optimal model among them.

6.2 External Validity

External validity indicates how well the results of a study can be generalized beyond the specific dataset, algorithms, and experimental conditions, determining whether these are applicable to a broader population or practical scenarios.

6.2.1 Dataset-Specific Demographics and Population Scope

Each of the three datasets used represents three distinct populations, as PIDD includes adult females only from Pima Indian heritage near Phoenix, Arizona, BRFSS-2015 provides a broader coverage across all 50 U.S states, and DiaHealth is derived from Bangladeshi clinical records reflecting the South Asian demographics. Consequently, despite being developed as a benchmark dataset from a high-risk Type 2 diabetes region, PIDD lacks generalizability to male patients, younger individuals, and other ethnic groups, whereas the findings of BRFSS can be generalized within the American adult population, although certain limitations remain, such as reliance on self-reported telephone responses and the underrepresentation of marginalized or hard-to-reach communities. Moreover, DiaHealth represents a valuable yet underutilized dataset in global diabetes prediction research as its findings may not be directly transferable to populations with differing dietary habits, genetic profiles, or healthcare accessibility. To conclude, while the use of these three datasets provides a broader analytical scope, it still may not be uniformly generalizable across all global populations.

6.2.2 Generalizability to Other Algorithms

The three specifically focused models of this study include two gradient boosting methods, named XGBoost and LightGBM, representing state-of-the-art ensemble techniques, along with Naive Bayes as a probabilistic baseline. The pre-split versus post-split data leakage findings are mostly applicable to a variety of supervised learning techniques since the major problem comes from data contamination rather than the behavior of any particular model. The observations of this study are mainly applicable to tree-based ensemble models in imbalanced medical classification, as both the inflation gap and SHAP feature importance variations between the pipelines may change when other algorithms like deep neural networks, support vector machines, or logistic regression are employed.

6.2.3 Generalizability to Other Medical Prediction Tasks

The methodological issues, such as pre-split resampling, misleading accuracy metrics, and the importance of SHAP-based explainability, identified in this research are not limited to diabetes prediction only, but rather they apply to a wide range of any binary medical classification involving class imbalance, including cancer detection, heart disease prediction, and early screening of neurological disorders. Since the study investigated these issues exclusively based on diabetes datasets, caution should be exercised to generalize

the findings to other medical domains, while further validation in different contexts is yet required.

6.3 Construct Validity

Construct validity evaluates whether the studys tools, metrics, and experimental configurations accurately measure the intended theoretical concepts to bridge the gap between operationalized measures and underlying constructs, ensuring that the results reflect reality rather than research bias.

6.3.1 Measuring Data Leakage Through Metric Inflation

This study looks at data leakage by comparing the difference in evaluation results between pipelines where preprocessing is done before splitting the data and those where it is done after. This method works well in practice by clearly showing how performance can get inflated alongside the large and consistent gaps found across all three datasets, which also strongly support this concept. However, it should be noted that highly inflated evaluation scores are not a direct mathematical proof of data leakage; they are more of a visible outcome of it. There is also a possibility that other factors, such as overfitting that is not related to leakage, might explain some portion of the difference. Even so, the fact that the same pattern appears across three very different datasets, four resampling methods, and three types of models makes other explanations less convincing. Nevertheless, future studies could apply more formal statistical methods for detecting leakage to strengthen these results further.

6.3.2 Evaluating Clinical Utility Through Standard Metrics

The observed results suggest that measures like MCC, F1-score, and recall are more useful than simple accuracy when working with imbalanced diabetes datasets. This idea is supported by a good amount of research in medical machine learning till now. Nonetheless, there is not a single exact and common definition of "clinical utility" in this field currently. The argument that these metrics provide better assessment of clinical usefulness through their measurement system operates as a valid point but has not yet received proof through real healthcare applications or professional evaluations from medical practitioners. The validation study needs to investigate actual patient results while medical experts provide their expertise to establish clinical usefulness. The research establishes statistical distinctions between model performances which the study interprets for their potential clinical application yet it stops short of assessing their performance in actual healthcare environments.

6.3.3 SHAP as a Measure of Feature Importance

SHAP, one of the most theoretically rigorous and widely accepted explainability frameworks, is used as the key tool for ranking feature importance and explaining model decisions. It is essential to understand what SHAP measures and what it does not, because SHAP values indicate each feature's contribution to a particular model's prediction rather than the actual causal association between clinical variables and diabetes in practical cases. For instance, SHAP analysis identifying glucose as the most important feature indicates its strong influence in the model's decision-making process rather than establishing it as the significant biological reason for diabetes across all populations. Furthermore, feature

importance scores from TreeSHAP and KernelSHAP are used for different model architectures and are not directly comparable across models, though relative rankings within each model are still valid. These interpretations align with standard practice in the SHAP literature and do not undermine our findings, but it is crucial to note the difference between model-based importance and biological causality.

6.4 Conclusion Validity

Conclusion validity, often referred to as statistical conclusion validity, denotes the extent to which the study's statistical and analytical procedures support the drawn conclusions. It ensures that whether a relationship (covariation) exists or not is correctly identified by the researchers, reducing the likelihood of Type I (false positive) and Type II (false negative) errors.

