

Heart Attack Predictions

By Charalampos Nikandrou

W2018260

Supervisor: Rolf Banziger


Data Science & Analytics – Dissertation 2024

Declaration

I, Charalampos Nikandrou, declare that I am the sole author of this Project, that all references cited have been consulted, that I have conducted all work of which this is a record, and that the finished work lies within the prescribed word limits.

Word Count: 13067

This has not previously been accepted as part of any other degree submission.

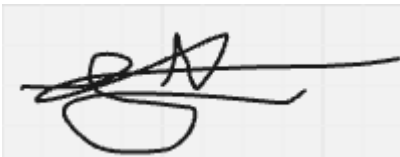
A handwritten signature in black ink, appearing to be 'C. Nikandrou', written on a light gray grid background.

Signed:

Date: 02/09/2024

Form of Consent

I, Charalampos Nikandrou, hereby consent that this Project, submitted in partial fulfilment of the requirements for the award of the MSc degree, if successful, may be made available in paper or electronic format for inter-library loan or photocopying (subject to the law of copyright), and that the title and abstract may be made available to outside organisations.

A handwritten signature in black ink, appearing to be 'C. Nikandrou', written on a light gray grid background.

Signed:

Date: 02/09/2024

Abstract

The primary focus of this dissertation is to predict heart attacks using Machine Learning algorithms giving emphasis to the precision in predictions. We employed a systematic approach, beginning with the application of various machine learning algorithms. Then to address data imbalance issues we utilized different Imbalance methods. The evaluation of our results was based on precision, accuracy, and AUC scores. Following this, we normalized and ranked the outputs to identify the most suitable Imbalance method. Next, this method was combined with various Feature Selection Techniques, and a similar evaluation and ranking process was made to determine the most effective Feature Selection approach. The top four models were then selected and extensively tuned to achieve the highest possible precision.

We used two datasets for the experiments the "Heart Attack Risk Prediction" dataset, which includes 8,763 synthetic patient records and 26 variables, and the "Heart Disease Health Indicators" dataset, which at first contained 253,661 synthetic instances with 22 features, and later was reduced to 50,000 instances for the flexibility.

For the first dataset, we used both Standard and Robust scaling. With Standard scaling, the best combination included the SMOTETomek imbalance method, RFE feature selection, and the GradientBoosting model, which increased the initial precision from 0.3 to 0.391. Furthermore, the 10 features which completed our results are: 'Cholesterol', 'Obesity', 'Alcohol Consumption', 'Diet', 'Stress Level', 'Sedentary Hours/Day', 'Income', 'BMI', 'Physical Activity Days/Week' and 'Sleep Hours/Day'. For Robust scaling, the combination of CC & SMOTE, ANNOVA feature selection, and the Decision Tree model gave us a precision improvement from 0.298 to 0.371. As for the 10 features were: 'Age', 'Sex', 'Cholesterol', 'Heart Rate', 'Diabetes', 'Smoking', 'Stress Level', 'Physical Activity Days/Week', 'Sleep Hours/Day' and 'HBP'. The second dataset processed using only Standard scaling, we identified that SMOTETomek, RFE, and XGB was the optimal combination, which improved the precision from 0.457 to 0.514. Also, the 10 features which were finalized our results are: 'GenHlth', 'HighBP', 'HighChol', 'Smoker', 'Diabetes', 'PhysActivity', 'Fruits', 'Veggies', 'DiffWalk' and 'Sex'. These findings highlight the potential of machine learning models in improving the accuracy of heart attack predictions, with precision being a crucial metric for enhancing clinical decision-making.

Keywords: *Heart Attack Prediction, Machine Learning, Imbalance Data and Feature Selecting*

Acknowledgements

I would like to express my deepest gratitude to my supervisor, Professor Rolf Banziger, for his invaluable help, guidance, and commitment throughout this journey. His insightful advice and unwavering support have been instrumental in the completion of this work.

I also wish to thank my sister, Dina, for her constant encouragement, support, and advice not only during the dissertation but throughout the entire year of my Master's program. Her belief in me has been a source of strength and motivation.

Table of Contents

1. <u>Introduction</u>	7
2. <u>Literature Review</u>	7-17
3. <u>Methodology</u>	17-32
4. <u>Dataset – Analysis</u>	32-41
5. <u>Dataset - Cleaning & Preprocessing</u>	41-42
6. <u>Results</u>	43-56
7. <u>Conclusion</u>	56-58
8. <u>Problems & Difficulties</u>	58
9. <u>Future Work</u>	59
10. <u>References</u>	59-61

1. Introduction

Cardiovascular diseases (CVDs) are the leading cause of death worldwide and they are responsible for around 17.9 million deaths annually. This group of disorders affecting the heart and blood vessels includes also coronary heart disease, cerebrovascular disease, and rheumatic heart disease. It is very concerning that more than four out of five CVD-related deaths are related to heart attacks and strokes, with one-third of these deaths occurring sooner in individuals under the age of 70.

The main behavioral risk factors for heart disease include unhealthy diets, physical inactivity, tobacco use, and harmful alcohol consumption. Environmental factors, such as air pollution, also play a significant role in exacerbating these risks. The demonstration of these risk factors often includes raised blood pressure, elevated blood glucose levels, increased blood lipids, and conditions like overweight and obesity. These intermediate risk factors are measurable and can be useful as critical indicators for people with high risk in heart attacks, heart failure, and other critical complications.

The early identification of those people with CVDs, combined with appropriate treatment, is crucial for preventing premature deaths. Ensuring access to essential medicines and basic health technologies in all primary health care facilities is a necessary step to provide effective treatment and counseling for those in need.

Often the first signs of cardiovascular disease can occur without prior symptoms. Heart attack symptoms typically include chest pain or discomfort, and may radiate to the arms, left shoulder, elbows, jaw, or back. Additional symptoms include difficulty breathing, nausea, light-headedness, cold sweats, and pallor. Women may experience symptoms such as shortness of breath, nausea, vomiting, and back or jaw pain more frequently than men.

Given the critical nature of heart attack prediction, this dissertation explores the use of Machine Learning algorithms to increase the precision of such predictions. By using synthetic datasets, the study aims to identify the most effective combination of imbalance data methods, feature selection techniques, and machine learning models to improve the accuracy and reliability of heart attack predictions. Also, we aimed to find the most important features that affect heart attacks. This research holds the potential to contribute significantly to the early identification and prevention of heart attacks. With that way we can support the public health initiatives which aimed at reducing of the global cardiovascular diseases.

2. Literature Review

Heart disease is one of the leading causes of death worldwide. Early detection and prediction of heart disease can significantly improve patient results through timely interventions. The combination of data mining with machine learning is the key for the solution for those challenges. Machine learning models are commonly applied to detect risks in the early stages of heart disease and heart attack predictions. This literature review explores the application of machine learning techniques for predicting heart attack risk. It focuses on three main pillars, whose interaction is necessary for achieving the highest possible success. These core pillars include feature selection methods, handling imbalanced data and the performance of various machine learning models mainly in heart disease and healthcare settings.

- **Feature Selection Techniques**

Feature selection is critically important in the medical field, especially for predicting heart attacks. One of the main reasons is that it enhances model accuracy by focusing on the most relevant variables, thus improving the predictive power and reducing the risk of overfitting. This leads to models that are not only more accurate but also easier for doctors to explain them, facilitating better decision-making based on key risk factors like age, blood pressure, cholesterol levels, and lifestyle habits. Also, it offers deeper insights into heart disease, allowing targeted solutions and preventive measures. Additionally, managing high-dimensional medical data becomes more feasible with feature selection, as it reduces computational complexity and resource demands. Especially for heart attack prediction, focusing on critical markers not only enhances predictive accuracy, but also minimizes the chances of false positives and negatives, leading to more reliable and actionable risk assessments. Eventually, feature selection in heart attack prediction supports a more efficient, insightful, and patient-centered approach to healthcare. In this study we explore various methods to identify significant features in medical datasets and more specifically in heart attacks predictions. According to our research there are three principal feature selecting methods: Filter, Wrapper and Embedded. [1]

1. Filter Methods

The filter method is used in machine learning to identify and select the most relevant features from a dataset before training a model. It operates independently of the learning algorithm by evaluating the statistical characteristics of each feature depending on the target variable. Commonly, this involves ranking features based on metrics like correlation, chi-square scores, or mutual information. This method is computationally efficient and helps reduce overfitting by eliminating irrelevant or redundant features early in the modeling process. It is particularly useful in scenarios with large datasets where computational resources and processing time are critical concerns.

2. Wrapper Methods

The wrapper method involves evaluating feature subsets using the actual performance of a predictive model. Unlike the filter method, which assesses features independently, the wrapper method considers the interaction between features and the model. This process typically employs a search algorithm, such as forward selection, backward elimination, or recursive feature elimination, to explore different combinations of features. Each subset is tested by training and evaluating the model, selecting the subset that gives the best performance metrics. Although, wrapper method needs more computations than filter, the wrapper method often provides higher predictive accuracy because it optimizes feature selection tailored to the specific model used.

3. Embedded Methods

The embedded method combines the selection process directly into the training of the model. Unlike filter and wrapper methods, the embedded method performs feature selection and model training at the same time. Techniques like Lasso (Least Absolute Shrinkage and Selection Operator) or tree-based methods (e.g., decision trees, random forests) naturally perform feature selection by assigning weights or importance scores to features during training. This approach is computationally efficient because it reduces the number of features while training and smooths the process. This method is particularly effective when dealing with high-dimensional data and complex relationships, offering a robust way to enhance model performance by focusing on the most significant features.

- **Examples in Feature Selecting methods for Heart Attacks Prediction**

In [3], Takci et al. In his research, he combined machine learning with feature selection algorithms to optimize heart attack prediction. Using the Statlog (Heart) dataset, which contains 270 records and 76 features, he implemented stepwise regression, Fisher filtering (FF), and Relief algorithms, ultimately identifying 13 key features. He employed a split method, with 90% of the data for training and 10% for testing, alongside 10-fold cross-validation to ensure robustness. Without feature selection, his models achieved 82.59% accuracy. However, with feature selection, the accuracy improved to 84.81%, with SVM-linear and Naive Bayes emerging as the top-performing models. Additionally, he evaluated the processing time and the ROC value, highlighting the efficiency and enhanced predictive power of combining these techniques.

In [4], Hussein et al. In this research, three different feature selection techniques were evaluated on six machine learning algorithms to enhance heart attack prediction. The study utilized a dataset comprising 303 records and 14 features. The feature selection techniques applied were Information Gain (IG), Gain Ratio (GR), and Symmetric Evaluator (SE), which collectively decreased the features to 9 key attributes. The algorithms tested included six decision tree-based methods: Decision Stump (DS), Hoeffding Tree (HT), J48, Logistic Model Trees (LMT), Random Forest (RF), and Random Tree (RT). Among these, the LMT algorithm consistently demonstrated the best performance both before and after the feature selection process.

In [5] Dash et al., In his research, an extensive survey of feature selection methods was conducted to comprehensively gather and present various techniques. This article serves a complete survey of feature selection methods. The study concluded by identifying four essential steps in a typical feature selection process: generation procedure, evaluation function, stopping criterion, and validation procedure. Generation procedures were categorized into three groups: complete, heuristic, and random, while evaluation functions were classified into five categories: distance, information, dependence, consistency, and classifier error rate measures. Thirty-two existing feature selection methods were systematically categorized based on their combinations of generation procedures and evaluation functions.

In [6] Dissanayake et al., In this article investigates the impact of feature selection methods on heart disease prediction accuracy. Ten techniques including ANOVA, Chi-square, mutual information, ReliefF, forward feature selection, backward feature selection, exhaustive feature selection, recursive feature elimination, Lasso regression, and Ridge regression and evaluated alongside with six classifiers (Decision Tree, Random Forest, SVM, k-NN, Logistic Regression, Gaussian Naive Bayes) on the Cleveland heart disease dataset (303 instances, 75 features). Using a subset of 14 selected features, backward feature selection achieved the highest accuracy of 88.52%, precision of 91.30%, sensitivity of 80.76%, and f-measure of 85.71% with a DT classifier. This marked improvement from 63.92% without feature selection underscores the effectiveness of these methods in disease classification with fewer features.

In [7] Anna et al., In this paper, they proposed the integration of chi-square (CHI) feature selection with Principal Component Analysis (PCA) to enhance machine learning model predictions. From an initial set of 74 features, they identified three feature groups that gave the optimal performance. The study highlights

that CHI-PCA combined with Random Forest (RF) achieved 98.7% accuracy for the Cleveland dataset, 99.0% for the Hungarian dataset, and 99.4% for the CH dataset. They compared these results with models trained on raw data and CHI-selected features, where CHI identified 13 relevant features. The classification models utilized default hyperparameter values and included Decision Tree (DT), Gradient-Boosted Tree (GBT), Logistic Regression (LOG), Multilayer Perceptron (MPC), Naïve Bayes (NB), and Random Forests (RF).

In [8] Adi et al., In this study, incorporating feature selection via backward elimination significantly boosted Naïve Bayes accuracy to 89.45% from an initial 84.29%. The research focuses on various data mining techniques, particularly the Naïve Bayes algorithm, which aids paramedic analysis in heart attack prediction using a dataset from the UC Repository. This dataset comprises 100 patient records with 7 attributes and 1 label. Employing a 10-fold cross-validation split, the study evaluated three feature selection methods: backward elimination, forward selection, and chi-square weight. Among these, Naïve Bayes combined with Forward Selection emerged as the most effective, achieving the highest accuracy of 95.44%. This highlights the superiority of forward selection in enhancing Naïve Bayes model performance for heart attack prediction.

In [9] Jović et al., In this paper, the focus was on feature selection across various application domains, including text mining, image processing/computer vision, bioinformatics, and industrial applications. The study surveyed different methods tailored to these diverse fields, emphasizing the importance of selecting the most effective techniques based on specific evaluation metrics within each subfield. High-dimensional feature spaces in bioinformatics, image processing, industrial applications, and text mining require sophisticated hybrid methodologies due to the complexity and potentially unknown nature of features. While no universal method exists, filter methods leveraging information theory and wrapper methods employing stepwise approaches have shown promising outcomes, underscoring their relevance in optimizing feature selection for diverse problem domains.

Application area	Subfield	Datasets	Feature selection methods	Evaluation metrics	Best performing	Study
Text mining	Text classification	229 text classification problem instances gathered from Reuters, TREC, OHSUMED, etc.	Accuracy, accuracy balanced, bi-normal separation, chi-square, document frequency, F1-measure, information gain, odds ratio, odds ratio numerator, power, probability ratio, random	Accuracy, F-measure, precision, and recall	Information gain (precision), bi-normal separation (accuracy, F-measure, recall)	[38]
	Text clustering	Reuters-21578, 20 Newsgroups, Web Directory	Information gain, chi-square, document frequency, term strength, entropy-based ranking, term contribution, iterative feature selection	Entropy, precision	Iterative feature selection	[39]
Image processing / computer vision	Image classification	Aerial Images, The Digits Data, Cats and Dogs	Relief (R), K-means (K), sequential floating forward selection (F), sequential floating backward selection (B), various combinations R + K + F/B	Average MSE of 100 neural networks	$R+K+B / R+K+F / R+K$, depending on the size of feature subset	[40]
	Breast density classification from mammographic images	Mini-MIAS, KBD-FER	Best-first with forward, backward and bi-directional search, genetic search and random search (k-NN and Naïve Bayesian classifiers)	Accuracy	Best first forward, best first backward	[41]
Bioinformatics	Biomarker discovery	Three benchmark datasets deriving from DNA microarray experiments	Chi-square, information gain, symmetrical uncertainty, gain ratio, OneR, ReliefF, SVM-embedded	Stability, AUC	Chi-square, symmetrical uncertainty, information gain, ReliefF	[42]
	Microarray gene expression data classification	Two gene expression datasets (Freije, Phillips)	Information gain, twoing rule, sum minority, max minority, Gini index, sum of variances, t-statistics, one-dimensional SVM	Accuracy	Consensus of all methods	[43]
Industrial applications	Fault diagnosis	Wind turbine test rig dataset	Distance, entropy, SVM wrapper, neural network wrapper, global geometric similarity scheme	Accuracy	Global geometric similarity scheme with wrapper	[22]

In [10] Ahmad et al., In this study, the goal was to develop a highly accurate ML model for predicting heart disease using the Cleveland heart disease dataset with 303 records and 14 out of 76 features. Given the dataset's high dimensionality, they employed the Jellyfish optimization algorithm to reduce it to a lower-dimensional subspace, enhancing model performance and mitigating overfitting. The Jellyfish algorithm, known for its rapid convergence and flexibility in feature selection, proved highly effective in optimizing features for the SVM classifier. This approach gave exceptional performance metrics: Sensitivity 98.56%, Specificity 98.37%, Accuracy 98.47%, and Area Under Curve (AUC) 94.48%. Comparing various ML models such as ANN, Decision Tree (DT), Adaboost, and SVM using the Jellyfish algorithm underscored SVM as the most accurate, achieving an accuracy of 98.09%.

• **Handling Imbalanced Data**

In real-world in the medical domain imbalanced data are very typical because most people are generally healthy than those who are ill. This is reflective of the broader human condition, where maintaining health is more common and vital for the overall survival of the population. In this project and research, we focused on Binary Classification problem where all the possible outcomes be included in one of the two classes. An imbalanced classification problem occurs when the examples across classes are not evenly distributed. This can range from a modest bias to an extreme imbalance, where the minority class is represented by only one instance against hundreds, thousands, or even millions in the majority class. That distribution has negatively impact the performance of predictive models, mostly for the minority class and this is a problem as we are more interested in the facts and the reasons which cause this class. To surpass this problem there are several types of imbalanced classification techniques: Collect more Data,

Data Sampling Algorithms, Cost-Sensitive Algorithms, One-Class Algorithms and Probability Tuning Algorithms. [2]

1. Collect more Data

One approach to handle the skewed distribution of our classes is by collecting more data, but this method is challenging due to its high cost and time-intensive nature. For these reasons it is mainly not recommended and preferred by the scientists in the medical field.

2. Data Sampling Algorithms

Another commonly used method involves data sampling algorithms that modify the composition of the training dataset to enhance the performance of standard machine learning algorithms. This process includes three techniques as Oversampling, Undersampling and Combined Oversampling and Undersampling.

a) Oversampling

Data oversampling requires duplicating examples of the minority class or generating new examples from existing minority class instances. This method includes several techniques such as Random Oversampling, SMOTE (Synthetic Minority Oversampling Technique), Borderline SMOTE, SVM SMOTE, k-Means SMOTE and ADASYN with SMOTE to be the most common and popular for heart attack predictions.

b) Undersampling

Undersampling entails removing examples from the majority class, such as randomly or using an algorithm to choose which examples to remove. There are a few Undersampling techniques such as Random Undersampling, Condensed Nearest Neighbor, Tomek Links, Edited Nearest Neighbors, Neighborhood Cleaning Rule and One-Sided Selection with Tomek Links to be more famous.

c) Combined Oversampling

Most Oversampling methods can be integrated with various Undersampling techniques. Most common combinations of over and undersampling include SMOTE with Random Undersampling or Tomek Links or Edited Nearest Neighbours

3. Cost-Sensitive Algorithms

Cost-sensitive algorithms are adjusted versions of machine learning models that consider the misclassification weights when fitting the model on the training dataset. These algorithms can be helpful when used on imbalanced classification. Some typical examples of Cost-sensitive algorithms are Logistic Regression (LR), Decision Trees (DT), Support Vector Machines (SVM), Artificial Neural Networks (ANN), Bagged Decision Trees, Random Forest (RF) and Stochastic Gradient Boosting

4. One-Class Algorithms

Algorithms originally designed for outlier and anomaly detection can be applied for classification tasks. They are particularly beneficial in cases of extreme class imbalance, where the positive class is significantly small. Common examples of One-Class Algorithms are One-Class SVM, Isolation Forests, Minimum Covariance Determinant, and Local Outlier Factor

5. Probability Tuning Algorithms

There are two ways to enhance predicted probabilities: Calibrating Probabilities and Tuning the Classification Threshold

a) Calibrating Probabilities

Some algorithms are trained using a probabilistic framework and as a result provide calibrated probabilities. This implies that if an algorithm predicts 100 examples to have the positive class label with an 80 percent probability, it will correctly predict the positive class label approximately 80 percent of the time. Calibrated probabilities are essential for a model to be considered effective in binary classification tasks where probabilities are needed as outputs or used to evaluate the model, such as in ROC AUC or PR AUC. Some examples of machine learning algorithms that predict calibrated probabilities are Logistic Regression (LR), Linear Discriminant Analysis, Naive Bayes (NB) and Artificial Neural Networks (ANN). Also, some examples of probability calibration algorithms include Platt Scaling and Isotonic Regression.

b) Tuning the Classification Threshold

Some algorithms initially predict probabilities, which must later be converted into definitive class labels. This conversion is necessary when the output needs to be in the form of class labels or when the model's performance is evaluated based on class labels. Examples of probabilistic machine learning algorithms that predict a probability by default include Logistic Regression (LR), Linear Discriminant Analysis, Naive Bayes (NB) and Artificial Neural Networks (ANN).

• Examples in Handling Imbalanced Data methods for Heart Attacks Prediction

In [11] Ramyachitra et al., In this paper investigates the challenges presented by imbalanced datasets and explores various solutions. They examine the defining characteristics of imbalanced datasets, review key techniques and algorithms, and provide a classification framework for managing them. Their survey highlights several approaches to address these issues, emphasizing the effectiveness of data-level processes. Specifically, they found that oversampling algorithms for data preprocessing, combined with balancing algorithms.

In [12] Kotsiantis et al., This paper explores various strategies for addressing the challenges posed by imbalanced datasets. While cost-sensitive learning is often seen as better for random resampling, innovative re-sampling and combination techniques can surpass it by introducing new information or removing redundancy in the data. These methods can significantly enhance the learning algorithm's performance. The relationship between training set size and classification performance is particularly crucial: small imbalanced datasets tend to inadequately represent the minority class, especially when class overlap and subclusters are present. Conversely, larger datasets better capture the minority class's characteristics, mitigating these issues and improving classification accuracy.

In [13] Wang et al., In this study, they developed the UCO (undersampling - clustering-oversampling) algorithm, combining random undersampling, clustering, and oversampling techniques, to address imbalanced data in heart attack prediction among stroke patients. The UCO(120) variant, with an undersampling number of 120, generated a nearly balanced dataset, significantly enhancing feature extraction for machine learning models. Applying this to the MIMIC-III database, which included 2,488 stroke samples (82 with heart attacks), they found that the Random Forest classifier achieved the highest performance, with an accuracy of 70.29% and a precision of 70.05%. The UCO algorithm's ability to balance data using SMOTE and clustering proved better for traditional oversampling and undersampling methods, making it a valuable approach for imbalanced datasets in medical applications. The experiments confirmed that the combination of UCO and Random Forest provided the most accurate predictions for heart attack risk in stroke patients.

In [14] Abid et al., This study examines heart failure survival prediction using the Heart-Failure-Clinical-Records dataset from the UCI machine learning repository, including 299 patient records with 13 clinical features. To handle class imbalance, the Synthetic Minority Oversampling Technique (SMOTE) was employed, increasing instances from 97 to 300 per class. Nine machine learning models were tested: Decision Tree, AdaBoost, Logistic Regression, Stochastic Gradient Descent, Random Forest, Gradient Boosting, Extra Tree Classifier (ETC), Gaussian Naive Bayes, and Support Vector Machine. Feature ranking by Random Forest identified key predictors such as Time, Creatinine, Ejection Fraction, Age, Platelets, CPK, and Sodium. Among the models, ETC with SMOTE outperformed others, achieving 92.6% accuracy, 93% precision, recall, and F-Score. This research highlights the effectiveness of SMOTE and tree-based models in predicting heart failure survival and highlights the potential of machine learning in improving clinical outcomes.

