

Heart Disease Prediction: A Comparative Study of Machine Learning Pipelines

1st Akbar Radjabov 2nd Adrian Fyrpil 3rd Jakob Jack Loula 4th Yoseph Naoom 5th Zakariya Shafe
School of Engineering School of Engineering School of Engineering School of Engineering School of Engineering
Jönköping University Jönköping University Jönköping University Jönköping University Jönköping University
Stockholm, Sweden Jönköping, Sweden Norrköping, Sweden Kristianstad, Sweden Jönköping, Sweden
raak21zu@student.ju.se fyad22vf@student.ju.se loja22vr@student.ju.se nayo22ma@student.ju.se moza22kw@student.ju.se

Abstract—Cardiovascular disease is the leading cause of death globally, creating a pressing need for accurate and efficient computational screening tools. This paper presents a comparative evaluation of four machine learning classifiers Decision Tree, Logistic Regression, XGBoost, and Neural Networks for binary heart disease prediction using the survey dataset from the CDC. We systematically evaluate 200 unique pipeline combinations, investigating isolated and combined effects of four class-imbalance handling strategies and four feature selection approaches across two datasets. Experiments are conducted under controlled conditions relative to an unmodified baseline. Models are evaluated primarily on Recall and F1-Score, along side with ROC-AUC; reflecting the importance of misdiagnosing a patient. Results indicate that XGBoost combined with SMOTE-ENN produced the highest F1-score, which reflects a good balance between precision and sensitivity. However if ranked by recall, applying Logistic Regression with ADASYN returned the best results. The recommended pipeline which is on XGBoost and SMOTE-ENN, achieves 0.557 recall and 0.467 F1-score.

I. INTRODUCTION

Heart disease can be defined as a group of different types of conditions specifically related to the heart [1]. A broader term that is often mentioned when discussing heart disease is Cardiovascular diseases (CVDs) [2]. Cardiovascular disease is an umbrella term describing disorders affecting not only the heart but also blood vessels. Heart disease represents a major subset of CVDs with disorders such as coronary heart disease, rheumatic heart disease and congenital heart disease. According to the Centers for Disease and Control Prevention, the most common type of heart disease in the United States is coronary artery disease (CAD).

According to the World Health Organization (WHO), CVDs are the leading cause of death globally. In addition to this CVDs have been known as one of the most expensive diseases causing burdens on healthcare systems [3] as well as economic burden on entire countries [4]. This burden is expected to grow in the upcoming 25 years primarily because of the effect of an aging population [5]. Traditional diagnostics approaches are often clinical examinations, medical imaging and laboratory tests [3] which can be costly, time consuming and sometimes even inaccessible. As a consequence of this there is a great utility of computational tools that are capable of assisting clinicians in identifying the condition of patients from available data.

Machine learning techniques show a significant potential in healthcare applications, particularly in disease prediction and decision support systems. By learning patterns from historical data the models can predict the condition of a patient given the characteristic data for that patient. Previous studies have shown that applied machine learning algorithms can be used for predicting the presence of heart disease using features such as pulse rate, age and sex [6]. Supervised machine learning techniques such as Naive Bayes, decision trees, random forest and K-nearest neighbours have shown promising predictive performance in this domain [7].

The goal of this study is to design and evaluate various machine learning models that are capable of predicting the presence of heart disease in patients, based on features such as BMI, Age, Gender, and the consumption of alcohol. By analyzing patterns within the chosen variables the proposed model aims to assist healthcare professionals in making more informed decisions. A systematic comparison of four ML classifiers will be implemented and used as a baseline. The aforementioned models will be Decision Tree, Logistic Regression, XGBoost, and Neural Network, but to handle the imbalanced dataset that was publicly available [8] [9], an isolated analysis on four SMOTE variations will be made. The aim is to also perform feature selection on the different model families, comparing the usual filterbased- with the more advanced wrapper/embedded feature selection variations.