6.4.1 Absence of Formal Statistical Significance Testing

The comparative analysis between pre-split and post-split pipelines, as well as among different preprocessing configurations, is based on differences in point estimate-based performance metrics instead of formal statistical tests like t-tests, Wilcoxon signed-rank tests, or McNemars test. The observed inflation discrepancies, frequently ranging from 15% to 20% or more across important metrics, are too substantial to be attributed to chance. Yet, minor differences between certain pipeline structures could not be statistically significant in the absence of formal significance testing. To thoroughly evaluate the significance of these performance discrepancies, future studies should incorporate appropriate statistical tests combined with repeated cross-validation.

6.4.2 Limited Cross-Validation in the Main Pipeline

In this study, the experiments are based on a single 80/20 train-test split instead of using repeated k-fold cross-validation. This decision was intentional, mainly to keep the comparison between pre-split and post-split pipelines straightforward and easier to understand and reproduce. Relying entirely on one split has disadvantages because the outcomes can sometimes be slightly overly positive or negative in comparison to the model's actual average performance. This is especially a concern for smaller datasets like PIDD. Even then, the main conclusion of the study, that pre-split resampling leads to inflated evaluation metrics, remains reliable, as it appears consistently across 32 different experimental setups involving three separate datasets. Still, using something like nested cross-validation for each pipeline would likely give more stable and dependable performance approximations, and would be a good improvement for future research building on this work.

6.4.3 Scope of the Resampling Techniques Evaluated

Four resampling techniques, SMOTE, SMOTEENN, SMOTETomek, and ADASYN, which are widely used and well-established oversampling and hybrid techniques in the imbalanced learning literature, are thoroughly evaluated in this research. However, other strategies such as Borderline-SMOTE, SVM-SMOTE, or purely cost-sensitive learning approaches were not implemented in this study. Some of these alternative techniques may exhibit different behavior under pre-split conditions, resulting in increased or decreased inflation. The conclusions about the risks of pre-split resampling are drawn from

the evaluation of the four broadly representative techniques employed in this research. Although these four approaches are representative, they do not constitute a comprehensive analysis of all potential resampling methods.

6.5 Summary

This chapter defines the boundaries within which the findings should be interpreted without calling the results into question. The core observations, supported by extensive experimental evidence across multiple datasets, models, and configurations, include pre-split resampling as the key cause of severe metric inflation consistently, consideration of accuracy as a primary metric is unreliable for imbalanced clinical datasets, and the notable differences in feature importance across biased and unbiased pipelines demonstrated by SHAP analysis. The demographic constraints of the dataset, the absence of formal statistical testing, and reliance on a one-train-test split suggest directions for further investigations that do not undermine the study's main argument. The limitations need to be addressed by the researchers through three best practices: inclusion of data diversity, use of nested cross-validation and diligent hypothesis testing.

7 Limitations

The study successfully identifies major methodological problems that prevent accurate diabetes prediction, while its researchers need to recognize their study's actual reach and existing limitations. The study requires researchers to conduct honest self-evaluation as they need to understand their research boundaries, which create restrictions on how to apply their study results.

7.1 Dataset Constraints and Generalizability

The datasets used in this study are standard in machine learning, but they carry inherent limitations. The Pima Indian Diabetes Dataset (PIDD) presents an extremely specific demographic study that only focuses on women of Pima Indian heritage whose data collection began in 1988. The dataset fails to represent modern global populations because it only includes biological and lifestyle attributes from specific time periods. Similarly, the BRFSS-2015 dataset depends entirely on self-reported phone surveys. Unlike clinical lab results, self-reported data are prone to recall bias and subjective interpretation. Whereas the inclusion of the DiaHealth dataset helps validate our methodology on a modern South Asian cohort, the overall findings of this study may still not perfectly generalize to all global populations or clinical environments without further targeted training.

7.2 Explainability Limitations (SHAP)

The ensemble models require transparent evaluation, which we achieve through our implementation of the SHAP (Shapley Additive exPlanations) framework. The mathematical capabilities of SHAP become restricted when it needs to assess features that show strong correlation with each other. SHAP demonstrates a tendency to divide the "importance" score between two clinical variables that show a strong association, for example, BMI and Skin Thickness. This presents several risk factors to clinicians, which delay their ability to identify the single most critical risk factor

7.3 Lack of Prospective Clinical Validation

Finally, this study is basically a retrospective computational analysis. The algorithms and pipelines were evaluated purely on historical data. We have not conducted prospective clinical validation, meaning these corrected models have not been deployed in an active hospital setting to predict diabetes in real, incoming patients over time. Until such real-world, clinical-trial-style validation occurs, the models remain theoretical diagnostic tools rather than field-ready medical software.

8 Ethical Considerations

While using machine learning into the healthcare sector, we have to consider that these researches will directly impact human lives. Model Building is not only a simple computational exercise, this holds ethical responsibilities which must be managed carefully and it is more important while working on healthcare.

8.1 Scientific Integrity in Performance Reporting

One of the main ethical concerns highlighted by our research is the publication and deployment of overconfident models. Our Pre-split resampling experiments prove that how data leakage can achieve a high performance in evaluation metrics without actually learning anything. When researchers publish these artificially inflated benchmarks, it creates a dangerous standard. Other data scientists and healthcare institutions may rely on these published papers to build their own tools. We have a strict ethical responsibility to ensure that the performance metrics we report are mathematically honest and clinically viable, rather than just mathematically impressive.