In [15] Rohit et al., In this study on heart disease prediction, he utilized two datasets from Kaggle and the UCI repository. To address imbalanced class labels in the Kaggle dataset, he employed three sampling strategies: Random Oversampling, Synthetic Minority Oversampling (SMOTE), and Adaptive Synthetic Sampling (ADASYN). Then he applied feature selection techniques, using ANOVA F-value and Mutual Information, to identify the most relevant features. Various machine learning classifiers were evaluated based on these selected features. The hybrid ensemble model, enhanced with evolutionary feature selection, demonstrated the best performance. Specifically, the proposed model achieved 99% accuracy with Random Oversampling, 93% with SMOTE, and 91% with ADASYN, surpassing existing methods. This indicates the effectiveness of our approach in improving prediction accuracy on both datasets.

In [16] Muhammad et al., This study presents a cost-effective and reliable method for predicting heart attacks using the UCI heart attack dataset, which contains 303 records with 76 attributes. For prediction, a subset of 14 crucial attributes is selected and used 13. To address the class imbalance (164 positive and 139 negative instances), the Synthetic Minority Oversampling Technique (SMOTE) is employed. Without any feature engineering to avoid bias and reduce costs, the dataset is directly processed through various machine learning algorithms. Among these, the SMOTE-enhanced Artificial Neural Network (ANN), when properly tuned, outperforms other models and many existing systems. The use of Stratified K-Fold validation ensures robust and unbiased results. This approach demonstrates that handling imbalanced data effectively with SMOTE, combined with raw data inputs, can achieve high accuracy and reliability in predicting heart attacks.

- **Machine Learning Models in Healthcare – Heart Attacks**

Machine learning models are crucial in the medical field, especially for heart attack prediction, due to their ability to analyze huge amounts of complex data and uncover patterns that might not be visible through traditional statistical methods. These models can handle high-dimensional datasets, such as electronic health records, genetic information, and medical imaging, allowing for complete risk assessments. For heart attack prediction, machine learning algorithms can integrate various patient data like age, lifestyle, biomarkers, and historical medical data to accurately predict the likelihood of a heart attack. This capability enables early intervention and preventive measures, potentially saving lives and reducing healthcare costs. Also, these models can continuously learn and adapt from new data, improving their predictive power over time. Their application also helps in identifying risk factors which lead to better clinical insights and more informed decision-making. In our research, we identified the most used machine learning algorithms for heart attack prediction, particularly those effective in feature selection and handling imbalanced data. These include Logistic Regression (LR), Decision Trees (DT), Random Forests (RF), Support Vector Machine (SVM), Naïve Base (NB) and Artificial Neural Networks (ANN)

- **Examples in Heart Attacks Prediction**

In [17] S. Ajmal Mohamed et al., This research enhances the prediction of heart attack risk by incorporating 12 critical attributes (such as blood pressure, lack of exercise, diet, heredity, and blood test reports) into the system. Additionally, it provides treatments and specialist suggestions. Utilizing the CHS (Cardiovascular Health Study) dataset, which includes data from 5,201 elderly subjects and an additional unit of 687 African Americans. The study applies under-sampling using the Kennard Stone (KS) algorithm to balance the dataset for SVM and EDC-AIRS algorithms. The KS algorithm efficiently selects representative samples across the data domain, ensuring an unbiased dataset. Both SVM and EDC-AIRS algorithms are employed to predict myocardial infarction (MI) risk, identifying key biomarkers like cognitive function, physical health, and lifestyle changes as significant predictors.

In [18] I. O. Awoyelu et al., This study focuses on developing predictive models to assess the risk of heart attack using data-driven techniques. The models were constructed using Decision Tree, Naïve Bayes, and Bagging classifiers, employing hybrid feature selection to refine the input variables. By analyzing clinical data from 206 patients, including 91 males and 115 females aged 18–86, collected from Obafemi Awolowo University Teaching Hospitals Complex in Nigeria, the study aims to improve cardiologists' decision-making. The dataset was divided into training and testing subsets, with 10-fold cross-validation applied to evaluate model performance. The Naïve Bayes classifier achieved the highest accuracy at 87.86%. The feature selection involved a hybrid approach, combining filter-based techniques (like Information Gain Ratio) and wrapper-based methods, which reduced computational complexity while enhancing model performance. RapidMiner software was used to simulate these models. The proposed hybrid predictive model successfully identified critical risk factors, making it a valuable tool for early heart attack risk detection and aiding in targeted clinical interventions.

In [19] Muhammad Rizwan et al., This study introduces a machine learning framework for predicting heart attacks using a dataset of 303 instances and 13 out of 14 features from Kaggle. The data were shuffled and split into training (80%) and testing (20%) sets to ensure a random distribution of the labeled classes. Six different machine learning models were evaluated: Decision Tree, K-Nearest Neighbors (KNN), Logistic Regression, Random Forest, Extreme Gradient Boost (XGBoost), and Support Vector Machine (SVM). K-Nearest Neighbors (KNN) was the top-performing algorithm, achieving an accuracy of 90.16% and a recall of 87.09%, making it the best suited for heart attack classification among the tested models. KNN also demonstrated the highest precision and F1-Score. In contrast, the Support Vector Machine (SVM) model had the lowest performance, with an accuracy of 70.49% and a precision of 55%. Logistic Regression and Random Forest excelled in recall metrics. The study underscores the effectiveness of KNN for heart attack prediction, outperforming other models in overall accuracy and efficiency. This finding highlights KNN's robustness in diagnosing heart attacks, suggesting its potential for reliable clinical use.

In [20] Janaranjani et al., This study employs Naive Bayes, Decision Tree, and Weighted Associative Rule Mining (WARM) to model coronary artery disease (CAD). Naive Bayes stands out for its superior performance in heart disease prediction systems. Using a historical cardiac illness database, the system diagnoses patients and promotes healthier lifestyles. Decision Tree achieves a notable 99.5% accuracy in predicting heart disease across different datasets due to its simplicity and effective segmentation of data. This research underscores Decision Tree's efficacy in evaluating raw clinical data for assessing heart disease severity, highlighting its superiority over Naive Bayes and WARM in accuracy and predictive power.

In [21] Sushmita et al., The web-based graphic user interface utilizes the Naïve Bayes algorithm, achieving an accuracy score of 81.25% in classification tasks. Rapid Miner was employed to determine the most suitable algorithm among Naïve Bayes, Decision Trees, K-Nearest Neighbour, and Random Forest, with Naïve Bayes demonstrating the highest accuracy on the UCI dataset. This interface provides convenient access to the classifier for users, enhancing usability and accessibility.

In [22] Issam et al., This study, focused on developing predictive models for hospital mortality in patients with acute myocardial infarction (AMI). Using real data from 787 patients across two countries (603 from the Czech Republic and 184 from Syria), characterized by 24 variables, he addressed the challenge of imbalanced and incomplete data. His research compared various classifiers and Bayesian network models to predict mortality and understand variable relationships comprehensively. Specifically, he enhanced the Chow–Liu and Tree-Augmented Naive Bayesian (TAN) algorithms to effectively handle these data complexities. Among these approaches, TAN with the TANi adaptation emerged as the most effective, demonstrating robust performance in dealing with incomplete and imbalanced datasets.

In [23] Alshraideh et al., This study explores the integration of multiple artificial intelligence techniques as Random Forest, SVM, Decision Tree, Naive Bayes, and k-Nearest Neighbours (KNN) increased with Particle Swarm Optimization (PSO) for feature selection. The objective was to predict heart disease presence and evaluate classifier accuracy using a dataset of 486 patient records from Jordan University Hospital (JUH). Data preprocessing included feature scaling to ensure uniformity across variables with differing scales. The SVM classifier, combined with PSO for feature selection, achieved outstanding performance with an accuracy of 94.3%. These findings highlight SVM's effectiveness in risk stratification for heart disease, emphasizing its potential for enhancing early detection and personalized treatment strategies.

In [24] Chitra et al., This study evaluates various machine learning (ML) algorithms for predicting heart attacks, including Decision Trees, Random Forest, XGBoost, K-Nearest Neighbors, Support Vector Machines, and Logistic Regression. The analysis is based on datasets from Cleveland, UCI, and Kaggle, including medical histories with attributes like age, gender, blood pressure, and sugar levels. The study compares these algorithms' accuracy, precision, and recall identifying the best performer. XGBoost emerges with the highest accuracy at 89.9%, while Bayesian Optimized SVM achieves 99.9% precision and XGBoost 94.67% recall. This underscores XGBoost's superiority in heart attack prediction, suggesting future enhancements through hybrid ML methods combining Random Forest and XGBoost.

Conclusion

This literature review highlights the pivotal role of machine learning in predicting heart attack risk, emphasizing the critical importance of feature selection, handling imbalanced data, and evaluating machine learning models. Feature selection techniques, including filter, wrapper, and embedded methods, are essential for the improvement of predictive models and enhancing model accuracy by focusing on the most relevant variables. Effective feature selection not only improves computational efficiency but also helps the doctors in making decisions based on key risk factors. Handling imbalanced data is another crucial aspect, as real-world medical datasets often exhibit significant class imbalances. Techniques such as oversampling, undersampling, and hybrid approaches, along with cost-sensitive algorithms and probability tuning, are necessary to ensure that predictive models perform reliably, particularly for the minority class, which is often of greater clinical interest. Finally, the application of diverse machine learning models, including Logistic Regression, Decision Trees, Random Forests, Support Vector Machines, Naïve Bayes, and Artificial Neural Networks, showcases their potential in analyzing complex datasets to reveal patterns and enhance predictive accuracy. This review underscores the synergy between these techniques and models in developing robust tools for early heart attack detection and risk assessment, opening the path for more personalized and timely actions in healthcare. Integrating advanced data handling and modeling techniques into healthcare systems can significantly enhance patient outcomes, reduce costs, and provide deeper insights for heart disease.

3. Methodology

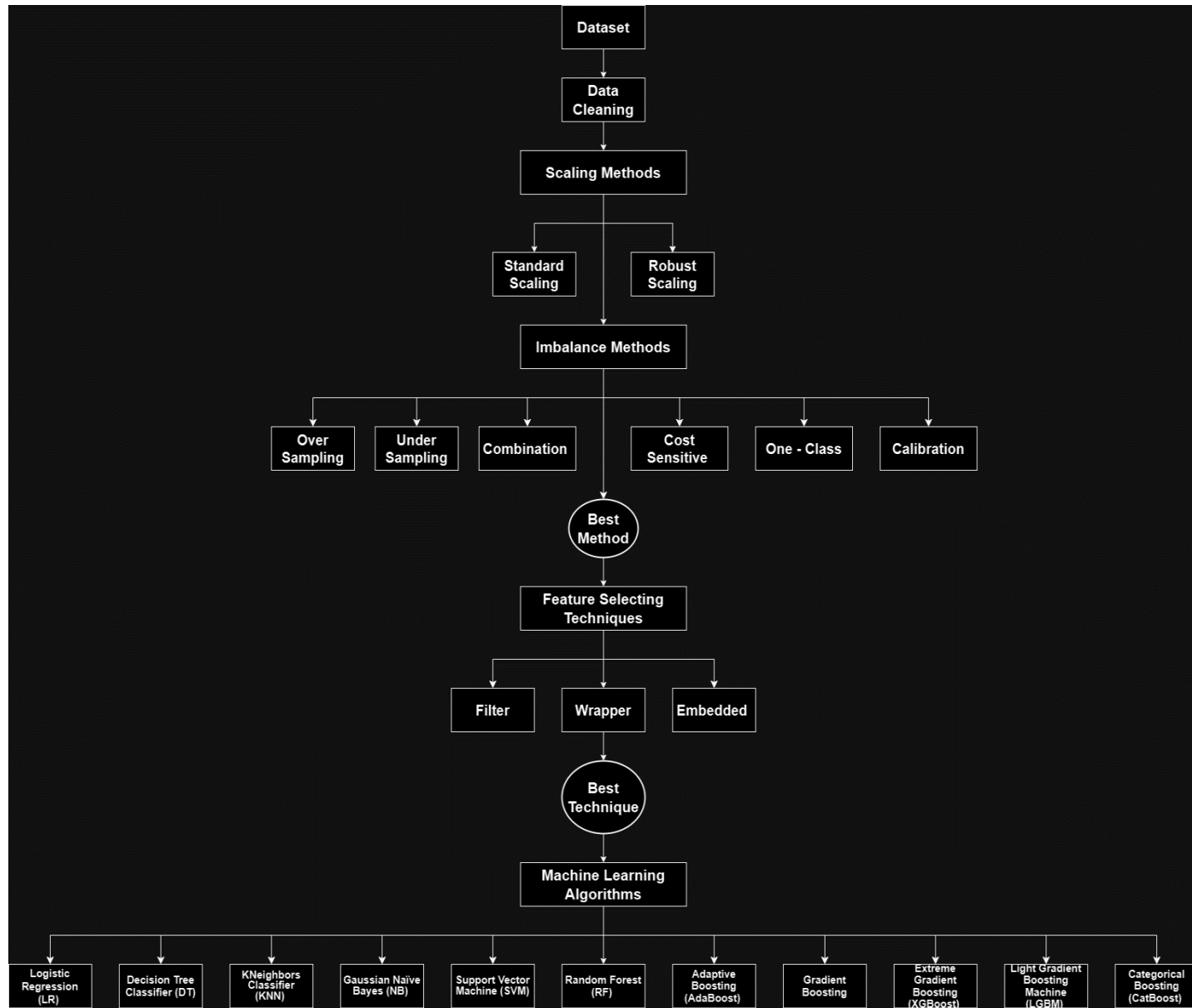
In this section of the dissertation, we will describe and analyze in deep depth the process and the plan that we followed, the specs and the tools that we used and all the different methods and techniques that we employed to achieve the final results. These methods include Scaling techniques for data normalization, Imbalanced data methods, Feature selection methods to identify the most relevant variables, the application of Machine Learning algorithms, the tuning techniques of the ML algorithms and the evaluation metrics. Each of these approaches plays a crucial role in the data, improving model's performance, and contributing to the accuracy and reliability of the predictive outcomes.

- **Report of the Process – Plan**

To achieve our final results, we followed the data cleaning process, and we moved with the application of various Machine Learning algorithms. Initially, we implemented various Machine Learning algorithms such as Logistic Regression (LR), Decision Tree Classifier (DT), KNeighborsClassifier (KNN), Naïve Bayes (NB), Support Vector Machine (SVM), Random Forest (RF), Adaptive Boosting (AdaBoost), Gradient Boosting, Extreme Gradient Boosting (XGBoost), Light Gradient Boosting Machine (LGBM) and Categorical Boosting (CatBoost) using the Standard Scaling method to ensure that features were on the same comparable scale. These models were evaluated using key performance metrics such as Precision, Accuracy, and AUC score. From our data analysis and initial evaluation results we could tell that the dataset was imbalanced, which lead us to apply a range of Imbalance Data Methods including Over-Sampling methods (Random Over Sampler (ROS), SMOTE and ADASYN), Under-Sampling methods (Random Under Sampler (RUS), Tomek Links (TL), Cluster Centroids (CC) and Near Miss), Combination (RUS & CC, CC & ADASYN, SMOTE & TL - SMOTETomek and CC & SMOTE), Cost-Sensitive method, One-Class methods (One-Class SVM and One-Class Isolation Forest) and Calibration method to our Machine Learning algorithms. After the implementation of these methods, we normalized the results and ranked our findings based on the evaluation metrics to identify the most effective imbalance handling technique.

Once the best imbalance data method was identified, we applied it to various Feature Selection Techniques like Filter (ANNOVA, Chi-Square, Fisher-Function, Fisher-skfeature and Mutual Info), Wrapper (Recursive Feature Elimination (RFE), Forward Selection (SFS) and Backward Elimination) and Embedded method (Lasso). We then normalized and ranked the results once again according to our evaluation metrics, selecting the most effective Feature Selection Method. With the optimal combination of Imbalance Data handling and Feature Selection Techniques identified, we focused on tuning the four most promising Machine Learning algorithms with various techniques like Grid Search Cross Validation, Randomized Search Cross Validation etc. to maximize their performance. This repeated process of evaluation and optimization allowed us to determine the best combination of Imbalance Data Method, Feature Selection Technique, and Machine Learning algorithm for those datasets, ensuring the most accurate and reliable predictive model.

Dendrogram of the Process - Plan



- **Specs and Tools**

We employed a laptop equipped with an Intel(R) Core(TM) i5-1035G1 CPU @ 1.00GHz (1.19 GHz) and 16.0 GB of RAM, running a 64-bit operating system with an x64-based processor. The primary tool that we used for the data processing and analysis was Jupyter Notebook (Anaconda 3), which helped us in various stages of the project, including Data Cleaning, the application of Imbalanced Data Methods, Feature Selection Techniques, and the implementation, tuning, and evaluation of Machine Learning algorithms. Additionally, Microsoft Excel was used for creating tables and generating graphs to visualize and represent the data and our results. Finally, we used Microsoft Word to organize and write the dissertation.

- **Scaling Methods**

On heart attack prediction, scaling methods playing a very crucial role in preparing the data for optimal model performance. Proper scaling ensures that all features contribute equally to the model, which is particularly important when using algorithms sensitive to feature size. To achieve the best precision in predicting heart attack cases, we employed two main scaling methods: Standard Scaling and Robust Scaling. Standard Scaling transforms the data by centering the features around zero with a standard deviation of one. This method is effective and secures that all the features with different units or scales are normalized, allowing the model to treat them equally. It works well in cases where the data distribution is approximately normal. Robust Scaling, on the other hand, is designed to handle outliers more beneficially by using the median and interquartile range (IQR) for scaling. This method is very useful in datasets where outliers might skew the results and also, makes sure that keeps the models robust and focused on the most relevant patterns. By applying these scaling techniques, we were able to increase the precision of the heart attack prediction models, ensuring that they were both accurate and flexible to variations in feature distributions. This careful preprocessing step was essential for achieving the high level of precision required for reliable heart attack predictions.

1. **Standard Scaling**

[25] Standard Scaling is a data preprocessing technique that standardizes features by removing the mean and scaling them to unit variance. In heart attack prediction, this method ensures that all features contribute equally to the model, despite their original scales. By changing the data to have a mean of zero and standardizing the variance, Standard Scaling improves the model's performance, particularly in algorithms that are sensitive to data magnitude. This technique is essential for improving the model's precision, ensuring that all the features are symmetrical and influence the same the prediction of heart attack cases.

2. **Robust Scaling**

[26] Robust Scaling is a preprocessing technique designed to handle datasets with outliers by scaling features using the median and interquartile range (IQR). In heart attack prediction, Robust Scaling makes sure that extreme values do not skew the data, making the model stronger to outliers. This method transforms the data around the median and scales it according to the IQR, which is less sensitive to outliers than the mean. By applying Robust Scaling, the model, we can keep high precision and accuracy, particularly in scenarios where the data may contain significant variations or anomalies.

- **Imbalanced Data Techniques**

[27] In machine learning, handling imbalanced data is crucial for developing accurate models. To address this, we employed various techniques and methods across six main strategies: Over - Sampling, Under - Sampling, Combination methods, Cost - Sensitive techniques, One - Class techniques and Calibration. For Over-Sampling, we utilized Random Over Sampler (ROS), SMOTE, and ADASYN to increase the minority class, ensuring that the model has equal examples of heart attacks and non-heart attacks cases. From Under - Sampling methods we applied Random Under Sampler (RUS), Tomek Links (TL), Cluster Centroids (CC), and Near Miss, which helped to balance the dataset by reducing the number of majority class examples. Also, combining these approaches can provide further improvements in data balance. We experimented with combinations such as RUS & CC, CC & RUS, CC & ADASYN, ADASYN & CC, SMOTE & Tomek Links, and CC & SMOTE, each offering unique advantages for the prediction models. Additionally, Cost - Sensitive learning was included to assign different weights to misclassification errors,

making the models more sensitive to heart attacks cases. From One - Class techniques, we explored One Class SVM and One Class Isolation Forest, which are effective in identifying heart attacks instances in highly imbalanced datasets. Finally, Calibration method was applied to adjust the probability estimates, enhancing the reliability and interpretability of the prediction model. These methodologies contributed to developing robust heart attack prediction models, capable of effectively handling the imbalance data.

1. Over – Sampling

Addressing the issue of class imbalance in heart attack prediction, Over - Sampling method (Random Over Sampler (ROS), SMOTE and ADASYN) plays a crucial role. By increasing the number of minority class instances, these techniques help the model learn from a more balanced dataset. This method can enhance the model's ability to recognize and predict heart attack cases accurately.

a) Random Over Sampler (ROS)

Random Over Sampler (ROS) is a straightforward and effective technique, used to address class imbalance in datasets. By randomly duplicating instances from the minority class, ROS increases their representation, ensuring that the model comes across these examples more frequently during training. This method helps to balance the dataset without introducing synthetic data, making it a simple and fast solution for improving the model's performance in predicting minority class events, such as heart attacks. Despite its simplicity, ROS can significantly enhance the sensitivity of the prediction model to rare but critical instances.

b) SMOTE

[28] SMOTE (Synthetic Minority Over-sampling Technique) is a powerful method to handle class imbalance by generating synthetic samples for the minority class. SMOTE (Synthetic Minority Over - sampling Technique) is a powerful method to handle class imbalance by generating synthetic samples for the minority class. Unlike random duplication, SMOTE creates new instances by adding between existing minority class examples, producing more diverse and informative samples. This technique enhances the dataset's balance, allowing the prediction model to better understand and recognize minority class patterns, such as heart attack cases. SMOTE's ability to introduce variety within the minority class makes it an effective tool for improving model performance and robustness in dealing with imbalanced data.

c) ADASYN

[29] ADASYN (Adaptive Synthetic Sampling) is an advanced Over - Sampling technique designed to address class imbalance by generating synthetic samples for the minority class. Unlike SMOTE, ADASYN focuses on creating more synthetic data for harder-to-learn examples, effectively adapting to the data distribution. This creates a more balanced and representative dataset, enhancing the model's ability to learn from minority class instances, such as heart attack cases. By prioritizing samples that are harder to classify, ADASYN improves the overall robustness and sensitivity of the prediction models, making it a valuable tool in handling imbalanced data.