The results of the study indicate that the XGBoost model combined with the SMOTE-ENN sampling technique yielded promising performance, as did the Logistic Regression model when used together with the ADASYN sampling method. Furthermore, the findings suggest that the feature selection methods had a limited impact, since the best-performing models achieved higher performance when using simpler feature selection approaches, including the baseline configuration with no feature selection as well as manual feature selection.

II. RELATED WORK

Machine learning (ML) methodologies have been extensively applied in medical domains for disease risk assessment. For instance, Chen et al. [10] developed a risk assessment model targeting the multimorbidity of diabetes, hypertension, and coronary heart disease within a primary care setting in

Shanghai, China. Their retrospective study specifically employed the XGBoost algorithm to construct the predictive model, demonstrating the utility of gradient boosting techniques for complex, comorbid condition analysis. In a related application, Wang et al. [11] investigated the use of ML models for predicting depression risk among patients with sleep disorders. Although the clinical endpoint of this study differs from the cardiovascular focus of the present paper, their methodological approach is noteworthy. The authors implemented and validated a diverse set of eleven different classifiers, including Support Vector Machines (SVM), Decision Trees, XGBoost, and Multilayer Perceptrons (MLP). This comprehensive model comparison serves as a pertinent example of how rigorous ML development and validation can be successfully translated to clinical prediction tasks, a principle that underpins the methodology of the current study.

Bhatt et al. [27] investigated heart disease prediction using a Kaggle cardiovascular disease dataset of 70,000 instances, applying four classifiers Decision Tree, Random Forest, XGBoost, and Multilayer Perceptron in conjunction with k-modes clustering as a preprocessing step. Rather than working with raw continuous features, the study focused on converting attributes such as age into categorical values through binning, and additionally derived composite features including BMI and Mean Arterial Pressure (MAP) to improve predictive power. Hyperparameter tuning was performed using GridSearchCV, and models were evaluated on accuracy, precision, recall, F1-score, and AUC. This work is relevant to the present study on several levels. First, it demonstrates that three of the four classifiers used in our pipeline, Decision Tree, XGBoost, and MLP consistently appear as competitive candidates for heart disease prediction across different datasets and preprocessing strategies. Second, their feature engineering approach, particularly the derivation of BMI as a composite feature, aligns with our own decision to include and subsequently evaluate the exclusion of BMI as a feature in our pipeline. Third, and notably, Bhatt et al. do not address class imbalance, which in our dataset constitutes the primary challenge. Their work thereby highlights precisely the methodological gap that the present study aims to fill through a systematic evaluation of oversampling strategies.

III. BACKGROUND

A. Decision Tree

Decision Trees are a tree-based technique that is used for both classification and regression tasks. In a classification setting, the model recursively partitions the feature space into smaller regions by selecting features and values that best separate the classes. This process starts from the root node, the internal decision nodes and ends at the terminal leaf nodes. The decision trees algorithm can handle vast volumes of information, it is simple to implement and is also the most commonly used in many fields [13].

At each node, the algorithm selects the feature that maximizes the separation between classes according to a splitting criterion such as Gini impurity or information gain. The

dataset is then divided into subsets based on the selected split, and the process continues recursively until a stopping condition is reached.

Decision Trees are particularly suitable for this project because they can naturally handle both numerical and categorical features and provide a high degree of interpretability. The resulting tree structure allows researchers to identify which patient attributes contribute most strongly to the prediction of heart disease. Additionally, feature importance scores derived from the tree can be used to support the feature selection analysis performed in this study. However, a known limitation of Decision Trees is their tendency to overfit the training data, especially when the tree becomes too deep.

B. XGBoost

XGBoost which stands for Extreme Gradient Boosting is an advanced ensemble learning algorithm. Instead of constructing a single tree, XGBoost builds a sequence of decision trees where each new tree attempts to correct the prediction errors of the previous ones. This iterative process gradually improves the model's predictive performance.