8.2 Clinical Consequences of Misclassification

Clinically, not all evaluation metrics are equally strong to prove the real-world clinical reliability and safety of a model. Different metrics capture different aspects of performance and may not reflect the true cost of misclassification in medical decision-making. Our findings highlight the importance of selecting appropriate evaluation metrics, such as MCC and F1-score, rather than relying solely on standard accuracy. We have already seen in the results section that if a dataset is highly imbalanced, then it can achieve high accuracy by ignoring the minority class.

In a real-world scenario, the impact of these errors is not equal. A false positive may only cause a healthy patient some temporary anxiety and possibly the cost of an additional blood test. On the other hand, a false negative is far more serious and concerning because it means a diabetic patient may be sent home with a false sense of security, delaying necessary treatment and potentially leading to severe and irreversible complications. A model which consistently misses true positive cases is not just inaccurate, it represents a critical failure in clinical responsibility and medical ethics.

8.3 Algorithmic Fairness and Demographic Bias

Finally, there is an ethical obligation, in order to make sure the fairness across all patient groups. If a model is trained on a small, narrow demographic region (such as PIMA indian) then the model learns the physiological patterns of that specific group. Such a model should not be deployed for a completely different demographic group without

doing a proper calibration. Because the model may misclassify for that group. Healthcare algorithms really need to be put to the test to make sure they aren't carrying over any old biases. Because, at the end of the day, every single patient deserves fair and accurate care, no matter who they are or where they come from.

9 Conclusion and Future Work

This research examined an important methodological issue in diabetes prediction research, which is how much pre-split resampling inflates model performance, which evaluation measures give a more useful view of model quality in an imbalanced medical datasets, and what SHAP reveals about model reasoning under correct and biased pipelines. To answer these questions, experiments were conducted on three publicly available datasets: PIMA indian dataset, BRFSS-2015, and Diahealth. The study compared pre-split resampling and post-split resampling under multiple preprocessing settings and used SHAP-based explainability to examine the features behind minority-class detection.

9.1 Summary of Findings

The findings show clearly that pre-split resampling leads to inflated performance results. Across the three datasets, the pre-split pipeline repeatedly produced higher scores than the post-split pipeline. Although those results looked strong at first, they did not represent a fully trustworthy evaluation, because resampling before the train-test split weakens the independence of the test set and introduces data leakage. This confirms that a model can appear much better than it really is when the pipeline is not designed correctly.

The study also shows that accuracy alone is not enough to judge model usefulness in imbalanced diabetes datasets. In medical classification, a model may achieve a high accuracy score while still failing to detect enough positive diabetic cases. For this reason, measures such as Recall, F1-score, and especially MCC provide a more balanced and realistic assessment. Among these, MCC was particularly useful because it considers all parts of the confusion matrix and gives a clearer picture of model quality when the classes are imbalanced.

Finally, the SHAP analysis showed that leakage does not only affect the final score; it can also affect what the model appears to learn. In PIMA indian dataset, the corrected post-split pipeline produced a more clinically meaningful explanation pattern, where Glucose remained dominant. In contrast, the pre-split pipeline repeatedly overemphasized Insulin, making the explanation less stable and less believable. For Diahealth, the minority-class SHAP analysis restored Glucose as the most important feature under the post-split pipeline, though models still relied on indirect variables (like height and weight), suggesting the dataset itself contains a noisier signal. For the survey-based BRFSS-2015, the post-split models produced a coherent cardiometabolic risk profile (BMI, GenHlth, Age, HighBP), while the pre-split models shifted toward indirect variables such as Income and PhysHlth.

9.2 Research Contributions

Taken together, the results of this thesis support three final conclusions and contributions to the field of medical machine learning:

- **A Corrected Pipeline:** This study proves that pre-split resampling creates misleadingly high performance and should not be treated as a valid evaluation strategy. Therefore, a main contribution of this study is establishing that resampling should be applied only after the train-test split, so that the final test set remains untouched and the evaluation remains fair.
- **A Metric Recommendation:** This research demonstrates that imbalanced medical datasets require more informative evaluation measures than accuracy alone. The future diabetes prediction studies should move away from depending mainly on accuracy and should instead report a broader set of evaluation measures, with MCC and F1-score being especially useful for real-world usefulness.
- **SHAP-Based Explainability for the Minority Class:** This study contributes the insight that SHAP can reveal whether a model is learning from clinically sensible patterns, and shows that those patterns change when the pipeline is biased. In that sense, explainability was not only used as a visualization tool in this research; it also served as a reliable method for checking the trustworthiness of model reasoning.

9.3 Future Work

In conclusion, this study shows that the best-performing model on paper is not always the most reliable model in practice. The post-split pipelines usually produced lower scores than the pre-split pipelines, but those lower scores were more honest, methodologically correct, and more meaningful for real-world interpretation. For diabetes prediction research, this is a very important lesson. A right pipeline should not only aim for high performance, but also for clean evaluation, better clinical alignment, and good explanation.