2. Under – Sampling

Under-sampling methods (Random Under Sampler (RUS), Tomek Links (TL), Cluster Centroids (CC) and Near Miss) are essential for addressing class imbalance by reducing the number of majority class instances. These techniques help create a more balanced dataset, allowing the prediction model to focus equally on both classes. These methods increase the model's ability to accurately predict minority class events, such as heart attacks, by preventing bias towards the majority class.

a) Random Under Sampler (RUS)

Random Under Sampler (RUS) is a straightforward technique used to address class imbalance by reducing the number of majority class instances. By randomly removing examples from the majority class, RUS helps to balance the dataset, ensuring the prediction model does not become biased towards the majority class. This method is simple and effective, allowing the model to focus more on the minority class, such as heart attack cases. Despite its simplicity, RUS can significantly improve the model's ability to detect and predict rare events by providing a more balanced training set.

b) Tomek Links (TL)

Tomek Links (TL) is another Under - Sampling technique which aims at enhancing class balance by identifying and removing borderline instances between classes. A TL exists when two instances from different classes are each other's nearest neighbors. Removing these links helps to clean the boundary between classes, reducing overlap and ambiguity. This method improves the dataset's balance and clarity, allowing the prediction model to better distinguish between classes, such as heart attack cases and non-cases. By refining the decision boundary, Tomek Links enhance the model's accuracy and robustness in handling imbalanced data.

c) Cluster Centroids (CC)

[30] Cluster Centroids (CC) aim to address class imbalance by reducing the majority class through clustering. This method involves creating clusters of majority class instances and replacing them with their own centroids, thus, reducing the number of majority samples while preserving their distribution. By focusing on representative centroids, CC helps to balance the dataset, making it easier for the prediction model to learn and distinguish between classes. This technique is particularly useful in heart attack prediction as it ensures the model remains unbiased and performs well across both majority and minority classes.

d) Near Miss

Near Miss is an Under - Sampling technique designed to mark class imbalance by selecting majority class instances that are closest to the minority class. This method focuses on retaining majority samples that are near the minority instances, refining the decision boundary between classes. By keeping only those majority class examples that are difficult to classify, Near Miss elevates the model's ability to distinguish between the two classes. This approach is particularly beneficial in heart attack prediction, as it ensures that the model is well-trained on challenging examples, leading to improved accuracy and sensitivity in detecting rare but critical events.

3. Combination

Combination methods address class imbalance by integrating both over-sampling and under-sampling techniques, manipulating the strengths of each. These methods aim to balance the dataset and at the same time to increase the minority class instances and reduce the majority class examples. These hybrid approaches enhance the model's ability to learn from a well-balanced dataset, improving its accuracy and robustness in predicting heart attack cases. Combining methods helps to reduce the biases and limitations in individual sampling techniques.

a) Random Under Sampler (RUS) & Cluster Centroids (CC)

The combination of Random Under Sampler (RUS) and Cluster Centroids (CC) is a powerful method to address class imbalance by integrating the strengths of both techniques. RUS randomly removes majority

class instances, while CC replaces clusters of majority samples with their centroids. This dual approach reduces the dominance of the majority class while maintaining its distribution. In heart attack prediction, using RUS and CC together helps create a more balanced and representative dataset, enhancing the model's ability to accurately detect and predict heart attack cases by focusing on both under-sampling and data distribution.

b) Cluster Centroids (CC) & ADASYN

The combination of Cluster Centroids (CC) and ADASYN is a method to address class imbalance by blending under-sampling and over-sampling techniques. CC reduces the majority class by replacing clusters with their centroids, ensuring a more balanced distribution. ADASYN, on the other hand, generates synthetic samples for the minority class, focusing on harder-to-learn examples. This combination elevates the dataset's balance and diversity, allowing the heart attack prediction model to learn from a representative dataset. By integrating CC and ADASYN, the model becomes more robust and accurate in identifying heart attack cases, making good use of the strengths of both sampling strategies.

c) SMOTE & Tomek Links (TL)

The combination of SMOTE and Tomek Links (TL) or SMOTETomek is a famous technique frequently used to address class imbalance effectively. SMOTE generates synthetic samples for the minority class by adding between existing instances, enhancing the diversity and representation of minority data. Tomek Links then refine the dataset by identifying and removing borderline instances between classes, clarifying the decision boundary. This powerful combination ensures a balanced and clean dataset, improving the model's ability to accurately predict heart attack cases. By employing both over-sampling and under-sampling, SMOTE and TL work together to increase the model's performance and robustness in dealing with imbalanced data.

d) Cluster Centroids (CC) & SMOTE

The combination of Cluster Centroids (CC) and SMOTE is an effective strategy for addressing class imbalance by integrating under-sampling and over-sampling techniques. CC reduces the number of majority class instances by clustering and averaging them into centroids, ensuring the majority class is well-represented and minimized. SMOTE then generates synthetic samples for the minority class by inserting between existing instances, increasing its diversity and representation. This dual approach creates a balanced dataset, enhancing the heart attack prediction model's ability to learn from both classes. By combining CC and SMOTE, the model becomes more accurate and robust, capable of effectively handling imbalanced data.

4. Cost Sensitive

The cost-sensitive method is a critical approach for handling imbalance data, particularly in scenarios where the costs of misclassification are unequal. In heart attack prediction, the consequences of misclassifying a heart attack case can be serious. Cost-sensitive learning addresses this by assigning higher weights to the minority class, ensuring that the model pays more attention to predicting these critical instances. By including misclassification costs directly into the learning process, this method enhances the model's sensitivity and accuracy in detecting heart attack cases. That leads to better decision-making and outcomes in medical predictions.

5. One – Class

The one-class method is a specialized approach used for handling class imbalance by focusing only on the minority class. In heart attack prediction, this technique involves training the model exclusively on heart attack instances, learning their unique characteristics and patterns. By treating the minority class as the target and considering all other data as outliers, one-class methods, such as One-Class SVM and Isolation Forest, can be very good in identifying rare but critical events. This approach enhances the model's ability to detect heart attack cases with high precision, ensuring that the minority class receives the attention it requires for accurate and reliable predictions.

a) One Class SVM

One-Class SVM is a powerful technique used to handle imbalance data by focusing on the minority class. In heart attack prediction, this method involves training the model exclusively on heart attack instances to learn their specific characteristics. The One-Class SVM then identifies new data points that vary from this learned profile as outliers. This approach is particularly effective for detecting rare events, such as heart attacks, by accurately capturing the unique patterns of the minority class. By using One-Class SVM, the model becomes highly sensitive to heart attack cases, improving prediction accuracy and reliability.

b) One Class Isolation Forest (IF)

One-Class Isolation Forest is an effective technique for handling class imbalance by focusing on the minority class. In heart attack prediction, this method isolates anomalies by randomly selecting features and splitting data points. The model is trained exclusively on heart attack instances, learning their specific patterns. New data points that significantly differ from these learned patterns are identified as outliers. This approach is expert at detecting rare events, such as heart attacks, due to its ability to capture the unique characteristics of the minority class. Using One-Class Isolation Forest increases the model's sensitivity and accuracy in predicting heart attack cases.

6. Calibration

Calibration methods are essential for improving the reliability of predictive models by adjusting the probability estimates. In heart attack prediction, these techniques ensure that the model's predicted probabilities accurately reflect the true likelihood of an event. Calibration improves the interpretability and reliability of the model, making it very important in medical circumstances where precise risk assessment is crucial. Methods such as Platt Scaling and Isotonic Regression are commonly used to line up the predicted probabilities with actual outcomes. By implementing calibration, the model provides more accurate and meaningful predictions. Also, helps in better decision-making and patient care.

• Feature Selecting Techniques

[31] Feature selection plays a critical role in enhancing model precision by identifying the most relevant predictors. Effective feature selection reduces the dimensionality of the dataset, improving the model's performance and interpretability. We employed a variety of techniques across three main categories: Filter, Wrapper, and Embedded methods. From the Filter methods, I utilized ANNOVA, Chi-Square, Fisher Function, Fisher Skfeature, and Mutual Information. These statistical techniques rank features based on their relevance, helping to eliminate those that contribute little to the prediction of heart attack cases. For Wrapper methods, we implemented Recursive Feature Elimination (RFE), Forward Selection (SFS), and Backward Elimination. These approaches evaluate feature subsets by training the model multiple times, recognizing the optimal combination of features that maximizes precision. Finally, we used Lasso as an Embedded method, which combines feature selection within the model training process by applying regularization. Lasso penalizes less important features, reducing their coefficients to zero, and thus

selecting the most critical variables. These feature selection techniques contributed to refining the model, ensuring that only the most significant predictors were used, leading to improved precision in heart attack prediction.

1. Filter

[32] Filter methods are essential for feature selection, especially in high-dimensional datasets like those used in heart attack prediction. These techniques evaluate the relevance of each feature independently of the model by using statistical measures. These methods rank features based on their importance, allowing me to retain only those that have a significant impact on predicting heart attack cases. By using Filter methods, I developed the feature selection process, improving the model's efficiency and precision.

a) ANNOVA

[33] ANOVA (Analysis of Variance) is a statistical technique used in feature selection to evaluate the significance of differences between groups. In the context of heart attack prediction, ANOVA helps identify which features have the most impact on the target variable by comparing the means of different feature groups. This method is particularly useful for evaluating continuous features, allowing me to retain only those with strong biased power.

b) Chi-Square

[34] The Chi-Square test is a statistical method used in feature selection to evaluate the independence between categorical features and the target variable. In heart attack prediction, this test helps identify which categorical features are most strongly associated with the outcome by measuring how expected and observed frequencies differ. By applying the Chi-Square test, we were able to select features that have a significant relationship with heart attack occurrences, ensuring that the model focuses on the most impactful predictors.

c) Fisher – Function

The Fisher Function is a classification analysis technique used in feature selection to identify the features that best differentiate between classes. In heart attack prediction, the Fisher Function evaluates each feature's ability to recognize between patients who had a heart attack and those who did not. By calculating the ratio of between-class variance to within-class variance for each feature, this method highlights those with the most significant discriminatory power.

d) Fisher – Skfeature

[35] Fisher Skfeature is an advanced feature selection method based on Fisher's classification analysis, specifically designed to handle high-dimensional data. Especially in heart attack prediction, Fisher Skfeature evaluates the importance of each feature by considering its ability to recognize these different classes. This method is better than the traditional Fisher analysis by efficiently processing large datasets, ensuring that the most relevant features are selected.

e) Mutual Info

[36] Mutual Information (MI) is a powerful feature selection technique that measures the dependency between a feature and the target variable. In heart attack prediction, MI quantifies how much information a feature provides about the occurrence of a heart attack. Unlike linear methods, Mutual Information can capture non-linear relationships, making it particularly effective for complex datasets.

2. Wrapper

[37] Wrapper methods are a powerful approach to feature selection that involves evaluating feature subsets based on model performance. Wrapper methods consider the interaction between features by repeatedly selecting and evaluating different combinations. These methods allow us to identify the optimal set of features that maximize the model's precision, ensuring that the final model was both accurate and efficient in predicting heart attack cases.

a) Recursive Feature Elimination (RFE)

[38] Recursive Feature Elimination (RFE) is a very powerful and famous Wrapper method used for feature selection that systematically removes the least important features to improve model performance. In the heart attack prediction field, RFE works by recursively training the model, ranking features based on their importance, and eliminating the least significant ones in each iteration. This process continues until the optimal subset of features is identified.

b) Forward Selection (SFS)

Forward Selection (SFS) is a Wrapper method used for feature selection that builds the model step by step by starting with no features and adding them one by one. In heart attack prediction, SFS begins by selecting the single feature that most improves the model's performance. It then continues adding the next most significant feature in each step until no further improvement is observed. This approach helps identify the most impactful features, ensuring that the model is both accurate and efficient.

c) Backward Elimination

[39] Backward Elimination is a Wrapper method for feature selection that begins with all features in the model and then systematically removes the least significant ones. In heart attack prediction, this method involves iteratively fitting the model, evaluating the significance of each feature, and eliminating the least impactful one at each step. The process continues until only the most relevant features remain. Backward Elimination helps refine the model by focusing on the most critical predictors, enhancing its precision and reducing complexity.

3. Embedded Method

Embedded methods are a feature selection approach that integrates the selection process directly into the model training. Unlike Filter and Wrapper methods, Embedded methods consider the interaction between features during the model's learning process, making them more effective and efficient.

a) Lasso

[40] Lasso (Least Absolute Shrinkage and Selection Operator) is an Embedded method that performs feature selection by applying a regularization technique during model training. In heart attack prediction, Lasso works by adding a penalty to the regression coefficients, as a result decreasing less important feature coefficients to zero. This process automatically selects the most significant features, reducing the model's complexity while keeping accuracy and precision.

- **Machine Learning Algorithms (ML)**

Machine learning (ML) algorithms play a crucial role in predicting heart attacks. We employed a diverse range of machine learning algorithms to ensure the highest precision in predicting heart attack cases. Each algorithm was chosen for its unique strengths and ability to handle different characteristics of the data. Logistic Regression (LR) was used for its simplicity and interpretability, providing a strong baseline

for classification. Decision Tree (DT) gave us a crystal-clear model structure, allowing an easy explanation of decision-making processes and KNeighborsClassifier (KNN), was useful for capturing the patterns in the data. Also, GaussianNB (NB) is used for its probabilistic approach, which is very effective in handling categorical data. We employed Support Vector Machine (SVM) for its robustness in high-dimensional data and the ability to create optimal decision boundaries. Further, Random Forest helped us to reduce overfitting and improve accuracy. Also, we applied AdaBoost and GradientBoosting for their ability to boost weak learners, increasing overall model's performance. In addition to these, we utilized advanced boosting algorithms such as XGBoost, LightGBM (LGBM), and CatBoost, which are known for their efficiency and high accuracy in handling large datasets with complex relationships. By employing this wide array of algorithms, I aimed to compare and identify the best-performing model, focusing on achieving the highest precision in predicting heart attacks. This approach ensured that the selected model was both accurate and reliable.

1. Logistic Regression (LR)

[41], [42] Logistic Regression (LR) is a famous and widely used machine learning algorithm for binary classification tasks. In heart attack prediction, LR is known for its simplicity, interpretability, and ability to provide probabilistic outcomes. By modeling the relationship between features and the likelihood of a heart attack, LR offers very good insights into the impact of each predictor. Despite its simplicity, Logistic Regression is effective in providing a solid baseline for classification, making it a crucial tool in the initial stages of model development, particularly when precision is a key metric.

2. Decision Tree Classifier (DT)

[43] Decision Tree Classifier (DT) is a flexible machine learning algorithm that can be very powerful in handling both categorical and continuous data. In heart attack prediction, DT is useful for its transparency, where decisions are made based on a series of simple rules extracted from the data. Each branch of the tree represents a decision, making it easy to explain how the model has concluded its predictions. DT's ability to capture complex interactions between features and its straightforward visualization make it an effective tool for understanding the key factors influencing heart attack risk.

3. K – Neighbors Classifier (KNN)

[44] KNeighborsClassifier (KNN) is a simple and at the same time an effective machine learning algorithm that classifies instances based on the closest training examples in the feature space. In heart attack prediction, KNN works by assigning the most common label among the nearest neighbors to a new data point, making it extremely useful for capturing the patterns in the data. Its non-parametric nature allows KNN to adapt to various data distributions without assuming any underlying model. While KNN is straightforward, its ability to use the information makes it a valuable tool in predicting heart attack cases with precision, especially when relationships within the data are complex.

4. Gaussian Naïve Bayes (NB)

[45] Gaussian Naïve Bayes (NB) is a probabilistic machine learning algorithm that applies Bayes' theorem with the assumption of Gaussian distribution for continuous features. In heart attack prediction, Gaussian NB is extremely effective for handling numerical data and providing quick, understandable results. Despite its simplicity and the strong independence assumption between features, Gaussian NB often performs well, especially in cases where the data distribution aligns with its assumptions. This algorithm's ability to estimate probabilities and handle overlapping classes makes it a valuable tool for achieving high precision in predicting heart attack cases.

5. Support Vector Machine (SVM)

[46] Support Vector Machine (SVM) is a powerful and flexible machine learning algorithm used for classification tasks, particularly effective in high-dimensional spaces. In heart attack prediction, SVM works by finding the optimal hyperplane that best separates the classes, ensuring the maximum margin between the closest data points of each class. This ability to create clear decision boundaries makes SVM incredibly valuable for complex datasets where precision is crucial. SVM's flexibility, with its use of various kernels to handle non-linear data, enhances its effectiveness in accurately predicting heart attack cases, contributing to a robust and precise model.

6. Random Forest (RF)

[47] Random Forest (RF) is a very strong ensemble learning algorithm that combines multiple decision trees to improve classification accuracy and robustness. In heart attack prediction, Random Forest works by building a large number of decision trees during training and outputting the mode of their predictions. This approach helps to reduce overfitting and increases the model's generalization ability. By averaging the results of several trees, Random Forest successfully captures complex relationships between features, improving the model's precision and reliability. Its ability to handle large datasets and maintain high accuracy makes Random Forest a valuable tool in predicting heart attack cases.

7. Adaptive Boosting (AdaBoost)

[48] Adaptive Boosting (AdaBoost) is a high-powered ensemble technique that improves model accuracy by combining multiple weak learners, typically decision trees, into a strong classifier. In heart attack prediction, AdaBoost works by sequentially training these weak learners, each focusing on the instances that previous models misclassified. This adaptive process assigns higher weights to harder-to-predict cases, ensuring the model becomes continuously more accurate with each iteration. By emphasizing the most challenging examples, AdaBoost improves the overall precision of the model, making it extremely effective for detecting subtle patterns in heart attack data.

8. Gradient Boosting

[49] Gradient Boosting is a powerful ensemble learning technique that builds a strong predictive model by combining multiple weak learners, typically decision trees, in a sequential way. In heart attack prediction, Gradient Boosting works by repeatedly improving the model, with each new tree correcting the errors made by the previous ones. This method successfully captures complex relationships within the data, making it highly effective for increasing model precision. By focusing on minimizing the errors of previous models, Gradient Boosting elevates the accuracy and robustness of predictions, making it a useful tool for accurately identifying heart attack cases.

9. Extreme Gradient Boosting (XGB)

[50] Extreme Gradient Boosting (XGBoost) is an advanced application of Gradient Boosting that is known for its speed and performance. In heart attack prediction, XGBoost assembles multiple decision trees sequentially, with each tree aiming to correct the errors of its prior's trees. It includes regularization techniques to turn aside overfitting, making the model both accurate and generalizable. XGBoost's ability to handle large datasets and complex interactions between features, along with its efficient computation, makes it a powerful tool for achieving high precision in heart attack predictions.

10. Light Gradient - Boosting Machine (LGBM)

[51] Light Gradient Boosting Machine (LGBM) is an extremely efficient and scalable application of Gradient Boosting, designed to handle large datasets with speed and accuracy. In heart attack prediction, LGBM can be excellent by using a leaf-wise growth strategy, which allows it to focus on the most significant splits, leading to better model performance. Its ability to manage complex data patterns while maintaining low memory usage makes LGBM very effective for high-dimensional data. By offering fast training and precise predictions, LGBM is a valuable tool for increasing the accuracy and efficiency of heart attack prediction models.

11. Categorical Boosting (CatBoost)

[52] Categorical Boosting (CatBoost) is a highly developed Gradient Boosting algorithm designed to handle specifically the categorical features efficiently. In heart attack prediction, CatBoost can be very good by automatically processing the categorical variables without requiring any other preprocessing, such as one-hot encoding. This ability reduces the risk of overfitting and improves model performance on complex datasets. CatBoost also offers fast training times and robust accuracy, making it particularly effective for predicting heart attack cases with high precision.

• Evaluation Metrics

On heart attack prediction, evaluating the performance of the predictive models is very important. Therefore, to secure their accuracy and reliability we employed several key evaluation metrics such as the Accuracy, the Precision, and the Area Under the Curve (AUC) score. Each of these metrics provided unique insights into the model's performance, especially in the area of predicting heart attack cases where precision is crucial. By combining these evaluation metrics, we were able to estimate in deep depth the models, making sure at the same time that they not only performed well overall but were particularly successful in predicting heart attack cases with the highest possible precision.

1. Accuracy

Accuracy is an essential evaluation metric that measures the proportion of correctly classified instances out of the total predictions made by the model. In heart attack prediction, accuracy provides a general overview of how well the model performs by calculating the percentage of correct predictions across both positive and negative cases. While it is a useful indicator of overall performance, accuracy alone may not fully capture the model's effectiveness in handling imbalanced datasets, where correctly identifying heart attack cases (the minority class) is more critical.

The equation for accuracy is:

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

True Positives (TP): The number of instances correctly predicted as the positive class.

True Negatives (TN): The number of instances correctly predicted as the negative class.

False Positives (FP): The number of instances incorrectly predicted as the positive class (when they actually belong to the negative class).

False Negatives (FN): The number of instances incorrectly predicted as the negative class (when they actually belong to the positive class).

2. Precision

Precision is a pivotal evaluation metric that measures the accuracy of positive predictions, specifically focusing on the proportion of true positives (correctly predicted heart attack cases) out of all positive predictions made by the model. In heart attack prediction, high precision is extremely important as it ensures that when the model predicts a heart attack, it is highly likely to be correct. This reduces the risk of false positives, making the model more reliable and effective in critical medical decision-making. Therefore, in a diagnostic model like heart attack predictions, we want to find all the people who have had heart attack (minority), so it is not so important if our model diagnoses some healthy people as with heart attack. Precision metric is extremely useful in this industry, as it highlights the model that gives higher costs due to unnecessary additional tests. A higher precision indicates fewer False Positives (FP), reducing the number of unnecessary tests.

The equation for precision using true positives (TP) and false positives (FP) is:

$$\text{Precision} = \frac{TP}{TP + FP}$$

True Positives (TP): The number of instances correctly predicted as the positive class.

False Positives (FP): The number of instances incorrectly predicted as the positive class (when they actually belong to the negative class).

3. AUC – Score

The AUC-ROC score, which stands for Area Under the Receiver Operating Characteristic Curve, is a valuable metric that evaluates model performance. It advantages true positive rates (TPR) and false positive rates (FPR) to measure how well the model distinguishes between classes. Automatically, it reflects the proportion of positive data points correctly identified by the model out of all positive data points. In other words, a higher TPR means fewer positive cases are missed. The AUC score is a holistic evaluation metric that measures the model's ability to determine between positive and negative classes across various thresholds. In heart attack prediction, a higher AUC score indicates that the model is effective at ranking true positives (actual heart attack cases) higher than false positives, reflecting its overall determination power. The AUC score is particularly valuable because it provides insight into the model's performance independent of any specific decision threshold, offering a robust assessment of its predictive capabilities. In general, an AUC of 0.5 indicates no discriminatory ability (i.e., the test cannot distinguish between patients with and without the disease or heart attack). An AUC between 0.7 and 0.8 is considered as acceptable, 0.8 to 0.9 is rated as excellent, and an AUC above 0.9 is regarded as outstanding.

The equation for AUC – Score:

$$AUC = \int_0^1 TPR(FPR) d(FPR)$$

True Positive Rate (TPR): $TPR = \frac{TP}{TP + FN}$

False Positive Rate (FPR): $FPR = \frac{FP}{FP+TN}$

- **Normalization of the Metrics**

Normalization is a process of scaling data values to a common range, between 0 and 1. By normalizing the data, you ensure that all features and attributes contribute equally when comparing and ranking them.

The mathematical equation for normalization is:

$$\text{Normalized Value} = \frac{(X - X_{min})}{(X_{max} - X_{min})}$$

- **Tuned Techniques**

1. Grid Search Cross Validation

[53] Grid Search Cross Validation is a systematic tuning technique used to find the optimal hyperparameters for a machine learning model. In heart attack prediction, this method contains very carefully searching through all possible combinations of predefined hyperparameters and evaluating model performance using cross-validation. By in deep depth exploring the parameter space, Grid Search ensures that the model is fine-tuned to achieve the highest precision, making it a crucial step in developing an accurate and reliable heart attack prediction model.

2. Randomized Search Cross Validation

[54] Randomized Search Cross Validation is an efficient hyperparameter tuning technique that randomly samples a specified number of combinations from a predefined set of hyperparameters. In heart attack prediction, this method allows for a broad exploration of the parameter space without the detailed computational cost of Grid Search. By testing random combinations, Randomized Search can quickly identify high-performing hyperparameters, leading to a well-tuned model that balances precision and efficiency. This approach is particularly useful when dealing with large or complex models where full grid search would be impractical.