At each iteration, XGBoost optimizes an objective function consisting of a loss component and a regularization term. The loss function, also called binary cross-entropy (BCE) measures how well the model fits the training data, while the regularization component penalizes overly complex trees to reduce overfitting [14].

$$\text{BCE} = -\frac{1}{n} \sum_{i=1}^n [y_i \log(p_i) + (1 - y_i) \log(1 - p_i)]$$

XGBoost is particularly well-suited for structured tabular datasets such as healthcare records, which makes it highly relevant for this study. The algorithm includes several features that improve predictive performance, including built-in regularization and efficient tree pruning. Additionally, XGBoost provides feature importance measures that can be used to evaluate the relevance of different attributes in the prediction task.

Due to its strong predictive performance and robustness, XGBoost has become one of the most widely used machine learning algorithms in medical prediction tasks and serves as an important benchmark model in this study. [14]

TABLE I
MACHINE LEARNING MODELS EVALUATED IN THE STUDY

Model	Type	Purpose in Study
Decision Tree	Tree-based classifier	Baseline interpretable model
Logistic Regression	Linear classifier	Statistical baseline model
XGBoost	Ensemble classifier	High-performance model
Neural Network	Deep learning model	Non-linear pattern detection

C. Logistic regression

Logistic regression is a fundamental machine learning model utilized for binary classification tasks. Unlike linear regression models that predict continuous outcomes, logistic regression models the probability that a given input instance

belongs to one of two discrete classes, typically denoted as class 0 or class 1. This is achieved through the application of a logistic function [15], which ensures that any real-valued input is mapped to a probability value strictly within the interval of 0 and 1. Technically, the model extends the linear regression framework by defining a linear predictor, z , which represents the weighted sum of the input features:

$$z = \mathbf{w}^T \mathbf{x} + b$$

Where $\mathbf{w}$ represents the weights, $\mathbf{x}$ is the feature vector, and b is the bias term. To transform this continuous value into a probability, the results are passed through the sigmoid (logistic) function, $\sigma(z)$:

$$\sigma(z) = \frac{1}{1 + e^{-z}}$$

The parameters $\mathbf{w}$ and b are pivotal to the model's accuracy, as they define the relationship between the features and the target variable. These values are typically optimized using Maximum Likelihood Estimation (MLE), a method that identifies parameter values that maximize the likelihood of the observed data being produced by the model [16]. For the model to produce stable and reliable results, certain statistical conditions must be satisfied. A key assumption of logistic regression is the requirement for a sufficiently large sample size [15]. Because the MLE process relies on large-sample (asymptotic) theory, an adequate number of observations is necessary to ensure the coefficients reach statistical significance and the model generalizes effectively to unseen data [16].

D. Neural Networks (MLP)

Neural networks are a class of deep learning models [16] that have gained significant prominence in recent years due to their integration into modern intelligent systems utilizing machine learning. A common architecture is the Multilayer Perceptron (MLP) [17], which typically utilizes a feed-forward structure. This architecture consists of an input layer, followed by one or more hidden layers, and terminates in an output layer. The hidden layers are composed of multiple computational nodes known as neurons. A critical component of the MLP is the implementation of suitable activation functions [18]. These functions provide a mathematical means of mapping the output to a specific range, thereby determining the activation state of a neuron. While the sigmoid function previously discussed in the context of logistic regression is a common choice, the Rectified Linear Unit (ReLU) is also widely utilized. The ReLU function is defined by a simplistic mathematical formulation:

$$f(x) = \max(0, x)$$

The derivative of the ReLU function is also relatively straightforward, which facilitates efficient computation during the training phase [19]:

$$f'(x) = \begin{cases} 0 & \text{if } x < 0 \\ 1 & \text{if } x > 0 \\ \text{undefined} & \text{if } x = 0 \end{cases}$$