Building on these findings, this research recommends that future work in diabetes prediction should strictly follow post-split resampling, use multiple evaluation metrics, and include explainability analysis as a regular part of model validation. Future studies can expand on this framework by applying these corrected pipelines to more diverse or hospital-specific datasets, and by exploring advanced modeling techniques to see if they can better isolate true clinical signals from noisy data.

Bibliography

[1] M. Kowsher, M. Y. Turaba, T. Sajed, and M. M. Rahman, "Prognosis and treatment prediction of type-2 diabetes using deep neural network and machine learning classifiers," in *2019 22nd International Conference on Computer and Information Technology (ICCIT)*, pp. 1–6, IEEE, 2019.

[2] W. N. Hauwanga, T. Y. Abdalhamed, L. A. Ezike, I. S. Chukwulebe, A. K. Oo, A. Wilfred, A. R. A. K. A. Khan, J. Chukwuwike, E. Florial, H. Lawan, *et al.*, "The pathophysiology and vascular complications of diabetes in chronic kidney disease: A comprehensive review," *Cureus*, vol. 16, no. 12, 2024.

[3] Z. Ullah, F. Saleem, M. Jamjoom, B. Fakieh, F. Kateb, A. M. Ali, and B. Shah, "Detecting high-risk factors and early diagnosis of diabetes using machine learning methods," *Computational Intelligence and Neuroscience*, vol. 2022, no. 1, p. 2557795, 2022.

[4] J. Wu, Z. Feng, J. Duan, Y. Li, P. Deng, J. Wang, Y. Yang, C. Meng, W. Wang, A. Wang, *et al.*, "Global burden of type 2 diabetes attributable to non-high body mass index from 1990 to 2019," *BMC Public Health*, vol. 23, no. 1, p. 1338, 2023.

[5] E. W. Suradji, S. S. Dhaliwal, Z. Li-Feng, E. Zhou, A. Lim, H. H. Wong, Y. Du, F. Fredicia, A. Dianky, P. Pasaribu, *et al.*, "Assessment of blood glucose measurement using new noninvasive technology: Protocol and methodology," *JMIR Research Protocols*, vol. 15, no. 1, p. e76558, 2026.

[6] Y. Barzegar, A. Barzegar, F. Bellini, F. DAscenzo, I. Gorelova, and P. Pisani, "Machine learning pipeline for early diabetes detection: A comparative study with explainable ai," *Future Internet*, vol. 17, no. 11, p. 513, 2025.

[7] S. Masood, M. A. Al-Bashrawi, M. A. Khan, and Y. K. Dwivedi, "From data to diagnosis: a systematic review on ai-driven approaches to diabetes prediction," *Artificial Intelligence Review*, 2026.

[8] M. Kiran, Y. Xie, N. Anjum, G. Ball, B. Pierscionek, and D. Russell, "Machine learning and artificial intelligence in type 2 diabetes prediction: a comprehensive 33-year bibliometric and literature analysis," *Frontiers in digital health*, vol. 7, p. 1557467, 2025.

[9] M. Matboli, A. Khaled, M. F. Ahmed, M. Y. Ahmed, R. Khaled, G. M. Elmakromy, A. M. A. Ghani, M. M. El-Shafei, M. R. M. Abdelhalim, and A. M. A. E. Gwad, "Machine learning-based stratification of prediabetes and type 2 diabetes progression," *Diabetology & Metabolic Syndrome*, vol. 17, no. 1, p. 227, 2025.

[10] M. Lugner, A. Rawshani, E. Helleryd, and B. Eliasson, "Identifying top ten predictors of type 2 diabetes through machine learning analysis of uk biobank data," *Scientific reports*, vol. 14, no. 1, p. 2102, 2024.

[11] N. Nghiem, N. Wilson, J. Krebs, and T. Tran, "Predicting the risk of diabetes complications using machine learning and social administrative data in a country with ethnic inequities in health: Aotearoa new zealand," *BMC medical informatics and decision making*, vol. 24, no. 1, p. 274, 2024.

[12] A. Khriss, M. Badaoui, W. Boutayeb, H. Al Ali, and A. Kerkour Elmiad, "Predictive modeling of diabetes risk factors and early complications using machine learning and deep learning approaches," in *Computational Mathematics and Modelling for Diabetes*, pp. 297–323, Springer, 2025.

[13] M. Hasan and F. Yasmin, "Predicting diabetes using machine learning: A comparative study of classifiers," *arXiv preprint arXiv:2505.07036*, 2025.

[14] K. Elmenshawy, N. Wael, R. Ahmed, and A. El-Douh, "Diabetes prediction using machine learning and explainable artificial intelligence techniques," *SciNexuses*, vol. 1, pp. 28–43, 2024.

[15] F. Rahman, S. Hossain, J.-J. Tiang, and A.-A. Nahid, "Diabetes prediction using feature selection algorithms and boosting-based machine learning classifiers," *Diagnostics*, vol. 15, no. 20, p. 2622, 2025.

[16] M. A. Talukder, M. M. Islam, M. A. Uddin, M. Kazi, M. Khalid, A. Akhter, and M. Ali Moni, "Toward reliable diabetes prediction: Innovations in data engineering and machine learning applications," *Digital Health*, vol. 10, p. 20552076241271867, 2024.