3. Cross Validation (CV)

[55] Cross Validation (CV) is a crucial technique used to evaluate the generalizability of a machine learning model by rating its performance on different subsets of the data. In my heart attack prediction study, I used 5-fold Cross Validation (CV= 5), where the dataset is split into five equal parts. The model is trained in four parts and tested on the remaining one, rotating this process until each part has been used as a test set. This approach helps ensure that the model's performance is consistent and reliable across different data splits, leading to a more accurate and robust prediction model.

4. [53] Grid Parameters

a) max_depth

The hyperparameter max_depth controls the maximum number of levels or layers in a model, controlling how deep the model can go in splitting the data. Adjusting max_depth helps balance the trade-off between model complexity and overfitting, ensuring the model captures relevant patterns without becoming too complex.

b) gamma

gamma in machine learning models controls the effect of a single training example on the model's decision boundary. A higher gamma value focuses the model more on specific points, potentially leading to overfitting, while a lower gamma allows for a broader, more generalized decision boundary.

c) n_estimators

The hyperparameter n_estimators determines the number of trees in ensemble models like Random Forest and Gradient Boosting. Increasing n_estimators typically improves model accuracy by combining the predictions of more trees, but it also increases computational cost and the risk of overfitting if not carefully managed.

d) learning_rate

learning_rate controls the step size at each iteration while training a model, especially in gradient-based algorithms like Gradient Boosting and Neural Networks. A smaller learning_rate allows the model to learn more slowly and steadily, which can lead to better convergence and accuracy, while a larger learning_rate speeds up the training but may risk overpassing the optimal solution.

e) max_features

The hyperparameter max_features in machine learning models decides the maximum number of features to consider when making a split at each node. By limiting max_features, you can reduce model variance and improve generalization, as the model is forced to consider a diverse subset of features, stopping reliance on any single feature.

f) criterion

criterion specifies the function used to measure the quality of a split. Common criteria include gini and entropy for classification tasks, or mse (mean squared error) for regression. The choice of criterion affects how the model decides to split the data, impacting its accuracy and interpretability.

g) class_weight

class_weight in classification models assigns different weights to different classes to handle class imbalances in the data. By adjusting class_weight, you can give more importance to minority classes, helping the model to perform better on imbalanced datasets by reducing bias towards the majority class.

h) subsample

The hyperparameter subsample in ensemble learning algorithms, such as Gradient Boosting, specifies the snippet of the training data to be used for fitting each base learner. By setting subsample to a value less

than 1.0, you introduce randomness into the model, which can help to avoid overfitting and improve generalization by reducing the correlation between individual models.

i) min samples split

min_samples_split defines the minimum number of samples required to split an internal node. Setting a higher value for min_samples_split makes the model more conservative by stopping it from splitting nodes with fewer samples, which helps in controlling overfitting and ensuring the model focuses on more substantial patterns in the data.

j) min samples leaf

The hyperparameter min_samples_leaf defines the minimum number of samples that must be present in a leaf node. Setting a higher value for min_samples_leaf ensures that leaf nodes contain more data, which can help prevent overfitting by making the model more robust and focusing on more reliable splits.

k) min child weight

The hyperparameter min_child_weight in gradient boosting algorithms, like XGBoost, control the minimum sum of instance weights (or the minimum number of samples) required in a child node. A higher min_child_weight value makes the model more careful by requiring more samples in leaf nodes, which helps prevent overfitting and ensures that the model focuses on significant splits.

l) colsample bytree

The hyperparameter colsample_bytree in ensemble methods like XGBoost specifies the snippet of features to be randomly selected for each tree. By controlling colsample_bytree, you can have diversity among the trees, which can help to avoid overfitting and improve the model's robustness by ensuring that each tree considers only a subset of the available features during training.

4. Datasets - Data Analysis

In this part of the dissertation, we will discuss the two datasets that we employed in our study. We will provide a complete description and analysis of these datasets, with the goal of identifying key factors, potential patterns, and examining correlations and relationships between the features and the target variable. Additionally, we will explore the interrelationships among the features themselves to gain a deeper understanding of the data and its structure.

1. Heart Attack Risk Prediction Dataset

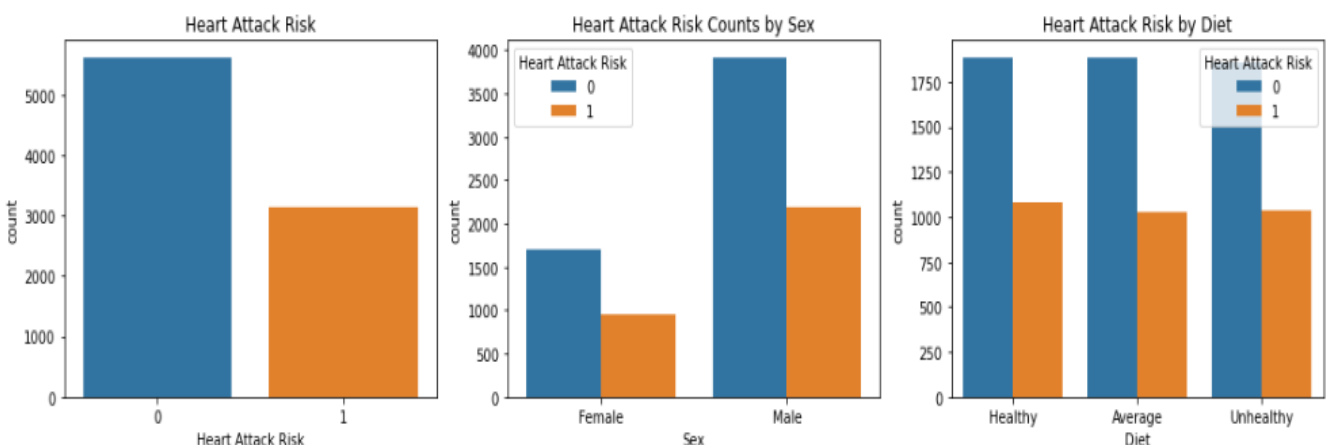
The first data set that we used for this study is 'Heart Attack Risk Prediction' from Kaggle. This dataset has been artificially made with ChatGPT to imitate a realistic environment. It serves as a valuable platform for beginners, offering them the opportunity to explore, experiment, and develop their skills in a controlled, simulated setting, ultimately deepening their understanding of these concepts. By working with this dataset, users can immerse themselves in scenarios that closely resemble those found in the real world. This dataset focuses on predicting the risk of heart attacks using a variety of features that summarize a broad spectrum of variables crucial to cardiovascular health. The dataset, contains 8,763 synthetic patient

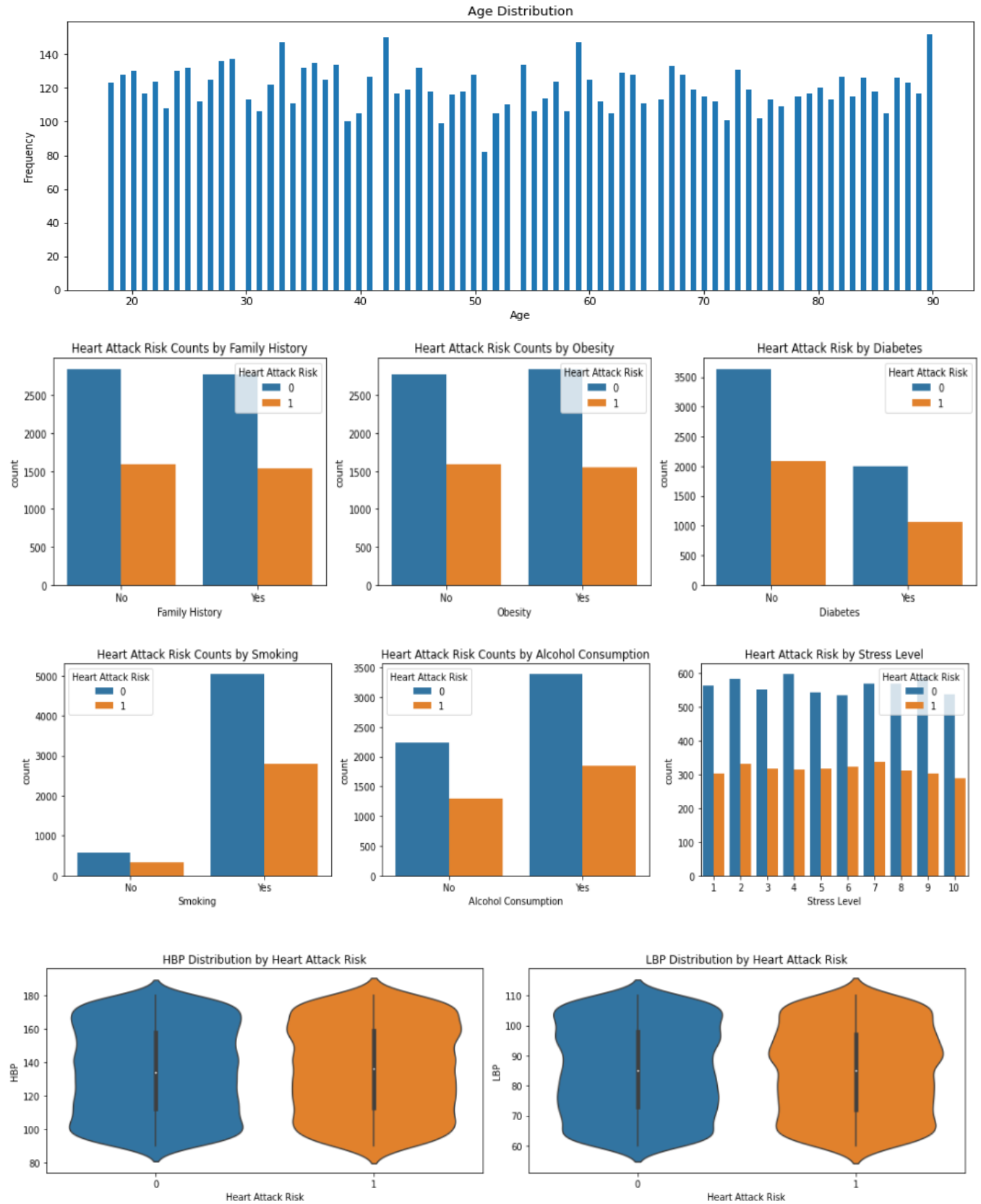
records from diverse geographical regions, gives a complete array of features that reflect both medical and lifestyle factors, providing a robust foundation for predictive analytics and machine learning applications. The 26 variables include demographic details such as age, gender, and income. Also, critical medical indicators like cholesterol levels, blood pressure, heart rate, diabetes status, and family history of heart disease. These medical variables are combined by lifestyle factors such as smoking habits, alcohol consumption, exercise hours per week, dietary habits, stress levels, and sedentary hours per day. Moreover, the dataset also considers socioeconomic factors and geographical attributes, including the patient's country, continent, and hemisphere, which add a contextual dimension to the analysis.

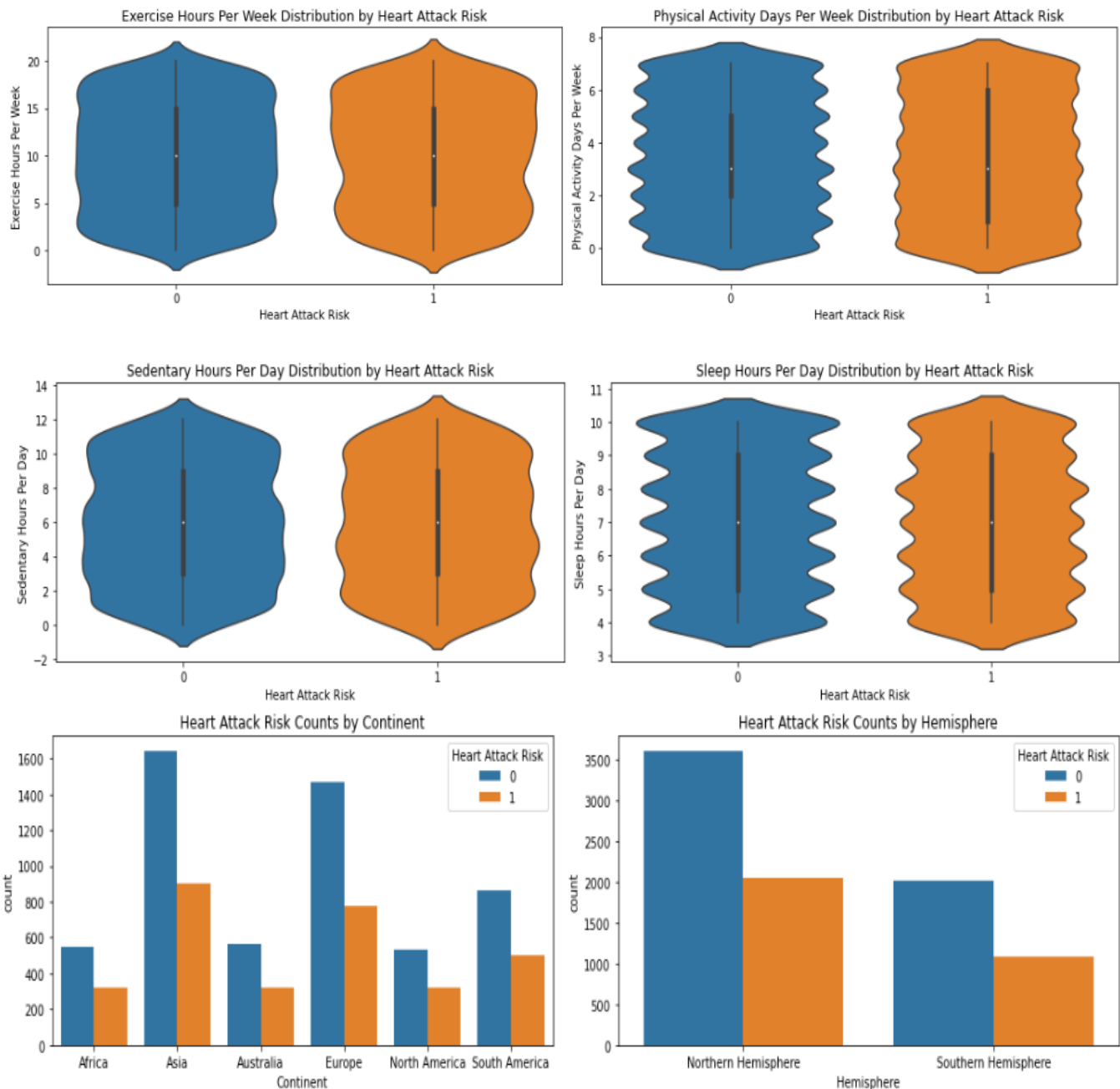
A key strength of this dataset is its binary classification feature, which indicates the presence or absence of heart attack risk, allowing the development of predictive models aimed at early detection and prevention of heart disease. This feature is particularly valuable as it allows for the application of various machine learning algorithms to identify patterns and correlations among the different variables, increasing the understanding of how these factors collectively influence heart health. The dataset's detailed structure includes specific variables such as BMI, triglyceride levels, physical activity days per week, sleep hours per day, and previous heart problems, which are essential for a comprehensive analysis of heart attack risk.

The descriptive statistics of the dataset reveal a mean age of approximately 54 years, with a standard deviation of 21 years, indicating a wide age range among the patients. Also, the dataset shows that most patients, 64.2%, are at no immediate risk of a heart attack and the patient population is mostly male, comprising 69.7% of the records. Cholesterol levels average around 260 mg/dL, with significant variability, while heart rates average 75 beats per minute. The dataset also highlights key risk factors such as smoking, with 89.7% of the patients identified as smokers, and diabetes, affecting 34.8% of the patients. These statistics underscore the importance of addressing lifestyle and medical factors in heart disease prevention strategies.

In conclusion, this dataset works as a vital resource for predictive analysis in cardiovascular health, providing insights that can drive the development of proactive strategies for heart disease prolepsis and management. The dataset's nature, combined with its global scope, makes it a powerful tool for advancing research in heart attack prediction and contributing to a healthier future.





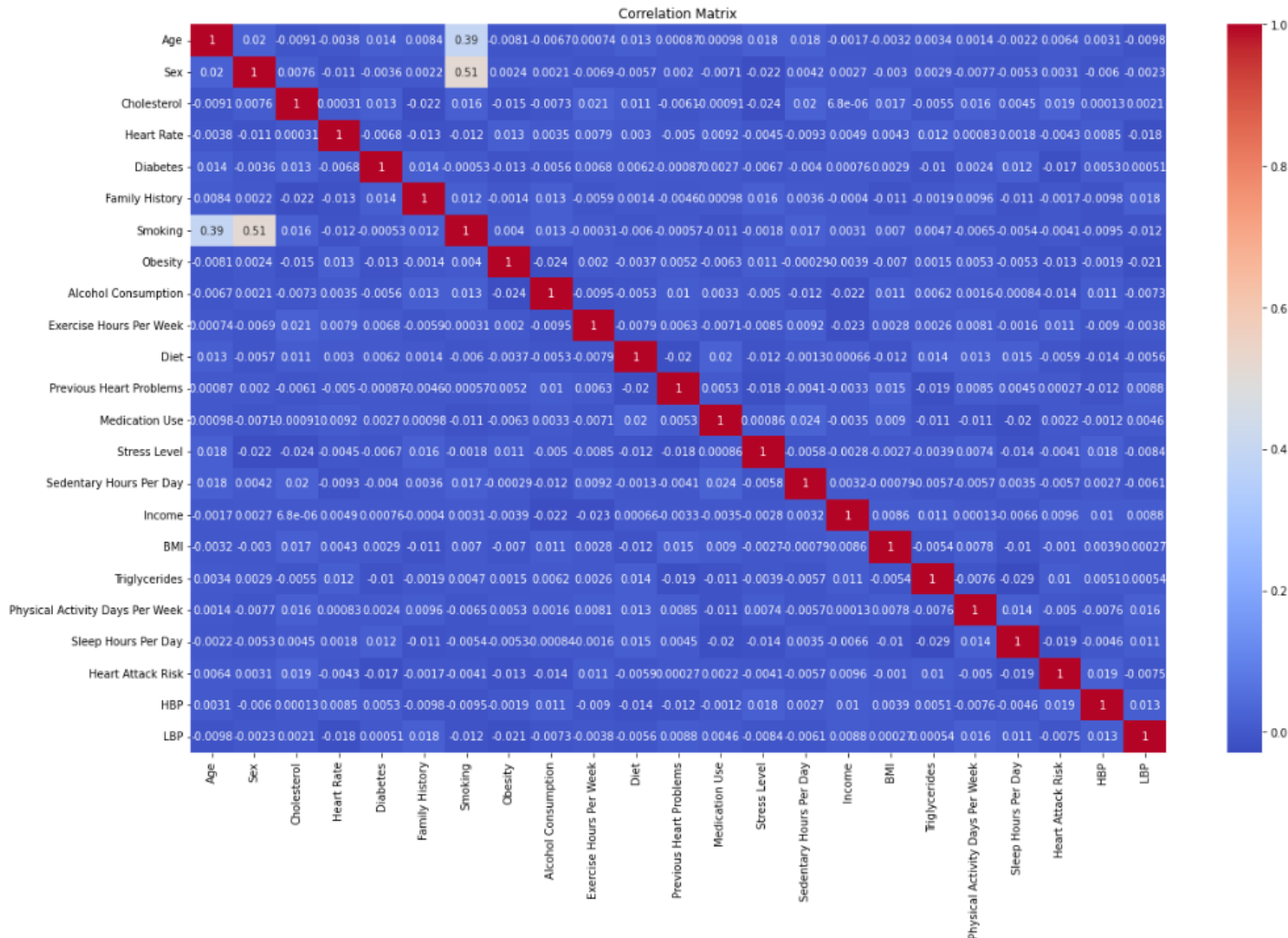


Correlation Matrix

The correlation matrix of the dataset offers valuable insights into the relationships between various features related to heart attack risk. More specifically, the matrix reveals a significant positive correlation between smoking and age (0.394), suggesting that older individuals in the dataset tend to have a higher likelihood of being smokers. Similarly, smoking also shows a solid correlation with sex (0.515), indicating that smoking habits are more prevalent among a particular gender in the dataset. On the other hand, correlations between other features such as cholesterol levels, heart rate, and diabetes with heart attack risk are relatively weak, with coefficients close to zero, highlighting the complexity and multifactorial nature of predicting heart attack risk. Also, the correlation between exercise hours per week and heart attack risk is slightly positive (0.011), suggesting that physical activity alone may not significantly reduce the risk.

without considering other factors. Additionally, the matrix reveals moderate correlations between socioeconomic factors, such as income and lifestyle-related variables, underscoring the interconnectedness of these aspects in determining cardiovascular health. Overall, the correlation matrix serves as a crucial tool for identifying key variables that could contribute to heart attack risk, guiding the focus of predictive models and further analysis in this study.

Correlation Matrix of Heart Attack Risk Prediction Dataset



2. Heart Disease Indicators Dataset

The second data set that we used for this study is 'Heart Disease Indicators' from Kaggle and provides complete health-related data. This dataset has been synthetically generated using ChatGPT to replicate a realistic environment, making it an excellent resource for beginners. It provides a safe and controlled setting where users can explore and experiment with their skills, helping them to be better in data analysis and machine learning concepts. Through interaction with this dataset, users can engage in scenarios that closely mirror real-world situations, allowing them to apply theoretical knowledge to practical, lifelike problems. The dataset is designed to predict heart attack risk by including a wide range of features that capture essential variables related to cardiovascular health. This dataset provides a huge array of health-

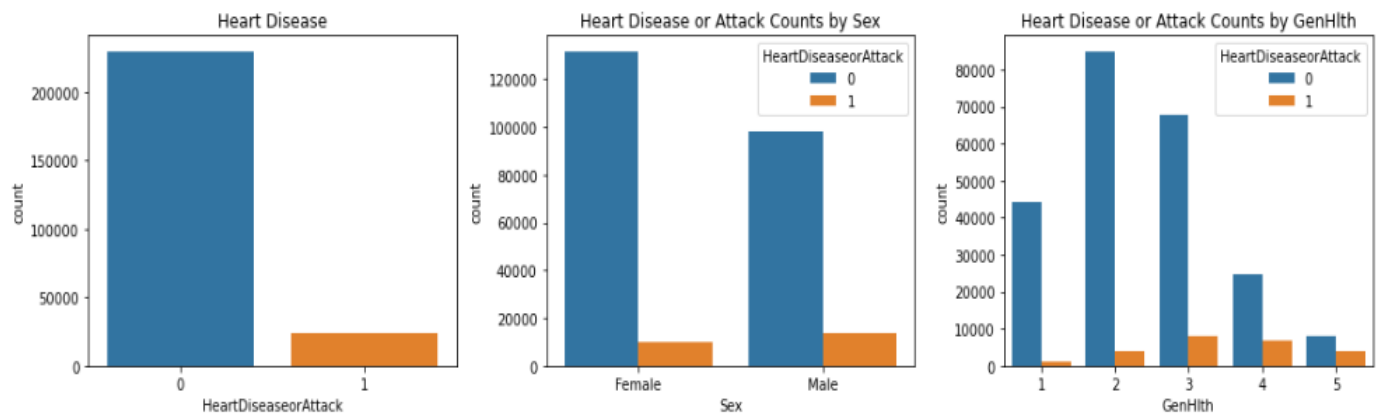
related indicators, lifestyle factors, and demographic information for a large group of individuals. Initially included 253,661 synthetic instances with 22 features and then we reduced it to a sample size of 50,000 instances to streamline and smooth the analysis. This reduction maintains the dataset's diversity and statistical significance, ensuring that the findings remain robust and generalizable.