The value x passed into the activation function is calculated by taking the dot product of the input vector and its corresponding weights, then adding a bias term b :

$$z = \sum_{i=1}^n w_i x_i + b$$

To iteratively improve model performance, the backpropagation algorithm [20] is employed as the standard mechanism for training. This process begins after the completion of a forward pass. The resulting prediction is compared to the actual target value to calculate the error using a loss function, such as Binary Cross-Entropy for binary classification problems. The backpropagation algorithm subsequently propagates this error backward through the network to determine the contribution of each weight to the total error by calculating the gradient of the loss function with respect to each weight. Following this, gradient descent is applied to update the weights and minimize the loss:

$$w_{\text{new}} = w_{\text{old}} - \eta \cdot \frac{\partial L}{\partial w}$$

In this formulation, $\frac{\partial L}{\partial w}$ represents the gradient of the loss with respect to the weight, and η denotes the learning rate. The learning rate is a hyperparameter that controls the magnitude of the step taken towards the global minimum during each iteration. Neural networks possess the capability to identify patterns and model data characterized by highly complex relationships [21]. However, they are often characterized as "black-box" models, as they do not inherently reveal the internal decision-making logic used to produce a specific output. Furthermore, this model architecture entails significant computational requirements; the training process can be resource-intensive and time-consuming [22], often requiring relatively large sample sizes to achieve optimal results.

E. SMOTE

An initial challenge identified within the dataset was a pronounced class imbalance. This disproportion is particularly problematic for predictive modeling, as standard learning algorithms tend to develop a significant bias toward the majority class, often at the expense of the minority class's predictive accuracy. To mitigate this, Chawla et al. [23] proposed the Synthetic Minority Over-sampling Technique (SMOTE). The core mechanism of the original 2002 algorithm involves over-sampling the minority class by generating "synthetic" examples rather than merely duplicating existing observations. For a given sample in the minority class, the algorithm identifies its k -nearest neighbors. A synthetic sample is then generated by interpolating between the feature vector of the sample (x_i) and one of its neighbors (x_{z_i}). This is calculated by taking the difference between the two vectors, multiplying it by a random number (δ) between 0 and 1, and adding the result to the original feature vector:

$$x_{\text{new}} = x_i + \delta \cdot (x_{z_i} - x_i)$$

Despite its effectiveness, the original SMOTE algorithm has limitations, particularly when interpolating near the decision

boundary. If a minority sample is located near majority class instances, the generated synthetic data may overlap with the majority class, increasing noise and blurring the class boundaries. Han et al. [24] addressed this by introducing Borderline-SMOTE. This variant identifies "borderline" minority samples those most at risk of misclassification. The authors proposed two versions: Borderline-SMOTE1, which interpolates only between a borderline minority sample and its minority neighbors, and Borderline-SMOTE2, which interpolates between the sample and both its minority and majority neighbors (while maintaining a closer proximity to the minority side) to better define the decision boundary.

F. Chi-Square Feature Selection

High-dimensional data often contains redundant or irrelevant features that can lead to the "curse of dimensionality," negatively impacting the generalization of predictive models. To mitigate this, feature selection is used to identify a subset of attributes that maintain a strong relationship with the target class. A prominent statistical approach for this is Chi-Square (χ^2) feature selection, a filter-based method that evaluates the independence between a feature and the class label.

The algorithm assesses the null hypothesis that a specific feature is independent of the class. A high χ^2 statistic indicates that the feature's distribution varies significantly across different classes, suggesting it possesses high discriminative power. For a given contingency table, the χ^2 value is calculated as:

$$\chi^2 = \sum_{i=1}^r \sum_{j=1}^c \frac{(O_{i,j} - E_{i,j})^2}{E_{i,j}} \quad (1)$$

where $O_{i,j}$ is the observed frequency and $E_{i,j}$ is the expected frequency of the i -th interval within the j -th class.

While the standard χ^2 test is designed for categorical data, Liu and Setiono [25] proposed the Chi2 algorithm to extend this capability to numerical attributes through an automated discretization process. The algorithm functions in two distinct phases. In the first phase, it performs a supervised, bottom-up merging of intervals. It begins by treating each distinct value as an interval and merges adjacent intervals if their χ^2 value falls below a significance threshold, determined by a user-defined inconsistency rate. In the second phase, the algorithm further refines the feature set by identifying and removing features that do not contribute to the overall consistency of the dataset. This dual approach ensures that the resulting feature subset is both discretized for efficiency and optimized for predictive relevance.