[17] S. K. Jena, D. K. Behera, A. K. Jena, and J. K. Rout, "A novel ensemble machine learning framework for improved diabetes prediction and complication prevention," *Procedia Computer Science*, vol. 258, pp. 4008–4017, 2025.

[18] P. Sampath, G. Elangovan, K. Ravichandran, V. Shanmuganathan, S. Pasupathi, T. Chakrabarti, P. Chakrabarti, and M. Margala, "Robust diabetic prediction using ensemble machine learning models with synthetic minority over-sampling technique," *Scientific reports*, vol. 14, no. 1, p. 28984, 2024.

[19] S. M. Ganie, P. K. D. Pramanik, M. Bashir Malik, S. Mallik, and H. Qin, "An ensemble learning approach for diabetes prediction using boosting techniques," *Frontiers in Genetics*, vol. 14, p. 1252159, 2023.

[20] S. Nakka, "Diabetes prediction using machine learning," 2025.

[21] A. Dutta, M. K. Hasan, M. Ahmad, M. A. Awal, M. A. Islam, M. Masud, and H. Meshref, "Early prediction of diabetes using an ensemble of machine learning models," *International Journal of Environmental Research and Public Health*, vol. 19, no. 19, p. 12378, 2022.

[22] R. P. Bhatta, "Ensemble based machine learning model for prediction of diabetes," *NPRC Journal of Multidisciplinary Research*, vol. 3, no. 2, pp. 116–129, 2026.

[23] S. M. Ganie, P. K. D. Pramanik, M. Bashir Malik, S. Mallik, and H. Qin, "An ensemble learning approach for diabetes prediction using boosting techniques," *Frontiers in Genetics*, vol. 14, p. 1252159, 2023.

[24] S. Firdous, G. A. Wagai, and K. Sharma, "A survey on diabetes risk prediction using machine learning approaches," *Journal of family medicine and primary care*, vol. 11, no. 11, pp. 6929–6934, 2022.

[25] M. N. Islam, M. M. H. Rimon, S. S.-E. Shamim, Z. M. Fahad, M. J. I. Mony, M. J. U. Chowdhury, *et al.*, "An improved ensemble-based machine learning model with feature optimization for early diabetes prediction," *arXiv preprint arXiv:2512.02023*, 2025.

[26] A. Ahmed, J. Khan, M. Arsalan, K. Ahmed, A. A. Shahat, A. Alhalmi, and S. Naaz, "Machine learning algorithm-based prediction of diabetes among female population using pima dataset," in *Healthcare*, vol. 13, p. 37, MDPI, 2024.

[27] P. Gupta, "Diabetes prediction using machine learning," *Journal of Electrical Systems*, vol. 20, pp. 2244–2257, 05 2024.

[28] A. El-Sayyid El-Bashbishy and H. M. El-Bakry, "Pediatric diabetes prediction using machine learning," *Scientific Reports*, vol. 16, no. 1, p. 1979, 2026.

[29] A. Parvez and M. J. Mufti, "Generalizable diabetes risk stratification via hybrid machine learning models," *arXiv preprint arXiv:2509.20565*, 2025.

[30] M. M. Chowdhury, R. S. Ayon, and M. S. Hossain, "An investigation of machine learning algorithms and data augmentation techniques for diabetes diagnosis using class imbalanced brfss dataset," *Healthcare Analytics*, vol. 5, p. 100297, 2024.

[31] Y. Jang, "Feature-based ensemble modeling for addressing diabetes data imbalance using the smote, rus, and random forest methods: a prediction study," *Ewha Medical Journal*, vol. 48, no. 2, 2025.

[32] I. Abousaber, H. F. Abdallah, and H. El-Ghaish, "Robust predictive framework for diabetes classification using optimized machine learning on imbalanced datasets," *Frontiers in Artificial Intelligence*, vol. 7, p. 1499530, 2025.

[33] H. Torkey, E. Ibrahim, E. E.-D. Hemdan, A. El-Sayed, and M. A. Shouman, "Diabetes classification application with efficient missing and outliers data handling algorithms," *Complex & Intelligent Systems*, vol. 8, no. 1, pp. 237–253, 2022.

[34] L. O. Joel, W. Doorsamy, and B. S. Paul, "On the performance of imputation techniques for missing values on healthcare datasets," *arXiv preprint arXiv:2403.14687*, 2024.

[35] A. E. Setiawan, S. Rustad, A. Syukur, A. S. Moch, M. Akrom, and A. W. Setiawan, "Smote-enn resampling to optimize diabetes prediction in imbalanced data," *Ingenierie des Systemes d'Information*, vol. 30, no. 5, p. 1163, 2025.

[36] H. putri Barus, F. A. Tarigan, *et al.*, "Comparison of xgboost and naive bayes models in type 2 diabetes prediction with rfe feature selection," *Sinkron: jurnal dan penelitian teknik informatika*, vol. 10, no. 1, pp. 352–360, 2026.

[37] I. Abousaber, H. F. Abdallah, and H. El-Ghaish, "Robust predictive framework for diabetes classification using optimized machine learning on imbalanced datasets," *Frontiers in Artificial Intelligence*, vol. 7, p. 1499530, 2025.

[38] P. Netayawijit, W. Chansanam, and K. Sorn-In, "Interpretable machine learning framework for diabetes prediction: Integrating smote balancing with shap explainability for clinical decision support," in *Healthcare*, vol. 13, p. 2588, MDPI, 2025.