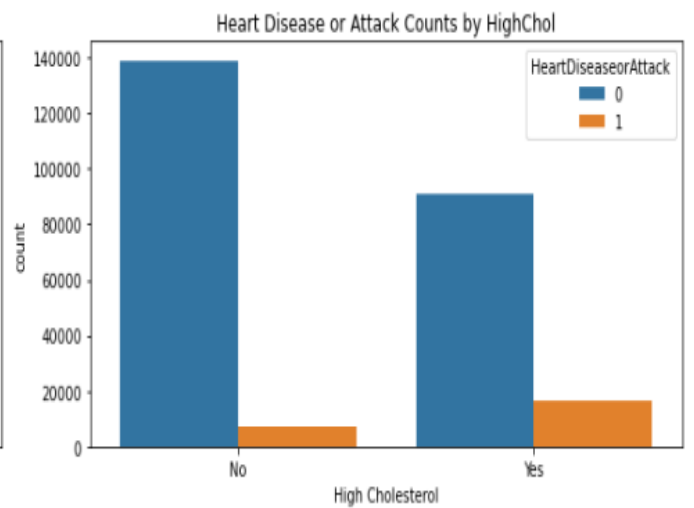
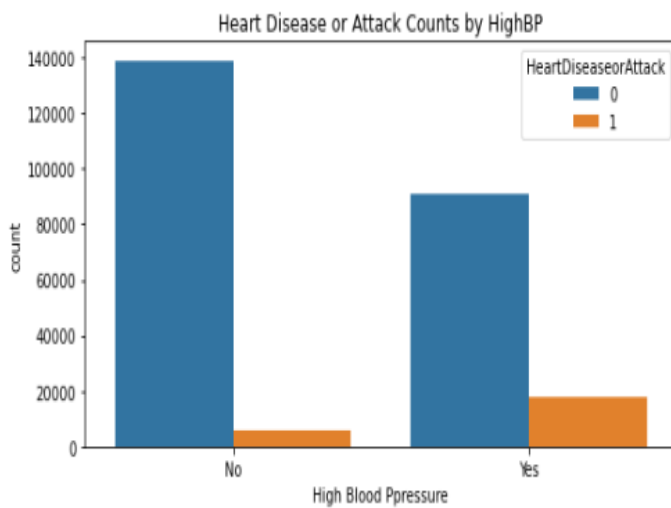
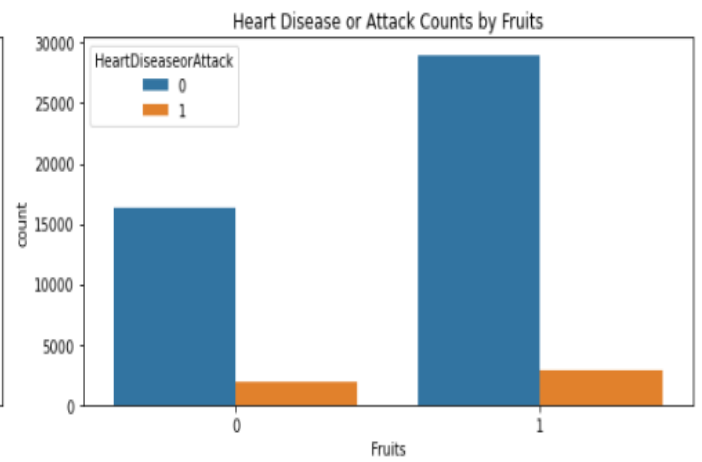
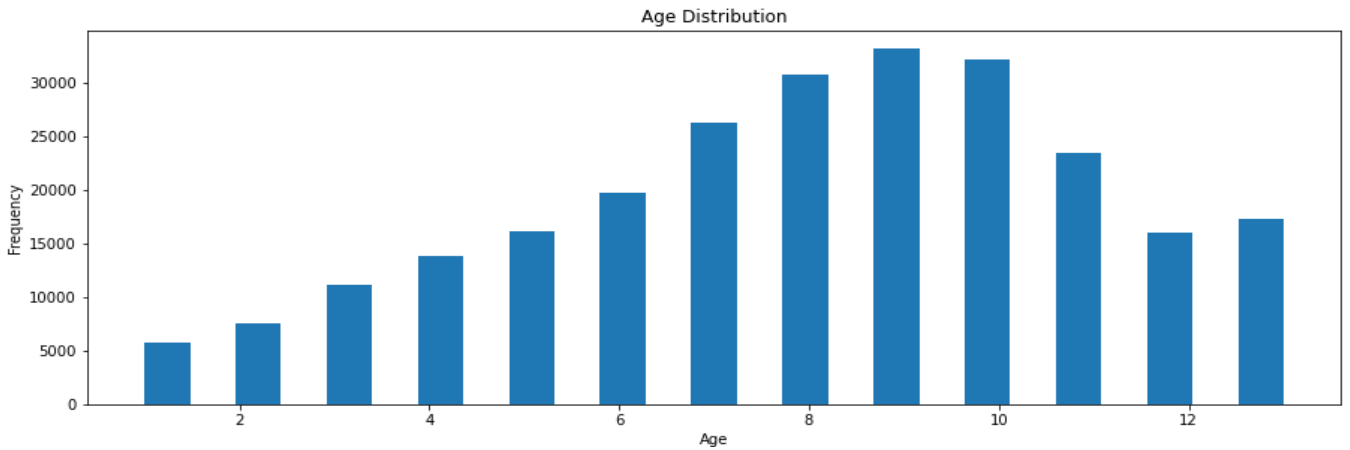
Key variables of the dataset were 'HeartDiseaseorAttack', which gives as the primary outcome variable, indicating whether an individual has experienced a heart disease or heart attack (binary: 0 = No, 1 = Yes). The popularity of heart disease or attack within the sampled dataset is approximately 9.4%, with 4,709 individuals affected out of the 50,000 sampled. Other critical variables include 'HighBP' (High Blood Pressure), 'HighChol' (High Cholesterol), and 'BMI' (Body Mass Index), all of which are well-known risk factors for cardiovascular diseases. For example, 42.5% of the sampled population reported having high blood pressure, and 42.3% reported high cholesterol levels, highlighting the widespread presence of these risk factors.

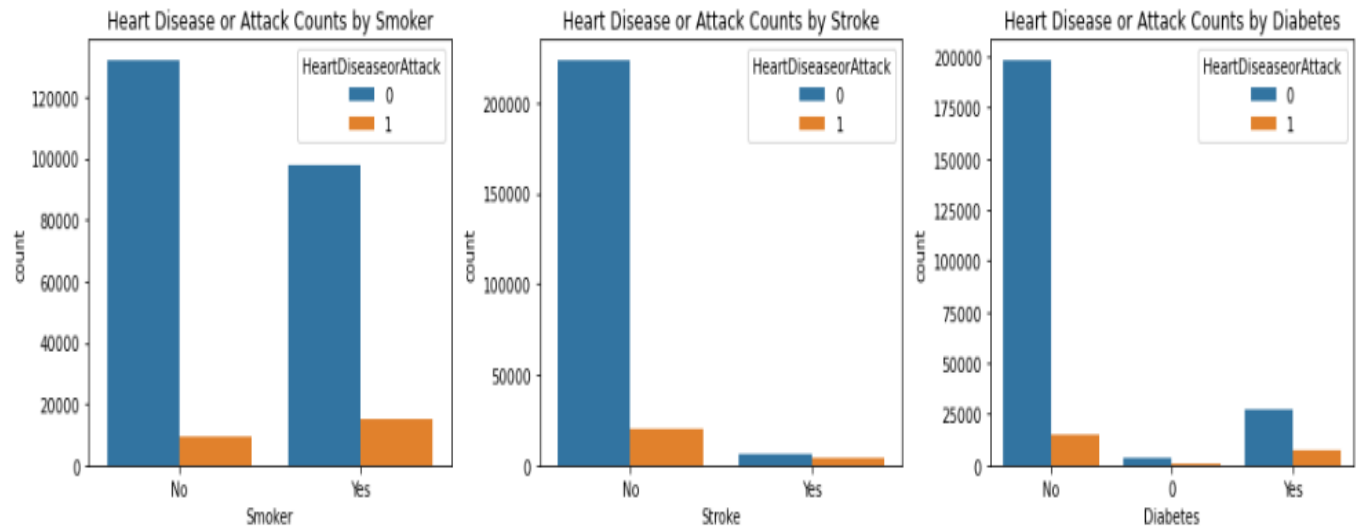
Lifestyle factors such as smoking, physical activity, and diet are also featured. The dataset reveals that 43.9% of individuals are smokers, while 75.9% engage in some form of physical activity. Diet habits are captured through variables like 'Fruits' and 'Veggies', indicating the frequency of fruit and vegetable consumption. Particularly, 81.1% of the sampled individuals regularly consume vegetables, and 63.6% consume fruits, reflecting relatively healthy dietary patterns within the sample.

Additionally, the dataset includes variables related to healthcare access and socioeconomic status, such as 'AnyHealthcare', 'NoDocbcCost', 'Education', and 'Income'. An overwhelming 95.1% of individuals have access to healthcare, though 8.5% have reported going without a doctor visit due to cost, which could influence their overall health outcomes. The educational level and income distribution within the dataset are also captured, providing context for understanding the social determinants of health.

The dataset's demographic variables, such as age and sex, improve the analysis. The gender distribution is slightly skewed, with 43.8% males and 56.2% females. Also, the 'Age' appears as a categorical column from 1 to 13 with 1 younger age and 13 older. The mean age within the sample is approximately 8 so we can suggest we speak approximately of 50 years old people, with a standard deviation of 3 which can say is reflecting around 20 years. Overall, this dataset offers a complete base for predictive modeling and analysis in heart attack prediction, capturing the interaction between medical, lifestyle, and socioeconomic factors that contribute to cardiovascular risk. The insights derived from this analysis could inform targeted interventions and healthcare strategies aimed at reducing the incidence of heart disease.





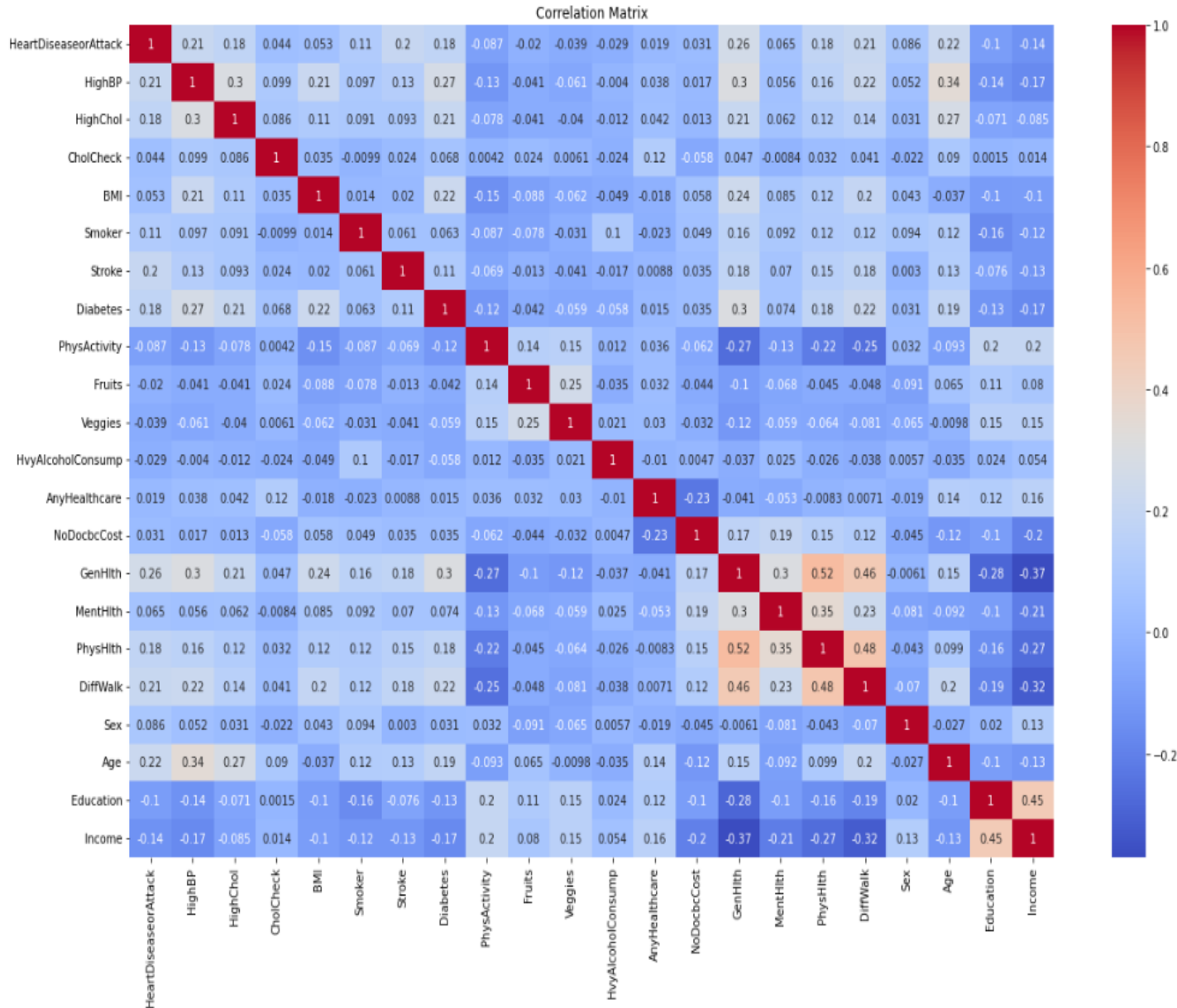


Correlation Matrix

The correlation matrix of the dataset reveals several crucial relationships among the variables, providing insights into the factors that contribute to heart disease or heart attack risk. Especially, 'HeartDiseaseorAttack' shows a moderate positive correlation with variables like 'HighBP' (0.21), 'Stroke' (0.21), and 'Age' (0.22), indicating that individuals with high blood pressure, a history of stroke, or older age are more likely to have heart disease or experience a heart attack. Additionally, there is a slightly important correlation between 'HeartDiseaseorAttack' and 'GenHlth' (0.26), suggesting that individuals with poorer general health are at a higher risk of heart disease.

Opposite, 'PhysActivity' shows a negative correlation with 'HeartDiseaseorAttack' (-0.09), underscoring the protective effect of physical activity against heart disease. Similarly, socioeconomic factors such as 'Income' (-0.14) and 'Education' (-0.09) are negatively correlated with heart disease risk, highlighting the potential impact of higher income and education levels in reducing cardiovascular risk. The matrix also shows that 'BMI' and 'Smoker' have weaker correlations with heart disease risk, emphasizing that while these factors are important, their impact may be less direct compared to other variables like blood pressure and stroke history. Overall, the correlation matrix serves as a crucial tool for identifying significant predictors of heart disease, guiding the development of effective risk assessment models.

Correlation Matrix of Heart Disease Indicators Dataset



5. Datasets - Cleaning & Preprocessing

Data cleaning and preprocessing are critical steps in the heart attack prediction process, as they ensure the accuracy and reliability of the model. By removing errors and handling missing values, we create a dataset that is suitable for analysis avoiding duplicates and potential bias. This process is essential for

achieving high precision, as a clean and well-preprocessed dataset allows the model to learn the true patterns in the data, leading to more accurate and trustworthy predictions.

1. Heart Attack Risk Prediction Dataset

To ensure the dataset was ready for modeling and analysis, we utilized several data cleaning and preprocessing steps. First, categorical variables were appropriately encoded. The 'Sex' column was mapped to binary values with 'Female' as 0 and 'Male' as 1, while the 'Diet' column was mapped into three categories: 'Healthy' (0), 'Average' (1), and 'Unhealthy' (2). Blood pressure values were split into systolic (HBP) and diastolic (LBP) pressures, and the 'Diabetes' column was inverted to maintain consistency with other binary features. Continuous variables, such as 'Exercise Hours Per Week,' 'Sedentary Hours Per Day,' 'Income,' and 'BMI,' were rounded to the nearest integer to reduce noise. Non-essential columns, including 'Patient ID,' 'Blood Pressure,' and geographical features as 'Country,' 'Continent' and 'Hemisphere' were dropped to focus on relevant predictors. Categorical variables were first converted into strings and then one-hot encoded, transforming them into binary indicator variables, dropping the first category to avoid multicollinearity. Continuous variables were downcast to smaller data types, such as int16 and float16 to optimize memory usage and computational efficiency. The processed dataset contains 8,763 records and 24 features, which we split into training and test sets with an 80-20 ratio, stratifying by the target variable to ensure balanced class distribution in both sets. This ensured that the data was in an optimal format for modeling and analysis, aiming to achieve the highest precision in predicting heart attacks.

2. Heart Disease Indicators Dataset

This dataset contained 253,661 instances and 22 features, which was significantly large for efficient modeling. To handle this, a stratified sampling technique was employed to reduce the dataset to a manageable size of 50,000 records, taking care of the original distribution of the target variable, 'HeartDiseaseorAttack'. Specifically, a target size of 50,000 was selected and the new dataset was sampled to keep the proportion of the classes which were approximately 90.6% of instances without heart disease or attack and 9.4% with. Then, we employed data cleaning steps to ensure consistency and quality. Duplicate records were removed, reducing the dataset to 48,050 instances. The 'Diabetes' variable, which had three categories (0, 1, 2), was cleaned by removing instances labeled as '1' and merging the '2' category into '1', thus transforming it into a binary feature which represents if someone has diabetes or not. Also, categorical features such as 'HighBP', 'HighChol', 'CholCheck', and others were converted into strings and then one-hot encoded to create binary indicator variables, dropping the first category to prevent multicollinearity. This step expanded the dataset to 47,143 instances and 43 features, ensuring that all categorical data was suitably represented for machine learning algorithms. Finally, for continuous features like 'BMI', 'GenHlth', 'MentHlth' and 'PhysHlth' the data types were kept in their original integer format to maintain accuracy and computational efficiency. Finally, the dataset was split into training and test sets with an 80-20 ratio, stratifying by the target variable to ensure that the class distribution kept the same across both of the sets.

6. Results

In this section we will represent the findings and our results from the analysis of the two datasets, divided in three main phases included *Phase 1*: Machine Learning algorithms, *Phase 2*: application of Imbalanced Data Methods and *Phase 3*: combination of the best Imbalanced Method with Feature Selecting Techniques and *Phase 4*: Tuning the best 4 models, aimed at predicting heart attack risk, with a primary focus on optimizing precision. For the first dataset 'Heart Attack Risk Prediction Dataset', we applied both Standard Scaling and Robust Scaling as we identified some extreme values and skewedness in the data. For the second dataset 'The Heart Disease Indicators Dataset' we applied only Standard Scaling. Then various Imbalanced Data Methods, Feature Selecting techniques and Machine Learning models were evaluated and the results focused on Precision, Accuracy, and AUC-score.

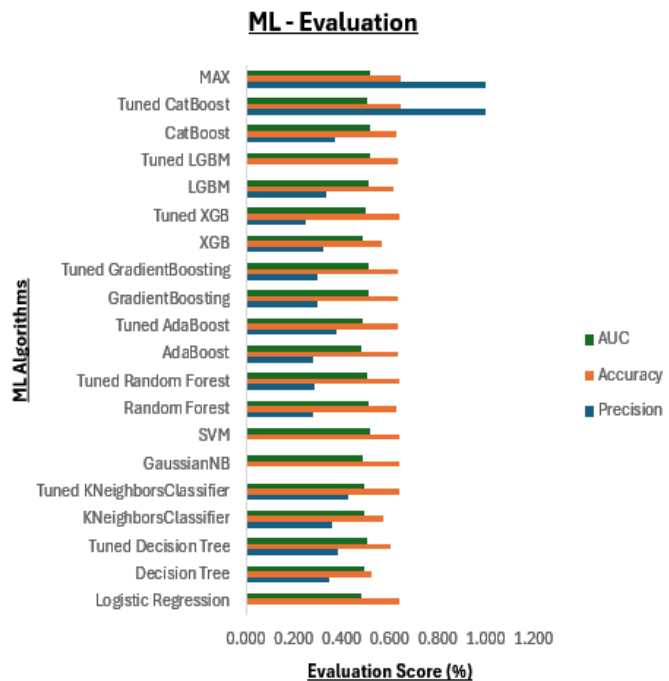
- ***Phase 1***

In phase 1 we applied various Machine Learning algorithms such as Logistic Regression (LR), Decision Tree Classifier (DT), KNeighborsClassifier (KNN), Naïve Bayes (NB), Support Vector Machine (SVM), Random Forest (RF), Adaptive Boosting (AdaBoost), Gradient Boosting, Extreme Gradient Boosting (XGBoost), Light Gradient Boosting Machine (LGBM) and Categorical Boosting (CatBoost) to understand how those algorithms behave with our dataset.

1. Heart Attack Risk Prediction Dataset

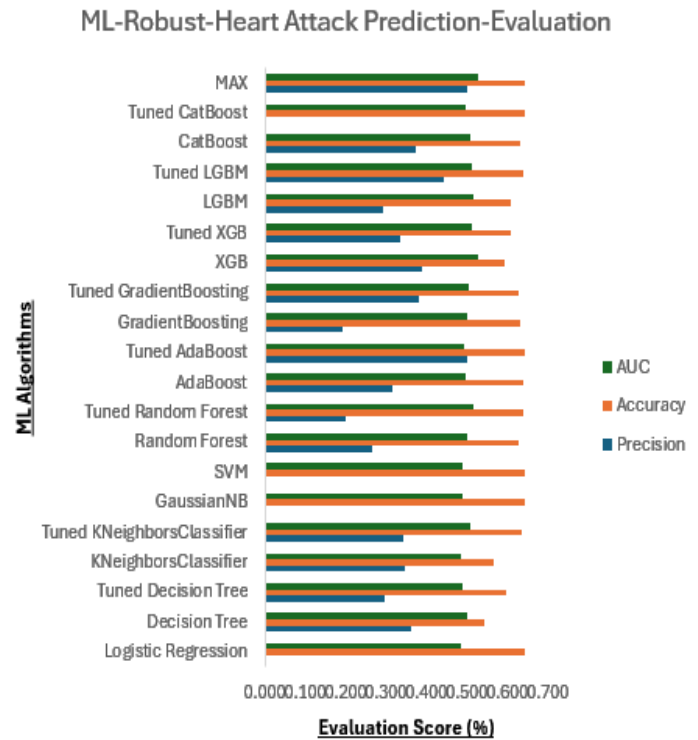
In our initial try at applying various Machine Learning models to the 'Heart Attack Risk Prediction' dataset using Standard Scaling method, show us a wide range of performance metrics. More specifically, the Tuned CatBoost model has a precision score of 1.000, suggesting to us that it made no false positive predictions. However, this precision score in combination with accuracy 0.642 and AUC of 0.506, indicates a potential issue with the model's overall performance. The combination of these metrics shows us that the model is biased about one class (usually the negative class, minority), making very few positive predictions. Those results with perfect precision but poor overall performance suggest that the model's ability to differentiate between classes is inadequate. On the other hand, models like Tuned KNeighborsClassifier and Tuned LGBM give us more balanced performances, with precision scores of 0.426 and 0.335 in combination with higher accuracy and AUC scores. These models, even if they did not have very good outputs, are making more balanced predictions across both classes, which is crucial for predicting heart attacks. Also, SVM model, with accuracy 0.642 and AUC 0.514, has a precision of 0.000. This outcome suggests that the SVM model is capable of distinguishing the classes but is not making any positive predictions. The overall performance across these models highlights the trade-offs in predictive modeling in imbalanced datasets like this one.