[39] T. Prama, M. Zaman, F. Sarker, and K. Mamun, "Diahealth: A bangladeshi dataset for type 2 diabetes prediction," *Mendeley Data*, vol. 1, p. 2024, 2024.

[40] M. Hassan, "Identifying and addressing research gaps: A comprehensive guide definition of research gap," 01 2025.

[41] H. Shojaee-Mend, F. Velayati, B. Tayefi, and E. Babaee, "Prediction of diabetes using data mining and machine learning algorithms: A cross-sectional study," *Healthcare Informatics Research*, vol. 30, no. 1, pp. 73–82, 2024.

[42] M. Alghamdi, M. Al-Mallah, S. Keteyian, C. Brawner, J. Ehrman, and S. Sakr, "Predicting diabetes mellitus using smote and ensemble machine learning approach: The henry ford exercise testing (fit) project," *PloS one*, vol. 12, no. 7, p. e0179805, 2017.

[43] K. M. Sujon, R. Hassan, K. Choi, and M. A. Samad, "Accuracy, precision, recall, f1-score, or mcc? empirical evidence from advanced statistics, ml, and xai for evaluating business predictive models," *Journal of Big Data*, vol. 12, no. 1, p. 268, 2025.

[44] C. Cao, D. Chicco, and M. M. Hoffman, "The mcc-f1 curve: a performance evaluation technique for binary classification," *arXiv preprint arXiv:2006.11278*, 2020.

[45] C. Cao, D. Chicco, and M. M. Hoffman, "The mcc-f1 curve: a performance evaluation technique for binary classification," *arXiv preprint arXiv:2006.11278*, 2020.

[46] Z. Rafie, M. S. Talab, B. E. Z. Koor, A. Garavand, C. Salehnasab, and M. Ghaderzadeh, "Leveraging xgboost and explainable ai for accurate prediction of type 2 diabetes," *BMC Public Health*, vol. 25, no. 1, p. 3688, 2025.

[47] K. Pang, "A comparative study of explainable machine learning models with shapley values for diabetes prediction," *Healthcare Analytics*, vol. 7, p. 100390, 2025.

[48] U. Galicia-Garcia, A. Benito-Vicente, S. Jebari, A. Larrea-Sebal, H. Siddiqi, K. B. Uribe, H. Ostolaza, and C. Martín, "Pathophysiology of type 2 diabetes mellitus," *International journal of molecular sciences*, vol. 21, no. 17, p. 6275, 2020.

[49] D. Care, "2. diagnosis and classification of diabetes: Standards of care in diabetes2024," *Diabetes Care*, vol. 47, no. Suppl 1, pp. S20–s42, 2024.

[50] J. Ma, Y. Guo, C. Jiang, and X. Fu, "Novel insights into the pathophysiology of diabetes-related complications: implications for improved therapeutic strategies, volume ii," 2025.

[51] S. A. Antar, N. A. Ashour, M. Sharaky, M. Khattab, N. A. Ashour, R. T. Zaid, E. J. Roh, A. Elkamhawy, and A. A. Al-Karmalawy, "Diabetes mellitus: Classification, mediators, and complications; a gate to identify potential targets for the development of new effective treatments," *Biomedicine & Pharmacotherapy*, vol. 168, p. 115734, 2023.

[52] S. Kovačević, J. Brkljačić, D. Vojnović Milutinović, L. Gligorovska, B. Bursać, I. Elaković, and A. Djordjevic, "Fructose induces visceral adipose tissue inflammation and insulin resistance even without development of obesity in adult female but not in male rats," *Frontiers in nutrition*, vol. 8, p. 749328, 2021.

[53] J. Dawi, Y. Misakyan, S. Affa, S. Kades, A. Narasimhan, F. Hajjar, M. Besser, K. Tumanyan, and V. Venketaraman, "Oxidative stress, glutathione insufficiency, and inflammatory pathways in type 2 diabetes mellitus: Implications for therapeutic interventions," *Biomedicines*, vol. 13, no. 1, p. 18, 2024.

[54] A. K. Rai, B. Lee, R. Gomez, D. Rajendran, M. Khan, and V. N. S. Garikipati, "Current status and potential therapeutic strategies for using non-coding rna to treat diabetic cardiomyopathy," *Frontiers in Physiology*, vol. 11, p. 612722, 2021.

[55] S. Hemat Jouy, H. Tonchev, S. M. Mostafa, and A. M. Mahmoud, "Post-covid metabolic fallout: a growing threat of new-onset and exacerbated diabetes," *Biomedicines*, vol. 13, no. 6, p. 1482, 2025.

[56] E. Vounzoulaki, K. Khunti, S. C. Abner, B. K. Tan, M. J. Davies, and C. L. Gillies, "Progression to type 2 diabetes in women with a known history of gestational diabetes: systematic review and meta-analysis," *bmj*, vol. 369, 2020.

[57] D. I. Musa and S. Omachi, "Exercise as medicine in the management of type 2 diabetes mellitus: an overview," *Int J Epidemiol Public Health Res*, vol. 1, no. 5, 2021.

[58] S. Larabi-Marie-Sainte, L. Aburahmah, R. Almohaini, and T. Saba, "Current techniques for diabetes prediction: review and case study," *Applied Sciences*, vol. 9, no. 21, p. 4604, 2019.

[59] E. K. Oikonomou and R. Khera, "Machine learning in precision diabetes care and cardiovascular risk prediction," *Cardiovascular Diabetology*, vol. 22, no. 1, p. 259, 2023.

[60] M. O. Olusanya, R. E. Ogunsakin, M. Ghai, and M. A. Adeleke, "Accuracy of machine learning classification models for the prediction of type 2 diabetes mellitus: A systematic survey and meta-analysis approach," *International journal of environmental research and public health*, vol. 19, no. 21, p. 14280, 2022.

[61] M. O. Edeh, O. I. Khalaf, C. A. Tavera, S. Tayeb, S. Ghouali, G. M. Abdulsahib, N. E. Richard-Nnabu, and A. Louni, "A classification algorithm-based hybrid diabetes prediction model," *Frontiers in Public Health*, vol. 10, p. 829519, 2022.

[62] S. K. S. Modak and V. K. Jha, "Diabetes prediction model using machine learning techniques," *Multimedia Tools and Applications*, vol. 83, no. 13, pp. 38523–38549, 2024.

[63] Z. Ullah, F. Saleem, M. Jamjoom, B. Fakieh, F. Kateb, A. M. Ali, and B. Shah, "Detecting high-risk factors and early diagnosis of diabetes using machine learning methods," *Computational Intelligence and Neuroscience*, vol. 2022, no. 1, p. 2557795, 2022.

[64] F. Rahman, S. Hossain, J.-J. Tiang, and A.-A. Nahid, "Diabetes prediction using feature selection algorithms and boosting-based machine learning classifiers," *Diagnostics*, vol. 15, no. 20, p. 2622, 2025.

[65] H. Naz and S. Ahuja, "Deep learning approach for diabetes prediction using pima indian dataset," *Journal of Diabetes & Metabolic Disorders*, vol. 19, no. 1, pp. 391–403, 2020.

[66] H. El-Sofany, S. A. El-Seoud, O. H. Karam, Y. M. Abd El-Latif, and I. A. Taj-Eddin, "A proposed technique using machine learning for the prediction of diabetes disease through a mobile app," *International Journal of Intelligent Systems*, vol. 2024, no. 1, p. 6688934, 2024.

[67] M. M. Chowdhury, R. S. Ayon, and M. S. Hossain, "An investigation of machine learning algorithms and data augmentation techniques for diabetes diagnosis using class imbalanced brfss dataset," *Healthcare Analytics*, vol. 5, p. 100297, 2024.

[68] A. Ahmed, J. Khan, M. Arsalan, K. Ahmed, A. A. Shahat, A. Alhalmi, and S. Naaz, "Machine learning algorithm-based prediction of diabetes among female population using pima dataset," in *Healthcare*, vol. 13, p. 37, MDPI, 2024.

[69] P. Chowdhury, "A comprehensive machine learning framework for predicting health risk categories using brfss survey data,"

[70] L. Andrade-Arenas and C. Yactayo-Arias, "Comparative evaluation of machine learning models for diabetes prediction: A focus on ensemble methods," *Ingenierie des Systemes d'Information*, vol. 30, no. 7, p. 1795, 2025.

[71] Y. Qin, J. Wu, W. Xiao, K. Wang, A. Huang, B. Liu, J. Yu, C. Li, F. Yu, and Z. Ren, "Machine learning models for data-driven prediction of diabetes by lifestyle type," *International journal of environmental research and public health*, vol. 19, no. 22, p. 15027, 2022.

[72] T. T. Prama, M. Zaman, F. Sarker, and K. A. Mamun, "Diahealth: A bangladeshi dataset for type 2 diabetes prediction." Mendeley Data, V1, 2024. Dataset.

[73] P. Netayawijit, W. Chansanam, and K. Sorn-In, "Interpretable machine learning framework for diabetes prediction: Integrating smote balancing with shap explainability for clinical decision support," in *Healthcare*, vol. 13, p. 2588, MDPI, 2025.

[74] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Smote: synthetic minority over-sampling technique," *Journal of artificial intelligence research*, vol. 16, pp. 321–357, 2002.

[75] G. E. Batista, R. C. Prati, and M. C. Monard, "A study of the behavior of several methods for balancing machine learning training data," *ACM SIGKDD explorations newsletter*, vol. 6, no. 1, pp. 20–29, 2004.

[76] M. S. Santos, J. P. Soares, P. H. Abreu, H. Araujo, and J. Santos, "Cross-validation for imbalanced datasets: avoiding overoptimistic and overfitting approaches [research frontier]," *ieee ComputatioNal iNtelligeNCe magaziNe*, vol. 13, no. 4, pp. 59–76, 2018.

[77] H. He, Y. Bai, E. A. Garcia, and S. Li, "Adasyn: Adaptive synthetic sampling approach for imbalanced learning," in *2008 IEEE international joint conference on neural networks (IEEE world congress on computational intelligence)*, pp. 1322–1328, Ieee, 2008.

[78] S. Kapoor and A. Narayanan, "Leakage and the reproducibility crisis in machine-learning-based science," *Patterns*, vol. 4, no. 9, 2023.