Heart Attack Prediction			
Model	Precision	Accuracy	AUC
Logistic Regression	0.000	0.642	0.479
Decision Tree	0.348	0.525	0.491
Tuned Decision Tree	0.382	0.604	0.504
KNeighborsClassifier	0.362	0.574	0.491
Tuned KNeighborsClassifier	0.426	0.637	0.494
GaussianNB	0.000	0.642	0.488
SVM	0.000	0.642	0.514
Random Forest	0.277	0.630	0.509
Tuned Random Forest	0.286	0.640	0.503
AdaBoost	0.280	0.635	0.480
Tuned AdaBoost	0.375	0.636	0.489
GradientBoosting	0.300	0.635	0.509
Tuned GradientBoosting	0.300	0.635	0.509
XGB	0.323	0.568	0.486
Tuned XGB	0.250	0.639	0.498
LGBM	0.335	0.613	0.509
Tuned LGBM	0.000	0.636	0.519
CatBoost	0.370	0.627	0.517
Tuned CatBoost	1.000	0.642	0.506
MAX	1.000	0.642	0.519



In our second attempt we used the Robust Scaler, which is very effective for handling data with outliers. The Tuned AdaBoost model has the highest precision of all the models, with precision 0.5, which indicates that it correctly predicted half of the positive cases. However, this high precision in combination with an accuracy of 0.642 and an AUC of 0.494, shows us that the model is highly correct in its positive predictions but its overall ability to distinguish between classes is limited, as evidenced by the lower AUC. However, the XGB model also performs well with a precision of 0.389 and the highest AUC of 0.526 among all models, showing a better balance between precision and overall classification performance. This suggests that XGB is more effective in distinguishing between positive and negative cases, making it a strong candidate despite its lower precision compared to Tuned AdaBoost. Also, models like Tuned CatBoost achieve a precision of 0.000, similar to Logistic Regression and GaussianNB, indicating no positive predictions were made. These results highlight the challenge of balancing precision with other performance metrics, especially when using scaling methods like Robust Scaler. The overall performance of these models suggests that models like Tuned AdaBoost can be very good for precision metric and others like XGB can give us a more balanced approach, making them more reliable for predicting heart attacks.

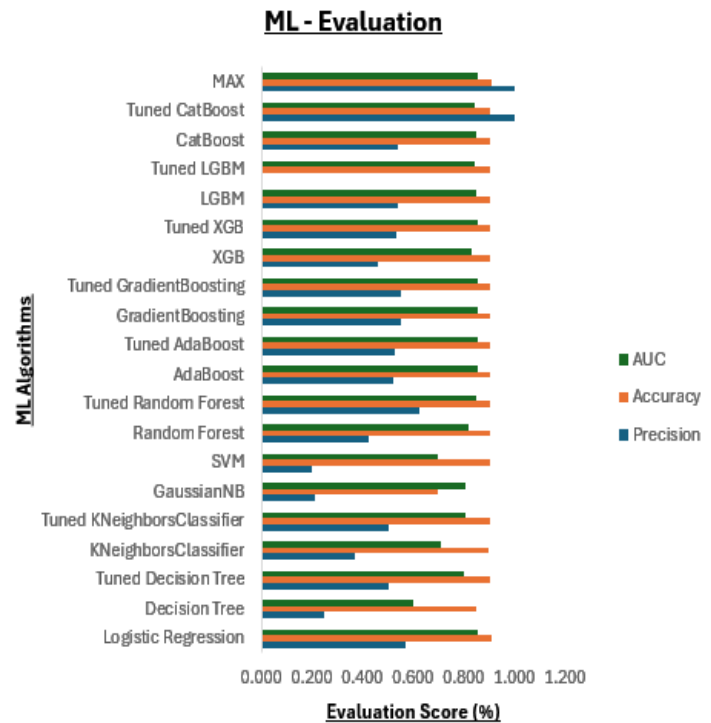
Heart Attack Prediction - Robust			
Model	Precision	Accuracy	AUC
Logistic Regression	0.000	0.642	0.483
Decision Tree	0.361	0.542	0.502
Tuned Decision Tree	0.298	0.594	0.489
KNeighborsClassifier	0.347	0.565	0.483
Tuned KNeighborsClassifier	0.340	0.633	0.507
GaussianNB	0.000	0.642	0.489
SVM	0.000	0.641	0.489
Random Forest	0.264	0.627	0.498
Tuned Random Forest	0.200	0.640	0.516
AdaBoost	0.316	0.638	0.498
Tuned AdaBoost	0.500	0.642	0.494
GradientBoosting	0.194	0.631	0.498
Tuned GradientBoosting	0.379	0.629	0.503
XGB	0.389	0.590	0.526
Tuned XGB	0.335	0.606	0.513
LGBM	0.293	0.609	0.514
Tuned LGBM	0.443	0.637	0.513
CatBoost	0.372	0.629	0.509
Tuned CatBoost	0.000	0.642	0.494
MAX	0.500	0.642	0.526



2. Heart Disease Indicators Dataset

For the analysis of the second dataset 'Heart Disease Health Indicators', we applied various machine learning models, and we evaluated them using Standard Scaling. The Logistic Regression model achieves precision of 0.571, accuracy of 0.907, and AUC of 0.855, showing us a strong balance between precision and overall classification performance, with high accuracy and excellent class differentiation ability. Tuned Random Forest and Tuned AdaBoost models both perform very well, with Tuned Random Forest to has a precision of 0.622 and Tuned AdaBoost 0.526. Both models show us high accuracy at 0.905 and 0.904, and very good AUC scores of 0.850 and 0.853, suggesting that these models are highly effective at predicting heart disease while keeping good the overall performance. However, the Tuned CatBoost model has a precision of 1.000, but with an accuracy of 0.904 and an AUC of 0.840. This high precision gives us to understand that the model perfectly predicted positive cases, but the slightly lower AUC and accuracy indicates the issues with class balance. The AUC score explains to us that even if the model is very good at predicting the positive class, it may be less effective at distinguishing between classes overall. GradientBoosting and Tuned GradientBoosting models also shows balanced performance with precision of 0.550, accuracy of 0.905, and AUC of 0.855, making them reliable choices for heart disease prediction. Overall, Logistic Regression and Tuned Random Forest have generally a strong performance.

Heart Disease Health Indicators			
Model	Precision	Accuracy	AUC
Logistic Regression	0.571	0.907	0.855
Decision Tree	0.246	0.846	0.597
Tuned Decision Tree	0.500	0.903	0.799
KNeighborsClassifier	0.367	0.895	0.707
Tuned KNeighborsClassifier	0.500	0.903	0.809
GaussianNB	0.211	0.694	0.806
SVM	0.200	0.902	0.697
Random Forest	0.426	0.901	0.821
Tuned Random Forest	0.622	0.904	0.850
AdaBoost	0.523	0.905	0.853
Tuned AdaBoost	0.526	0.905	0.853
GradientBoosting	0.550	0.905	0.855
Tuned GradientBoosting	0.550	0.905	0.855
XGB	0.457	0.901	0.833
Tuned XGB	0.533	0.905	0.852
LGBM	0.541	0.905	0.849
Tuned LGBM	0.000	0.903	0.842
CatBoost	0.540	0.905	0.847
Tuned CatBoost	1.000	0.904	0.840
MAX	1.000	0.907	0.855



• Phase 2

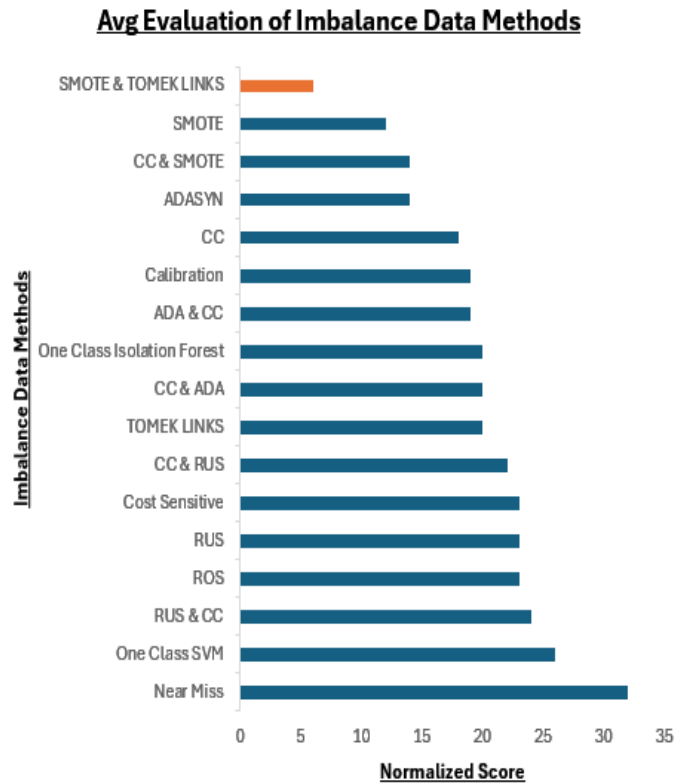
In phase 2, we applied various Imbalanced Data Methods such as Over - Sampling, Under - Sampling, Combination methods, Cost - Sensitive techniques, One - Class techniques and Calibration to our Machine Learning algorithms. Then we averaged the scores for each technique across all evaluation metrics to gain a holistic overview of our dataset.

1. Heart Attack Risk Prediction Dataset

In this phase, we applied various imbalance data methods to the Heart Attack Prediction dataset to estimate their impact on model performance. The table below shows us the evaluation metrics and a combined normalized score, as we wanted to create a fair environment for the ranking of the effectiveness of each technique. The SMOTE & TOMEK LINKS method seems as the most effective technique, with a precision of 0.364, accuracy of 0.597, and AUC of 0.510, with a result to has the highest normalized score (6). This shows us that combining SMOTE with Tomek Links not only improves the balance between classes but also increases the model's ability to correctly identify heart attack cases while keeping a strong overall performance. Other techniques such as CC & SMOTE and SMOTE, also performed well, having normalized scores 14 and 12 respectively. These methods reveal a good balance between precision and AUC, indicating their suitability for handling imbalanced data in heart attack prediction. On the other hand, methods like Near Miss and One Class SVM ranked lower, with total normalized scores of 32 and 26. These techniques showed lower precision and accuracy, suggesting they might not be as effective in dealing with the class imbalance present in this dataset. Overall, the ranking highlights the importance of choosing the right imbalance technique, as it significantly impacts the model's predictive power. The results suggest that SMOTE combined with Tomek Links is the most reliable approach for achieving a balanced and accurate heart attack prediction model in this context.

Heart Attack Prediction - Imbalance Methods

Technique	Precision	Accuracy	AUC	Total Normalized Score
Near Miss	0.353	0.5	0.5	32
One Class SVM	0.308	0.616	0.5	26
RUS & CC	0.355	0.496	0.5	24
ROS	0.343	0.546	0.5	23
RUS	0.356	0.494	0.5	23
Cost Sensitive	0.319	0.538	0.51	23
CC & RUS	0.362	0.454	0.5	22
TOMEK LINKS	0.3	0.606	0.5	20
CC & ADA	0.365	0.46	0.5	20
One Class Isolation Forest	0.281	0.621	0.5	20
ADA & CC	0.338	0.573	0.5	19
Calibration	0.128	0.641	0.51	19
CC	0.365	0.459	0.51	18
ADASYN	0.349	0.584	0.5	14
CC & SMOTE	0.367	0.459	0.51	14
SMOTE	0.344	0.585	0.51	12
SMOTE & TOMEK LINKS	0.364	0.597	0.51	6

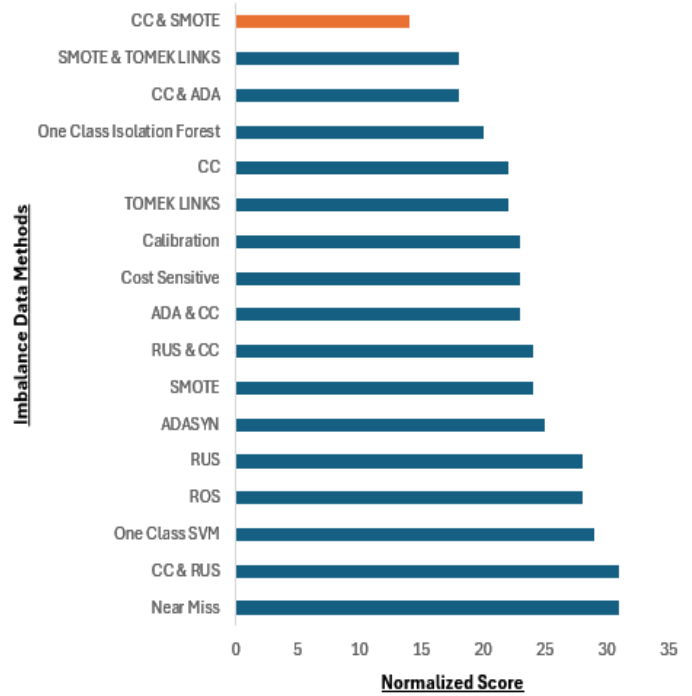


For the second scale method (Robust) that we used, the results are provided in the table below. The combination of CC & SMOTE seems to have the highest performance with a total normalized score of 14, show us its effectiveness in balancing the dataset and how it can improve the model's precision, accuracy, and AUC. With a precision of 0.367, this method can give us some promises for the correct identification of heart attacks. Other techniques such as SMOTE & TOMEK LINKS and CC & ADA also performed well, as they have normalized scores of 18. These methods demonstrated a good balance between the evaluation metrics, making them suitable alternatives for handling imbalanced data. On the other side, techniques like Near Miss and CC & RUS ranked lower, with a total normalized score of 31. These methods struggled to achieve high precision and accuracy, suggesting that they may not be as effective in addressing the class imbalance in this dataset when using RobustScaler. The results highlight the importance of selecting the appropriate data imbalance technique in combination with RobustScaler. The CC & SMOTE technique stands out as the most reliable approach for achieving a balanced and accurate heart attack prediction model in this scenario.

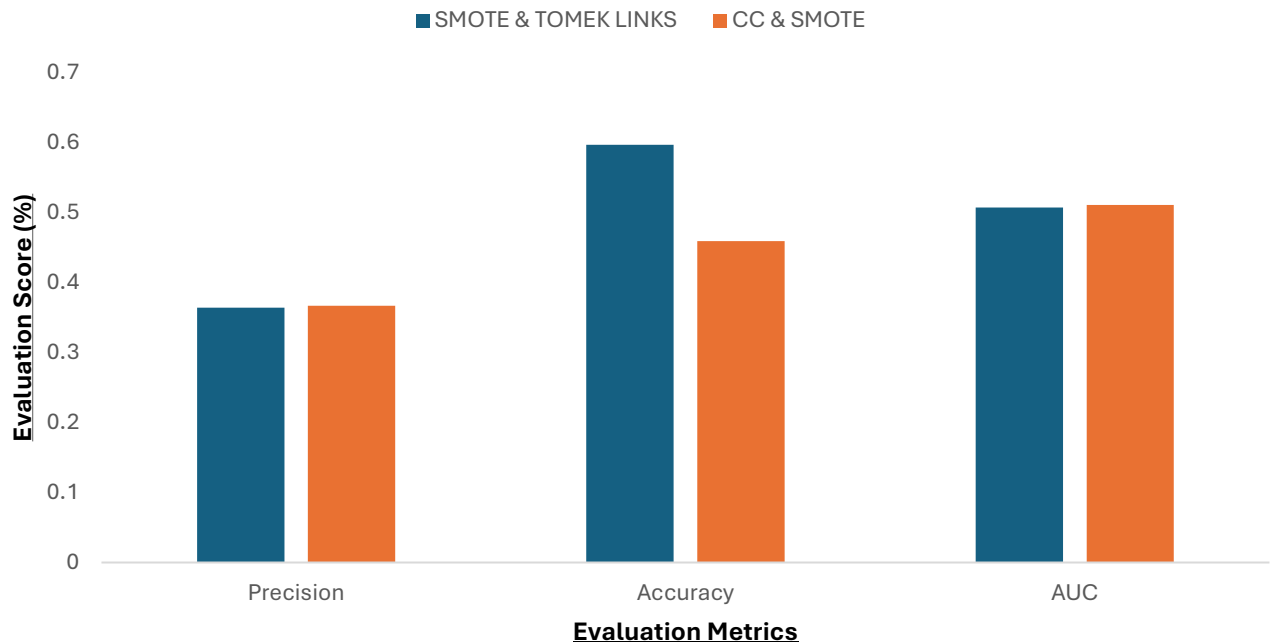
Heart Attack Prediction - Imbalance Methods - RobustScaler

Technique	Precision	Accuracy	AUC	Total Normalized Score
Near Miss	0.359	0.507	0.5	31
CC & RUS	0.359	0.441	0.5	31
One Class SVM	0.269	0.625	0.5	29
ROS	0.355	0.546	0.5	28
RUS	0.358	0.502	0.5	28
ADASYN	0.33	0.589	0.5	25
SMOTE	0.355	0.584	0.5	24
RUS & CC	0.361	0.505	0.5	24
ADA & CC	0.362	0.577	0.5	23
Cost Sensitive	0.355	0.534	0.51	23
Calibration	0.158	0.641	0.51	23
TOMEK LINKS	0.349	0.607	0.5	22
CC	0.364	0.447	0.51	22
One Class Isolation Forest	0.277	0.622	0.5	20
CC & ADA	0.362	0.443	0.5	18
SMOTE & TOMEK LINKS	0.35	0.596	0.51	18
CC & SMOTE	0.367	0.459	0.51	14

Avg Evaluation of Imbalance Data Methods



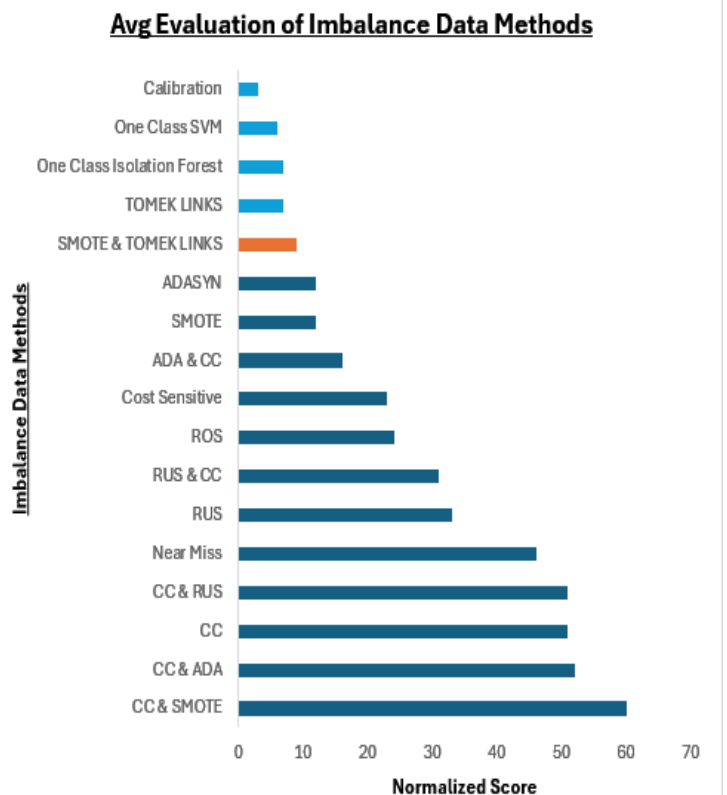
SMOTE & TOMEK LINKS (SMOTETomek) VS CC & SMOTE



2. Heart Disease Indicators Dataset

In the Heart Disease Health Indicators dataset, among the techniques, SMOTE & TOMEK LINKS appears as the most balanced and effective method, achieving a total normalized score of 9. This method demonstrated a good balance across precision (0.375), accuracy (0.860), and AUC (0.795), making it a reliable choice for handling imbalanced data. Although other techniques like TOMEK LINKS, One Class Isolation Forest, One Class SVM, and Calibration showed high scores, they often resulted in a precision of 1.00, which indicates potential issues with overfitting or class imbalance that can skew the model's performance. Also, methods like CC & SMOTE, CC & ADA, and CC & RUS ranked lower, with normalized scores ranging from 51 to 60. These techniques struggled to balance the trade-offs between precision, accuracy, and AUC. Overall, while SMOTE & TOMEK LINKS did not achieve the highest scores in individual metrics, its consistent performance across all metrics makes it the preferred choice for this dataset. This method provides a stable base for further model development and testing, ensuring a more reliable prediction of heart disease.

Heart Disease Health Indicators - Imbalance Methods				
Technique	Precision	Accuracy	AUC	Total Normilzed Score
CC & SMOTE	0.122	0.287	0.64	60
CC & ADA	0.121	0.286	0.65	52
CC	0.122	0.292	0.65	51
CC & RUS	0.122	0.29	0.66	51
Near Miss	0.129	0.433	0.67	46
RUS	0.224	0.697	0.82	33
RUS & CC	0.224	0.698	0.81	31
ROS	0.258	0.805	0.77	24
Cost Sensitive	0.321	0.849	0.78	23
ADA & CC	0.382	0.859	0.79	16
SMOTE	0.373	0.86	0.8	12
ADASYN	0.375	0.858	0.8	12
SMOTE & TOMEK LINKS	0.375	0.86	0.8	9
TOMEK LINKS	0.477	0.888	0.81	7
One Class Isolation Forest	0.478	0.89	0.81	7
One Class SVM	0.507	0.889	0.81	6
Calibration	0.495	0.905	0.82	3



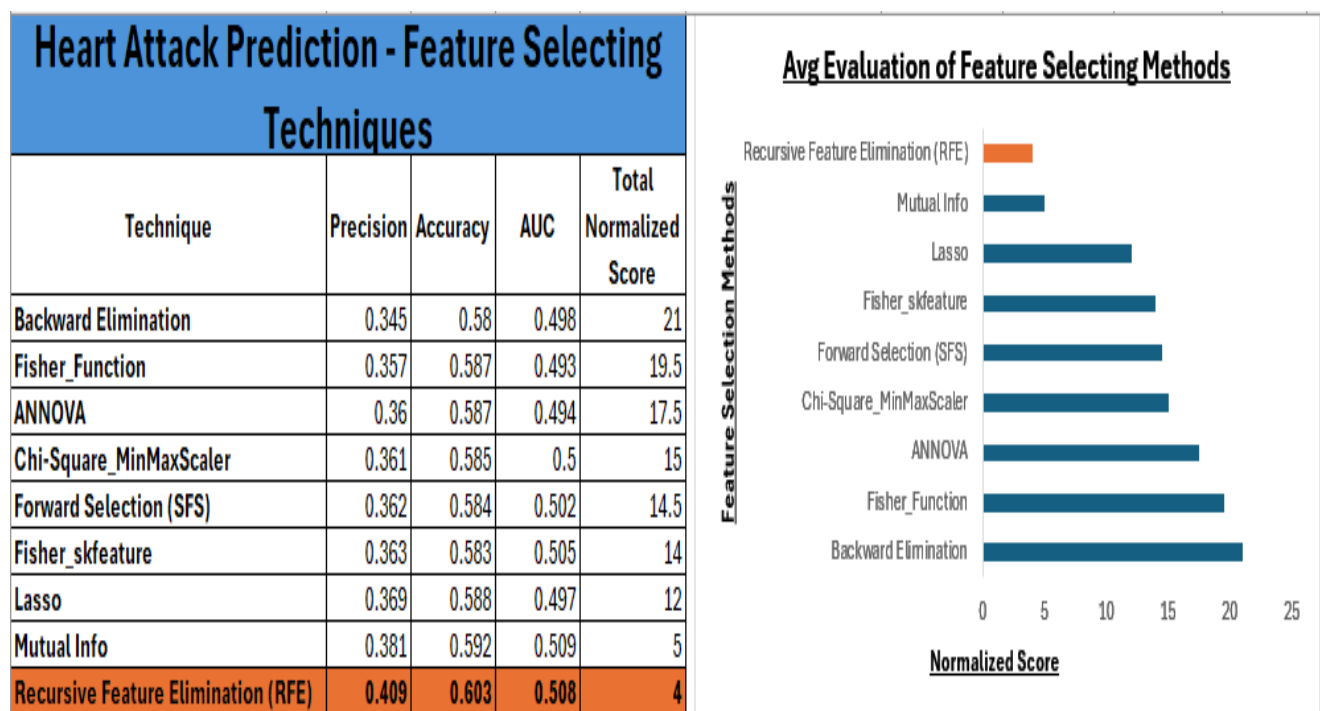
• Phase 3

In Phase 3, we address various Selecting Methods such as Filter, Wrapper and Embedded combined the most effective imbalance method, which we have found from the phase 2 to our Machine Learning algorithms.

1. Heart Attack Risk Prediction Dataset

In our first dataset with Standard Scale method, we applied various feature selection techniques following the implementation of the SMOTETomek imbalance method. The table below presents the performance

of each technique, ranked according to their total normalized score, ensuring a fair comparison across the metrics. Between the tested techniques, Recursive Feature Elimination (RFE) presents as the top performer, achieving a total normalized score of 4. RFE has the highest precision (0.409), which is crucial in heart attack prediction, where false positives can have serious consequences. Additionally, RFE achieved accuracy (0.603) and AUC (0.508), making it the most balanced technique across all metrics. Also, Mutual Info with a total normalized score of 5, showing strong performance in both precision (0.381) and AUC (0.509), though slightly lower accuracy (0.592). This makes Mutual Info a viable option, particularly for models where AUC is a priority. On the other hand, techniques like Backward Elimination and Fisher Function ranked lower, with normalized scores of 21 and 19.5. These techniques reveal lower precision and AUC scores, indicating that they may not be as effective for this specific dataset and prediction task. Overall, RFE stands out as the most effective feature selection technique in this dataset, providing a solid base for building accurate and reliable heart attack prediction models. This technique's ability to increase precision while keeping the balance between accuracy and AUC makes it a preferred choice for further model development.

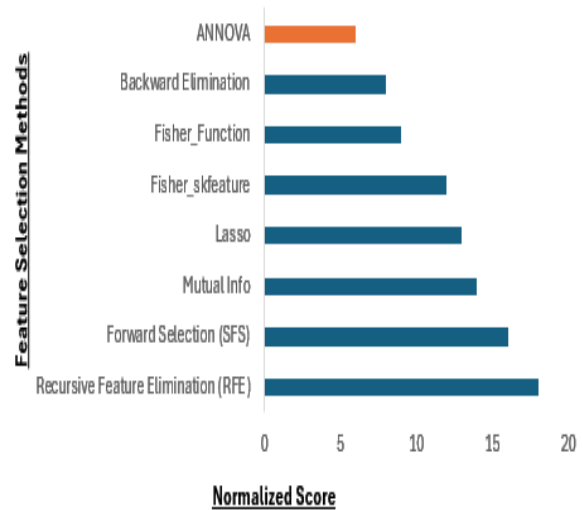


In the first dataset with Robust Scale method, we applied the feature selection techniques following the implementation of the CC & SMOTE imbalance method which was the most effective methods from phase 2. The table below shows us that the ANNOVA technique appears as the top feature selection method with a total normalized score of 6. ANNOVA achieved a precision of 0.362, accuracy of 0.451, and an AUC of 0.505. This indicates that ANNOVA effectively balances the trade-off between these three critical metrics, making it the strongest choice when the dataset is scaled using the Robust Scaler. Backward Elimination and Fisher Function are following behind, with normalized scores of 8 and 9. These techniques also performed well in terms of accuracy and AUC, but they have lower precision compared to ANNOVA. On the other side, techniques like Recursive Feature Elimination (RFE) and Forward Selection (SFS), which previously showed us a strong performance with standard scaling, ranked lower in this scenario with normalized scores of 18 and 16. Generally, ANNOVA stands out as the most effective technique for improving heart attack prediction models in a dataset scaled with the Robust Scaler, offering a solid balance across all evaluated metrics.

Heart Attack Prediction - Feature Selecting Techniques - RobustScaler

Technique	Precision	Accuracy	AUC	Total Normalized Score
Recursive Feature Elimination (RFE)	0.36	0.428	0.5	18
Forward Selection (SFS)	0.36	0.435	0.501	16
Mutual Info	0.359	0.442	0.503	14
Lasso	0.324	0.438	0.514	13
Fisher_skfeature	0.361	0.453	0.491	12
Fisher_Function	0.362	0.45	0.503	9
Backward Elimination	0.365	0.438	0.505	8
ANNOVA	0.362	0.451	0.505	6

Avg Evaluation of Feature Selecting Methods



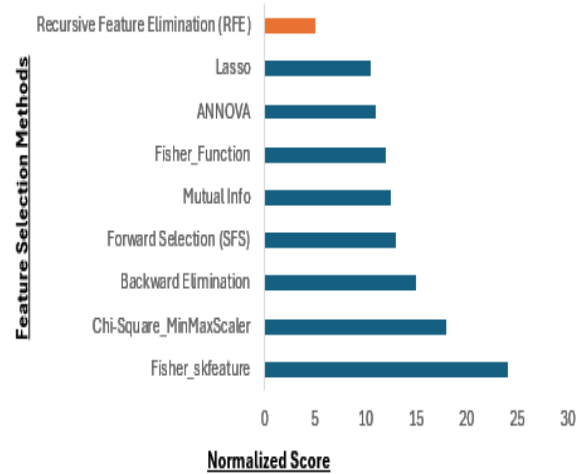
2. Heart Disease Indicators Dataset

In the second dataset, we followed the same process only with Standard scale. The table below ranks these techniques according to their effectiveness, normalized to ensure a fair comparison. Recursive Feature Elimination (RFE) presents as the best-performing technique, achieving the highest total normalized score of 5. RFE shows a higher precision (0.399), accuracy (0.883), and AUC (0.787), indicating its robust capability in identifying the most relevant features for heart disease prediction. Lasso and ANNOVA are behind, with normalized scores of 10.5 and 11 respectively. Both techniques give us strong results across all metrics, particularly in AUC, which is crucial for distinguishing between the positive and negative classes. Lasso, with a precision of 0.375 and an AUC of 0.801, reveals its strength in balancing predictive accuracy and model complexity. Other techniques like Fisher Function and Mutual Info also performed well, with lower normalized scores. These methods offered a good balance between precision and accuracy, making them viable alternatives depending on the specific model requirements. Finally, Fisher_skfeature ranked last with a total normalized score of 24, indicating that it was less effective compared to other techniques for this dataset. Overall, Recursive Feature Elimination (RFE) was shown as the most effective feature selection technique for this dataset, especially when high precision is critical for heart disease prediction.

Heart Disease Health Indicators - Feature Selecting Techniques

Technique	Precision	Accuracy	AUC	Total Normalized Score
Fisher_skfeature	0.363	0.583	0.505	24
Chi-Square_MinMaxScaler	0.367	0.868	0.773	18
Backward Elimination	0.366	0.866	0.793	15
Forward Selection (SFS)	0.369	0.869	0.789	13
Mutual Info	0.375	0.871	0.774	12.5
Fisher_Function	0.371	0.875	0.779	12
ANNOVA	0.372	0.875	0.779	11
Lasso	0.375	0.856	0.801	10.5
Recursive Feature Elimination (RFE)	0.399	0.883	0.787	5

Avg Evaluation of Feature Selecting Methods



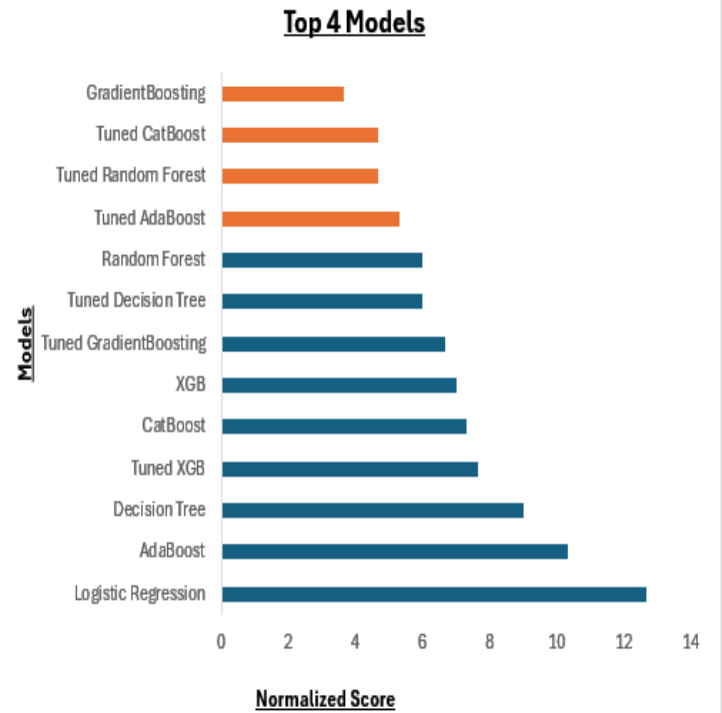
• Phase 4

In the last phase, we will present the final results derived from our comprehensive process of model selection and optimization. Focusing on heart attack prediction, we have identified the top four models by combining the best imbalance handling method from phase 2 with the most effective feature selection technique from phase 3. These models were further fine-tuned with a specific emphasis on precision, aiming to identify the most suitable and reliable model for our dataset.

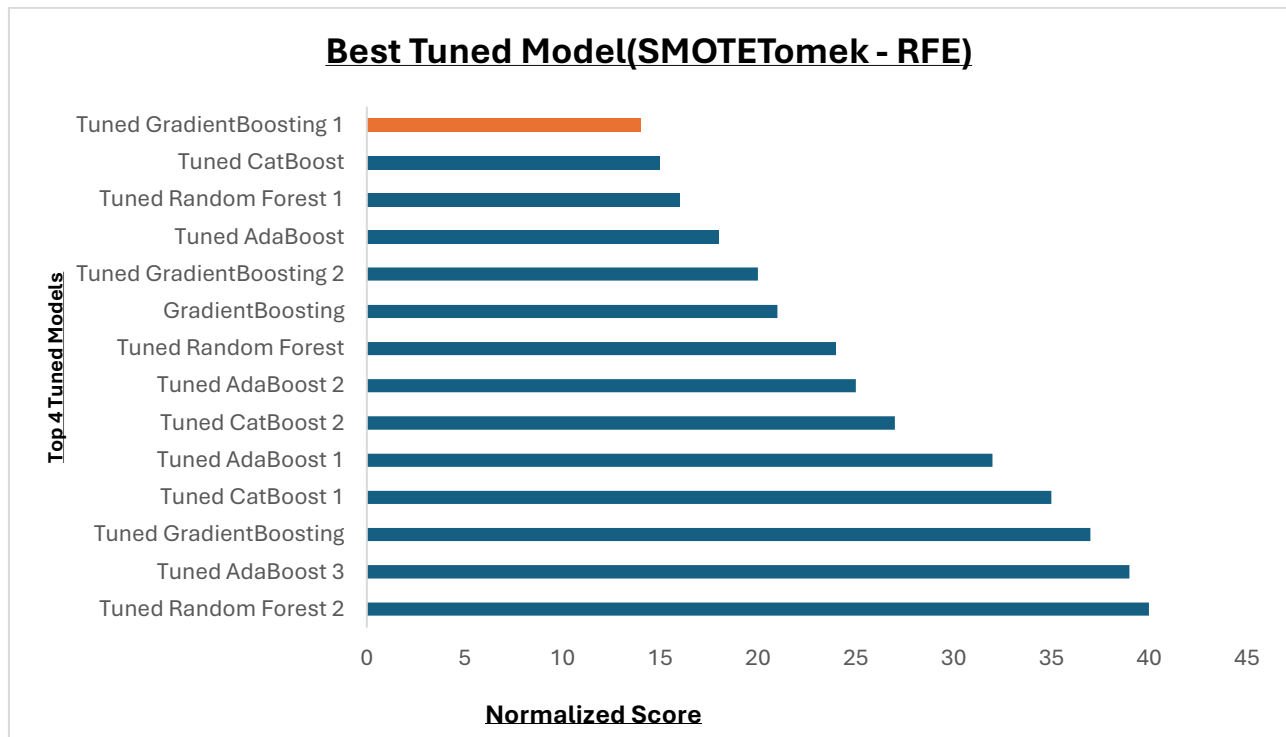
1. Heart Attack Risk Prediction Dataset

The table below presents the analysis of various models applied to the 'Heart Attack Prediction' dataset using a combination of SMOTETomek and Recursive Feature Elimination (RFE) with Standard scaling. Our findings show that the GradientBoosting model achieved the highest precision (0.412) with an accuracy (0.637) and AUC (0.513), making it the top-performing model according to the total normalized score (3.67). The other 2 models which fill our quartet are CatBoost and Random Forest where both showing strong performance across the metrics. Finally, the AdaBoost, which also performed well with a precision of 0.400 and accuracy of 0.639, but with lower AUC completes our final quarter. These top four models, GradientBoosting, CatBoost, Random Forest, and AdaBoost, were selected for further tuning based on their overall balanced performance and focusing on precision, which is crucial in heart attack prediction to minimize false negatives.

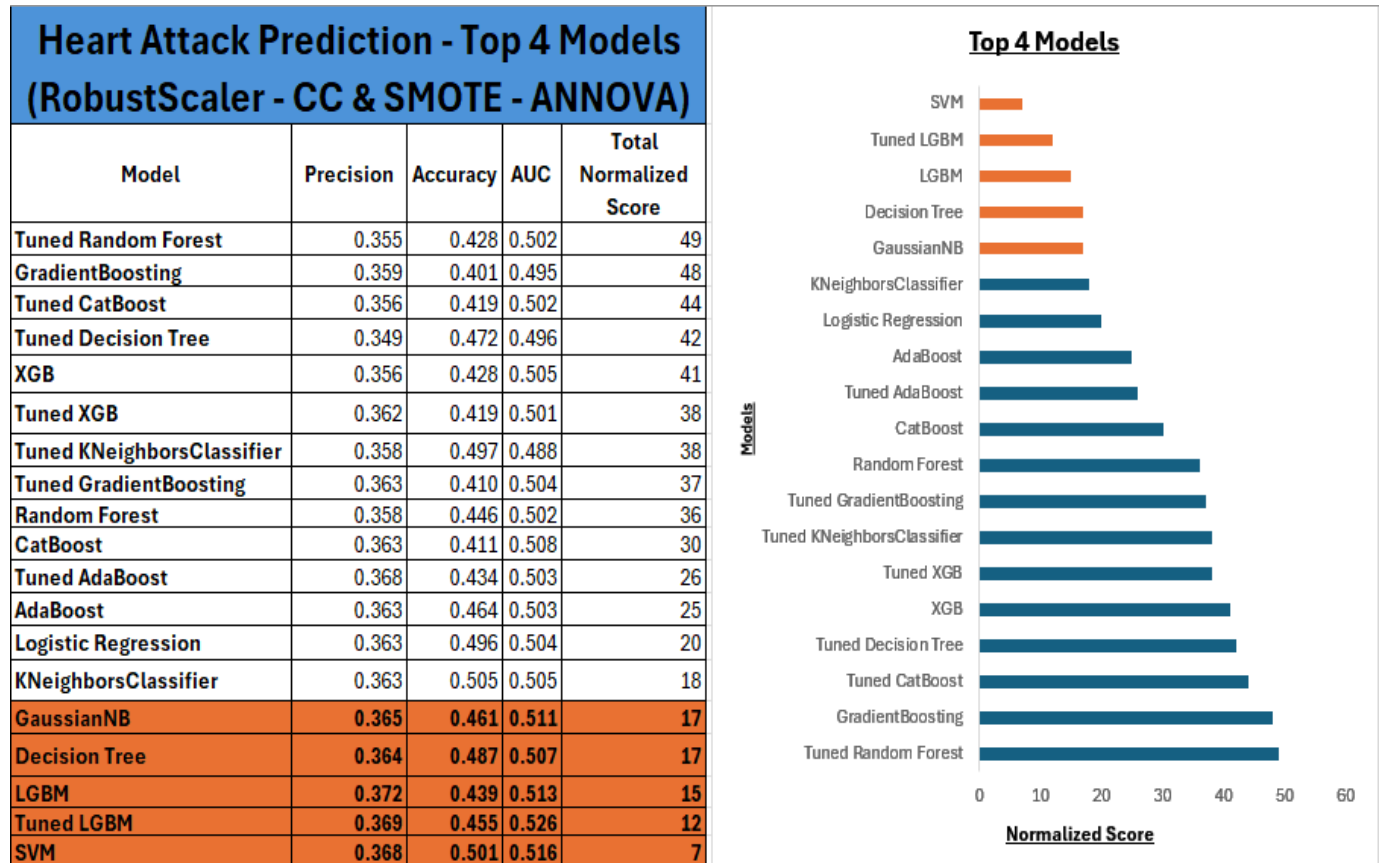
Heart Attack Prediction - Top 4 Models (SMOTETomek - RFE)				
Model	Precision	Accuracy	AUC	Total Normalized Score
Logistic Regression	0.349	0.491	0.482	12.67
AdaBoost	0.363	0.571	0.501	10.33
Decision Tree	0.368	0.547	0.508	9
Tuned XGB	0.364	0.618	0.506	7.67
CatBoost	0.359	0.617	0.516	7.33
XGB	0.373	0.588	0.514	7
Tuned GradientBoosting	0.326	0.624	0.516	6.67
Tuned Decision Tree	0.389	0.621	0.506	6
Random Forest	0.371	0.597	0.526	6
Tuned AdaBoost	0.400	0.639	0.494	5.33
Tuned Random Forest	0.386	0.609	0.526	4.67
Tuned CatBoost	1.000	0.642	0.484	4.67
GradientBoosting	0.412	0.637	0.513	3.67



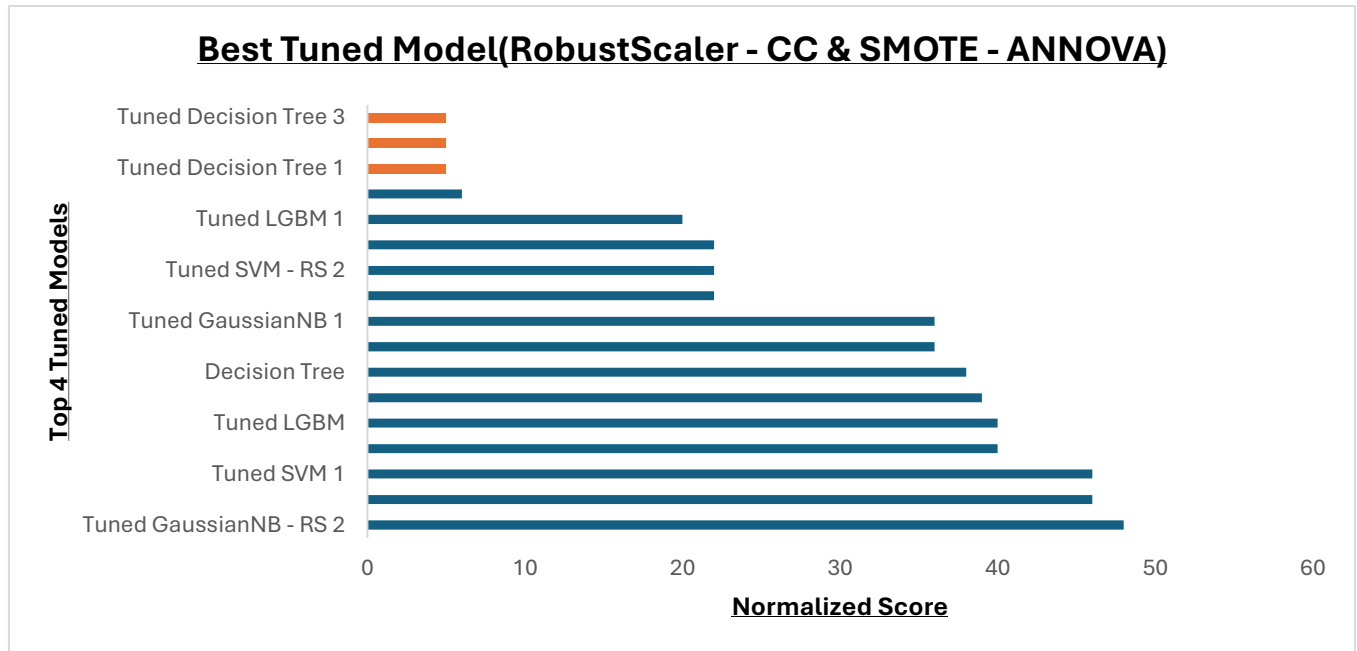
After extensive testing and tuning, the models were ranked based on their precision, accuracy, and AUC scores. Among all the models, Tuned GradientBoosting 1 showed as the top performer, with a precision of 0.391, accuracy of 0.614, and an AUC score of 0.509. This model has been selected as the best candidate for heart attack prediction based on the given evaluation metrics.



For the first dataset with Robust Scale method, the top 4 models which were selected for further analysis using CC & SMOTE and ANNOVA includes SVM, LGBM, Decision Tree, and GaussianNB. The models were evaluated based on their precision, accuracy, AUC, and a total normalized score. The SVM model achieved the highest precision (0.368) combined with accuracy (0.501) and AUC (0.516), making it the best performer according to the total normalized score (7). The LGBM model also performed well, particularly in precision (0.372) and AUC (0.513). As for Decision Tree and GaussianNB both of them show balanced performance across all metrics, indicating their potential for reliable heart attack prediction.