[79] H. He and E. A. Garcia, "Learning from imbalanced data," *IEEE Transactions on knowledge and data engineering*, vol. 21, no. 9, pp. 1263–1284, 2009.

[80] D. Chicco and G. Jurman, "The advantages of the matthews correlation coefficient (mcc) over f1 score and accuracy in binary classification evaluation," *BMC genomics*, vol. 21, no. 1, p. 6, 2020.

[81] T. Fawcett, "An introduction to roc analysis," *Pattern recognition letters*, vol. 27, no. 8, pp. 861–874, 2006.

[82] T. Saito and M. Rehmsmeier, "The precision-recall plot is more informative than the roc plot when evaluating binary classifiers on imbalanced datasets," *PloS one*, vol. 10, no. 3, p. e0118432, 2015.

[83] S. A. Hicks, I. Strümke, V. Thambawita, M. Hammou, M. A. Riegler, P. Halvorsen, and S. Parasa, "On evaluation metrics for medical applications of artificial intelligence," *Scientific reports*, vol. 12, no. 1, p. 5979, 2022.

[84] J. Amann, A. Blasimme, E. Vayena, D. Frey, V. I. Madai, and P. Consortium, "Explainability for artificial intelligence in healthcare: a multidisciplinary perspective," *BMC medical informatics and decision making*, vol. 20, no. 1, p. 310, 2020.

[85] E. Tjoa and C. Guan, "A survey on explainable artificial intelligence (xai): Toward medical xai," *IEEE transactions on neural networks and learning systems*, vol. 32, no. 11, pp. 4793–4813, 2020.

[86] M. T. Ribeiro, S. Singh, and C. Guestrin, "" why should i trust you?" explaining the predictions of any classifier," in *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pp. 1135–1144, 2016.

[87] R. Ganguly and D. Singh, "Explainable artificial intelligence (xai) for the prediction of diabetes management: An ensemble approach," *International Journal of Advanced Computer Science and Applications*, vol. 14, no. 7, 2023.

[88] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," *Advances in neural information processing systems*, vol. 30, 2017.

[89] H. Lee, M.-B. Park, and Y.-J. Won, "Ai machine learning–based diabetes prediction in older adults in south korea: cross-sectional analysis," *JMIR Formative Research*, vol. 9, no. 1, p. e57874, 2025.

[90] J. W. Smith, J. E. Everhart, W. C. Dickson, W. C. Knowler, and R. S. Johannes, "Using the adap learning algorithm to forecast the onset of diabetes mellitus," in *Proceedings of the annual symposium on computer application in medical care*, p. 261, 1988.

[91] O. A. Alade, A. Selamat, and R. Sallehuddin, "The effects of missing data characteristics on the choice of imputation techniques," *Vietnam Journal of Computer Science*, vol. 7, no. 02, pp. 161–177, 2020.

[92] Centers for Disease Control and Prevention, "Behavioral risk factor surveillance system (brfss) survey data." `https://www.cdc.gov/brfss/`, 2015. Accessed: Apr. 12, 2026.

[93] A. Teboul, "Diabetes health indicators dataset." https://www.kaggle.com/datasets/alexteboul/diabetes-health-indicators-dataset, 2021. Accessed: Apr. 12, 2026.

[94] M. J. Azur, E. A. Stuart, C. Frangakis, and P. J. Leaf, "Multiple imputation by chained equations: what is it and how does it work?," *International journal of methods in psychiatric research*, vol. 20, no. 1, pp. 40–49, 2011.

[95] J. W. Tukey *et al.*, *Exploratory data analysis*, vol. 2. Springer, 1977.

[96] R. Blagus and L. Lusa, "Smote for high-dimensional class-imbalanced data," *BMC bioinformatics*, vol. 14, no. 1, p. 106, 2013.

[97] M. V. Anand, B. KiranBala, S. Srividhya, K. C, M. Younus, and M. H. Rahman, "Gaussian naïve bayes algorithm: a reliable technique involved in the assortment of the segregation in cancer," *Mobile information systems*, vol. 2022, no. 1, p. 2436946, 2022.

[98] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pp. 785–794, 2016.

[99] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu, "Lightgbm: A highly efficient gradient boosting decision tree," *Advances in neural information processing systems*, vol. 30, 2017.

[100] Ž. Vujović *et al.*, "Classification model evaluation metrics," *International Journal of Advanced Computer Science and Applications*, vol. 12, no. 6, pp. 599–606, 2021.

[101] D. M. Powers, "Evaluation: from precision, recall and f-measure to roc, informedness, markedness and correlation," *arXiv preprint arXiv:2010.16061*, 2020.

[102] S. M. Lundberg, G. Erion, H. Chen, A. DeGrave, J. M. Prutkin, B. Nair, R. Katz, J. Himmelfarb, N. Bansal, and S.-I. Lee, "From local explanations to global understanding with explainable ai for trees," *Nature machine intelligence*, vol. 2, no. 1, pp. 56–67, 2020.

[103] A. V. Ponce-Bobadilla, V. Schmitt, C. S. Maier, S. Mensing, and S. Stodtmann, "Practical guide to shap analysis: Explaining supervised machine learning model predictions in drug development," *Clinical and translational science*, vol. 17, no. 11, p. e70056, 2024.