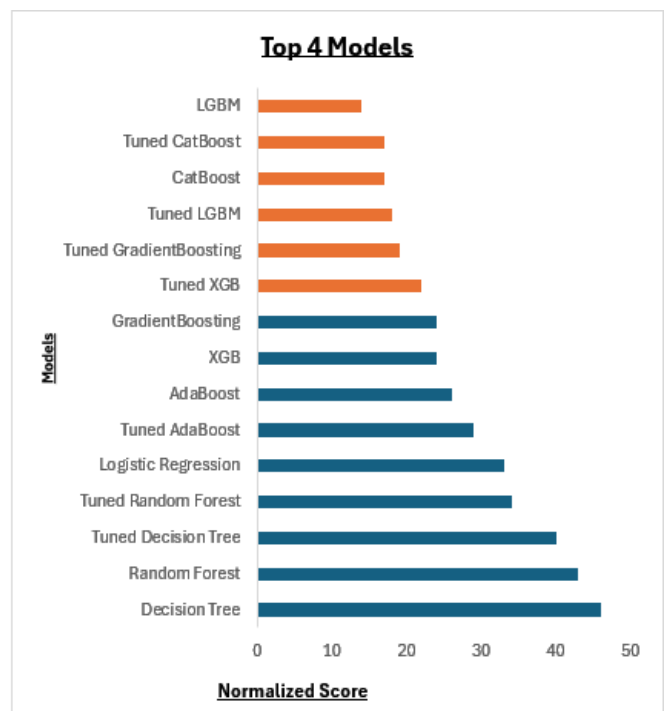
The results for Robust scale methods are slightly different, as between the models which were tested and tuned, Tuned Decision Tree 1, Tuned Decision Tree - RS 2, and Tuned Decision Tree 3 showed as the top performers, all achieving the same precision of 0.371, accuracy of 0.492, and an AUC score of 0.516. The consistency in performance across these variations of the Decision Tree model highlights its robustness in this specific scenario.



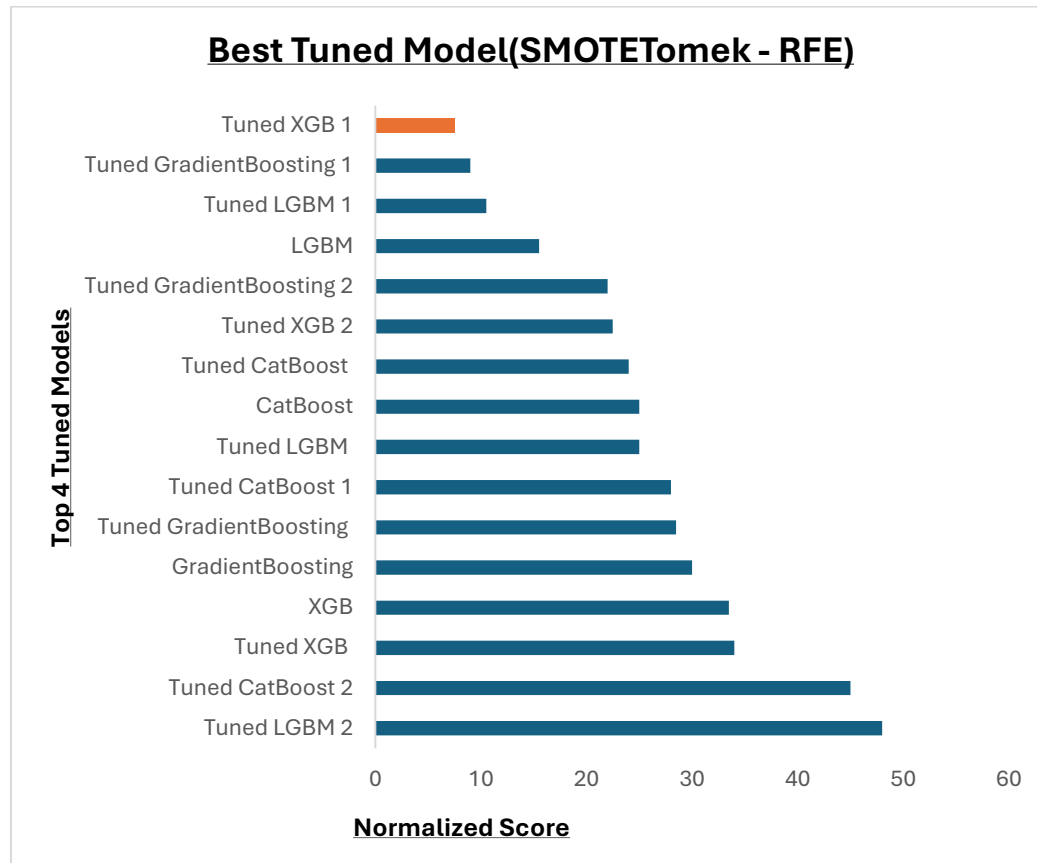
2. Heart Disease Indicators Dataset

In the second dataset, the top 4 models which were selected for further analysis after the application of SMOTETomek with Recursive Feature Elimination (RFE) are LGBM, Catboost, GradientBoosting and XGB. Also, the models are evaluated based on precision, accuracy, AUC, and a total normalized score. The LGBM model, reveals the highest precision (0.510) and accuracy (0.904) with AUC of 0.808, which made it the first candidate for further analysis. Tuned CatBoost also showed strong precision (0.496) and accuracy (0.903), making it the second choice. Furthermore, Tuned GradientBoosting with Tuned XGB have shown very good performance with precision (0.44 and 0.42) and AUC (0.819 and 0.816) make them the last two candidates from our list of four best models.

Heart Disease Health Indicators - Top 4 Models (SMOTETomek - RFE)				
Model	Precision	Accuracy	AUC	Total Normalized Score
Decision Tree	0.262	0.865	0.593	46
Random Forest	0.317	0.883	0.754	43
Tuned Decision Tree	0.320	0.885	0.732	40
Tuned Random Forest	0.366	0.892	0.772	34
Logistic Regression	0.242	0.743	0.838	33
Tuned AdaBoost	0.435	0.898	0.803	29
AdaBoost	0.348	0.870	0.824	26
XGB	0.458	0.902	0.804	24
GradientBoosting	0.395	0.890	0.821	24
Tuned XGB	0.421	0.899	0.816	22
Tuned GradientBoosting	0.440	0.900	0.819	19
Tuned LGBM	0.487	0.903	0.807	18
CatBoost	0.496	0.903	0.802	17
Tuned CatBoost	0.494	0.903	0.806	17
LGBM	0.510	0.904	0.808	14



Finally, for the second dataset the best model is Tuned XGB 1 with a precision of 0.514, accuracy of 0.904, and an AUC score of 0.821, with a total normalized score of 7.5. This model demonstrated a strong balance between precision, accuracy, and AUC, making it the top choice for this dataset.



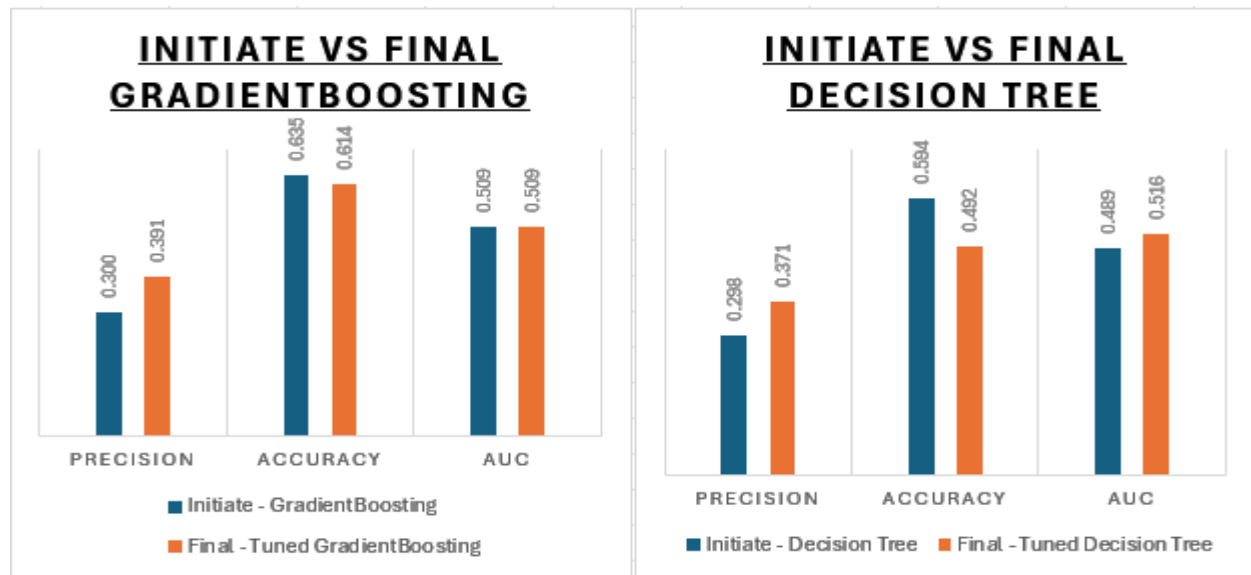
7. Conclusion

In this dissertation, the main objective was to identify the most effective Machine Learning model for heart attack prediction, with a specific emphasis on precision as it is crucial to recognize the potentially life-threatening consequences of false positives. To achieve this goal, we used two distinct datasets: the 'Heart Attack Prediction' and the 'Heart Disease Health Indicators'. Our methodology was structured around some key phases: data preprocessing, the application of various Imbalance methods, Feature selection and finally, an extensive model tuning to optimize the performance. For the 'Heart Attack Prediction' dataset, we employed two scaling methods: Standard Scaling and Robust Scaling. The decision to use both scaling methods was driven by the presence of skewed data with extreme values, which could potentially impact model performance. The process began with the application of various Machine Learning algorithms to different data Imbalance methods. These methods were evaluated based on their ability to handle the innate imbalance in the datasets, which is a common challenge in medical data. After applying these methods, we normalized and ranked the results to identify the most effective imbalance method.

For the Standard Scale on the first dataset, the SMOTETomek method showed as the best-performing imbalance technique. Then, we applied several Feature selection techniques and found that Recursive Feature Elimination (RFE) was the most effective. The combination of SMOTETomek and RFE led to

GradientBoosting being identified as the top-performing model. Initially, this model had a precision of 0.3, accuracy of 0.635, and an AUC of 0.509. After tuning, the model's precision improved to 0.391, with accuracy and AUC scores remaining consistent at 0.614 and 0.509. The top 10 features that contributed to this performance included 'Cholesterol', 'Obesity', 'Alcohol Consumption', 'Diet', 'Stress Level', 'Sedentary Hours Per Day', 'Income', 'BMI', 'Physical Activity Days Per Week', and 'Sleep Hours Per Day'. These features highlight key health and lifestyle factors that are critical in predicting heart attack risk.

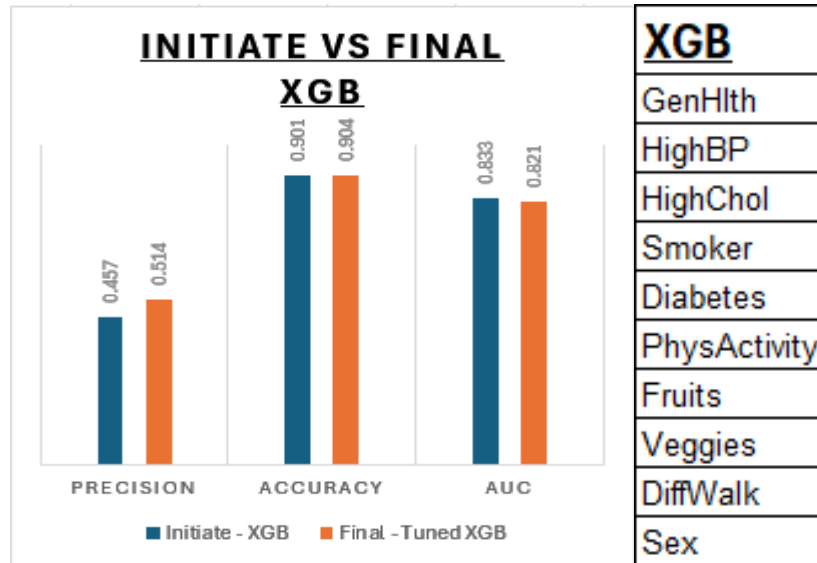
For the same dataset but with Robust Scaling, the results appeared different with CC & SMOTE was the best Imbalance method, and ANNOVA was the most effective Feature Selection technique. The tuned Decision Tree model was the best model, improving from an initial precision of 0.298, accuracy of 0.594, and AUC of 0.489 to a final precision of 0.371, accuracy of 0.492, and AUC of 0.516. The top features for this case were 'Age', 'Sex', 'Cholesterol', 'Heart Rate', 'Diabetes', 'Smoking', 'Stress Level', 'Physical Activity Days Per Week', 'Sleep Hours Per Day', and 'HBP'.



<u>GradientBoosting</u>	<u>Decision tree</u>
Cholesterol	Age
Obesity	Sex
Alcohol Consumption	Cholesterol
Diet	Heart Rate
Stress Level	Diabetes
Sedentary Hours/Day	Smoking
Income	Stress Level
BMI	Physical Activity Days/Week
Physical Activity Days/Week	Sleep Hours/Day
Sleep Hours/Day	HBP

Finally, for the 'Heart Disease Health Indicators' dataset, we were only used the Standard Scaling method. Here, SMOTETomek and RFE again proved to be the most effective combination of Imbalance method with

the Feature selection technique. The XGB model, initially with a precision of 0.457, accuracy of 0.901, and AUC of 0.833, was identified as the best model after tuning. The model's final performance showed a precision of 0.514, accuracy of 0.904, and an AUC of 0.821. The most significant features in this dataset were 'GenHlth', 'HighBP', 'HighChol', 'Smoker', 'Diabetes', 'PhysActivity', 'Fruits', 'Veggies', 'DiffWalk', and 'Sex'.



Overall, the GradientBoosting model, especially in the context of the Heart Attack Prediction dataset with Standard Scaling, showed as the most reliable model as we considered precision and the general performance very critical and important in heart attack prediction. This dissertation demonstrates the importance of carefully selecting and tuning machine learning models, particularly in the context of healthcare, where precision is crucial. The results underscore the effectiveness of combining advanced data preprocessing techniques with strong feature selection and model tuning to improve predictive accuracy in critical applications like heart attack prediction.

8. Problems & Difficulties

During the dissertation, several challenges appeared which impacted the process and results. One of the main issues was the synthetic nature of both datasets, which affected the findings. Synthetic data, while useful for testing, lacks the complexities of real-world data, potentially leading to results that may not fully translate to actual clinical scenarios. Another significant challenge was the large size of the second dataset, which made it necessary to reduce the dataset to manage processing time and computational resources. The extensive size of the datasets also meant that testing various algorithms, imbalance methods, feature selection techniques, and tuning processes was time-consuming and required better computational power. Each step in the process, from data balancing to model tuning, demanded careful consideration and significant time effort.

Overall, while these difficulties made the project more complex and challenging, however, provided valuable insights into managing large-scale data analysis in the context of heart attack prediction, leading to the identification of effective predictive models.

9. Future Work

Future work in the field of heart attack prediction can expand upon the findings of this dissertation by applying these models in clinical settings. Conducting prospective studies in real-world environments would allow the assessment of the models and the impact on the decision-making providing valuable insights. Moreover, the methodologies applied here can be adapted to predict a wide range of other diseases and disorders by incorporating the unique features associated with each condition. This adaptability would make the models more flexible and valuable across different areas of healthcare. Lastly, implementing this predictive information into the heart attack risk prediction model could provide crucial support to the doctors, helping them in clinical decision-making and potentially improving patient outcomes by offering more accurate and timely predictions.

10. References

- [1] Jason Brownlee et al., " [How to Choose a Feature Selection Method For Machine Learning](#)", on August 20, 2020 in Data Preparation
- [2] Jason Brownlee et al., " [Step-By-Step Framework for Imbalanced Classification Projects](#)". on March 19, 2020 in Imbalanced Classification
- [3] TAKCI, HİDAYET (2018) "Improvement of heart attack prediction by the feature selection methods," Turkish Journal of Electrical Engineering and Computer Sciences: Vol. 26: No. 1, Article 1.
- [4] Hussein Abdullah Jaber, Mortada Sadoun Thabet, Rabab Abdul Hussein Fahd, Dalal Khatib Muhbis, and Alaa Khalaf Hamoud et al., "Heart Attack Prediction Model Based on Feature Selection and Decision Tree Approaches "Vol 13, No. 1, March 2024, pp. 01-08
- [5] M. Dash, H. Liu et al., "Feature Selection for Classification ", Department of Information System & Computer Science, National University of Singapore, Singapore 119260, Received 24 January 1997; revised 3 March 1997; accepted 21 March 1997
- [6] Kaushalya Dissanayake, Md Gapar and Md Johar et al., " Comparative Study on Heart Disease Prediction Using Feature Selection Techniques on Classification Algorithms"
- [7] Anna Karen Garate-Escamila, Amir Hajjam El Hassani, Emmanuel Andres et al., "Classification models for heart disease prediction using feature selection and PCA"
- [8] Adi Purnomo et al 2020 J. Phys.: Conf. Ser. 1511 012001, "Adding feature selection on Naïve Bayes to increase accuracy on classification heart attack disease"
- [9] A. Jović, K. Brkić and N. Bogunović et al., "A review of feature selection methods with applications", MIPRO 2015, 25-29 May 2015, Opatija, Croatia
- [10] Ahmad Ayid, Ahmad and Huseyin Polat "Prediction of Heart Disease Based on Machine Learning Using Jellyfish Optimization Algorithm"
- [11] Dr.D. Ramyachitra and P. Manikandan et al., "IMBALANCED DATASET CLASSIFICATION AND SOLUTIONS: A REVIEW"

- [12] Sotiris Kotsiantis, Dimitris Kanellopoulos and Panayiotis Pintelas et al., "Handling imbalanced datasets: A review"
- [13] MENG WANG, XINGHUA YAO² and YIXIANG CHEN et al., "An Imbalanced-Data Processing Algorithm for the Prediction of Heart Attack in Stroke Patients"
- [14] ABID ISHAQ, SAIMA SADIQ, MUHAMMAD UMER, SALEEM ULLAH, SEYEDALI MIRJALILI, (Senior Member, IEEE), VAIBHAV RUPAPARA AND MICHELE NAPPI, (Senior Member, IEEE) et al., "Improving the Prediction of Heart Failure Patients' Survival Using SMOTE and Effective Data Mining Techniques"
- [15] K Rohit Chowdary et al 2022 J. Phys.: Conf. Ser. 2325 012051, "Early heart disease prediction using ensemble learning techniques"
- [16] Muhammad Waqar, Hassan Dawood, Hussain Dawood, Nadeem Majeed, Ameen Banjar and Riad Alharbey et al., "An Efficient SMOTE-Based Deep Learning Model for Heart Attack Prediction"
- [17] S. Ajmal Mohamed and Mr. Balamurali et al., "Predicting The Heart Attack From Accessible Patients Medical Datasets Using Data Mining Technique"
- [18] I. O. Awoyelu, Y. Egbekunle and O. Ogunlade et al., "PREDICTIVE MODELS FOR HEART ATTACK DISEASE RISK"
- [19] Muhammad Rizwan, Sadia Arshad, Hafsa Aijaz, M. Zeeshan UI Haque and Rizwan Ahmed Khan et al., "Heart Attack Prediction using Machine Learning Approach"
- [20] Janaranjani N, Divya P, Madhukiruba E, Dr. R. Santhosh, R. Reshma, D. Selvapandian et al., "Heart Attack Prediction using Machine Learning"
- [21] Sushmita Manikandan et al., "Heart Attack Prediction System"
- [22] SALMAN, ISSAM (2019) "Heart attack mortality prediction: an application of machine learning methods," Turkish Journal of Electrical Engineering and Computer Sciences: Vol. 27: No. 6, Article 24.
- [23] Mohammad Alshraideh, Najwan Alshraideh, Abedalrahman Alshraideh, Yara Alkayed, Yasmin Al Trabsheh and Bahaaldeen Alshraideh et al., "Enhancing Heart Attack Prediction with Machine Learning: A Study at Jordan University Hospital"
- [24] Mrs. M.G. Chitra and Dr. Ramya Govindaraj et al., "Effective Heart Attack Prediction method using Machine Learning Algorithm"
- [25] Noor Fatima et al., "[Understanding Standardization in Data Preprocessing](#)"
- [26] Jason Brownlee et al., "[How to Scale Data With Outliers for Machine Learning](#)"
- [27] Justin M. Johnson* and Taghi M. Khoshgoftaar et al., "Survey on deep learning with class imbalance"
- [28] [Cory Maklin et al., "Synthetic Minority Over-sampling TEchnique \(SMOTE\)"](#)
- [29] [Adaptive Synthetic Sampling \(ADASYN\)](#)
- [30] Yan-Ping Zhang, Li-Na Zhang and Yong-Cheng Wang et al., "[Cluster-based Majority Under-Sampling Approaches for Class Imbalance Learning](#)"
- [31] NIRAJAN JHA et al., "[Understanding Feature Selection Techniques in Machine Learning](#)"
- [32] Nicholas Pudjihartono, Tayaza Fadason, Andreas W. Kempa-Liehr and Justin M. O'Sullivan et al., "A Review of Feature Selection Methods for Machine Learning-Based Disease Risk Prediction"
- [33] Will Kenton et al., "[What Is Analysis of Variance \(ANOVA\)?](#)"

- [34] [Chi-Square Test of Independence](#)
- [35] Nate Rosidi et al., "[Advanced Feature Selection Techniques for Machine Learning Models](#)"
- [36] Rupak (Bob) Roy – II et al., "[Mutual Information Score — Feature Selection](#)"
- [37] Sole Galli et al., "[Feature Selection with Wrapper Methods in Python](#)"
- [38] [RFE](#) by scikit-learn
- [39] Vikas Verma et al., "[A comprehensive guide to Feature Selection using Wrapper methods in Python](#)"
- [40] "[Least Absolute Shrinkage and Selection Operator \(LASSO\)](#)" by github
- [41] Hariharan N et al., "[Logistic Regression in Machine Learning](#)"
- [42] Daniel Jurafsky & James H. Martin et al., "[Logistic Regression](#)"
- [43] Alaa Sheta, Walaa El-Ashmawi and Abdelkarim Baareh et al., "Heart Disease Diagnosis Using Decision Trees with Feature Selection Method"
- [44] "[What is the k-nearest neighbors \(KNN\) algorithm?](#)" by IBM
- [45] Kashishdafa et al., "[Gaussian Naive Bayes: Understanding the Basics and Applications](#)"
- [46] Anshul Saini et al., "[Guide on Support Vector Machine \(SVM\) Algorithm](#)"
- [47] Sruthi E R et al., "[Understanding Random Forest Algorithm With Examples](#)"
- [48] Segun Akinola, Reddy Leelakrishna and Vijayakumar Varadarajan et al., "Enhancing cardiovascular disease prediction: A hybrid machine learning approach integrating oversampling and adaptive boosting techniques"
- [49] Azal Ahmad Khan, Omkar Chaudhari and Rohitash Chandra et al., "A review of ensemble learning and data augmentation models for class imbalanced problems: Combination, implementation and evaluation"
- [50] Davide Boldini, Francesca Grisoni, Daniel Kuhn, Lukas Friedrich & Stephan A. Sieber et al., "Practical guidelines for the use of gradient boosting for molecular property prediction"
- [51] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye and Tie-Yan Liu et al., "LightGBM: A Highly Efficient Gradient Boosting Decision Tree"
- [52] Aravind Kolli et al., "[Understanding CatBoost: The Gradient Boosting Algorithm for Categorical Data](#)"
- [53] Rahul Shah et al., "[Tune Hyperparameters with GridSearchCV](#)"
- [54] James Bergstra and Yoshua Bengio et al., "Random Search for Hyper-Parameter Optimization"
- [55] Tyler J. Bradshaw, Zachary Huemann, Junjie Hu and Arman Rahmim et al., "[A Guide to Cross-Validation for Artificial Intelligence in Medical Imaging](#)"
- [56] "[Tuning the hyper-parameters of an estimator](#)" by Scikit-learn