IV. METHODOLOGY

A. Data Source

The datasets used in this study originates from the CDC, a large national health organization in the US. The data is based upon a health survey which is conducted biannually and the target values is a self reported heart disease diagnosis. The data from the CDC was reformatted and converted to a .csv file in Kaggle, specifically for the years 2020 and 2022. Our

models will be run on the 2022 dataset with its 39 features and 250 000 data points, aswell as the combined dataset of 2020 and 2022, which has 16 features and almost 570 000 data points.

B. Data Analysis

A comprehensive exploratory data analysis was conducted on both the 2022 and the combined 2020/2022 datasets. The primary objective was to assess data integrity, understand the univariate and bivariate feature distributions, and identify key data characteristics such as class imbalance and feature interactions. These insights would directly affect the methodological choices for preprocessing, feature selection, and modelling.

Data Quality and Class Imbalance

The initial phase of the exploratory data analysis focused on assessing the data quality. As the data was collected from an open-source platform, it was important to evaluate its authenticity and usefulness. After a careful review of both the new data source and the two provided datasets, the assessment was that both datasets fulfilled the required quality criteria.

However, the primary challenge became evident when examining the target variable. As illustrated in Fig. 1 and Fig. 2, both datasets exhibit a significant class imbalance. In the 2022 dataset, the minority class representing individuals with heart disease accounts for approximately 5.46% of the observations, while in the combined dataset this proportion is slightly higher at 7.21%.

Although such distributions may reflect realistic real world conditions, the relatively small representation of the minority class can pose challenges for predictive modeling. In particular, models trained on imbalanced data tend to become biased toward the majority class, which can lead to poor detection of heart disease cases.

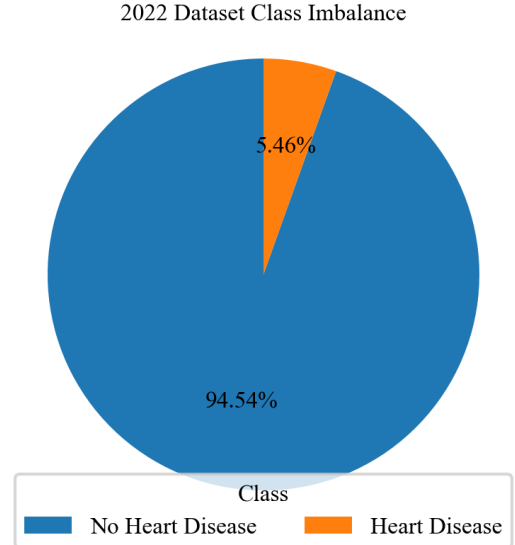


Fig. 1. Distribution of heart disease cases showing the class imbalance (2022 dataset).

2020 + 2022 Dataset Class Imbalance

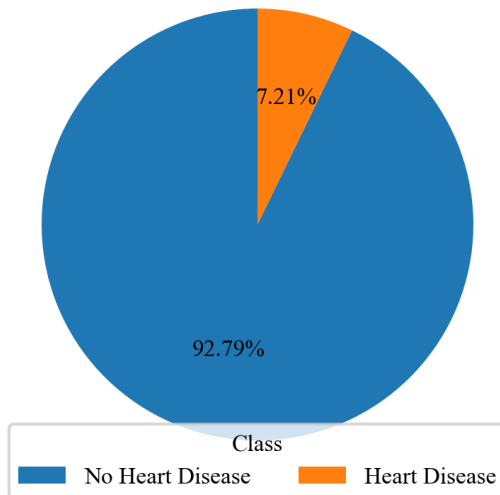


Fig. 2. Distribution of heart disease cases showing the class imbalance (combined dataset).

Insights from Data Distribution

To get a better understanding of the composition of the data population, univariate analysis was performed on some key features from both datasets. For instance, the gender distribution among heart disease patients, shown in Fig. 3 and Fig. 4, reveals that a higher proportion of cases are male with a percentage of 60.39% and 63.29% for the 2022 and combined datasets, respectively.

Despite this difference within the positive class, the overall gender distribution in the datasets remains relatively balanced.

2020 + 2022 Dataset Gender Distribution Among Heart Disease Patients

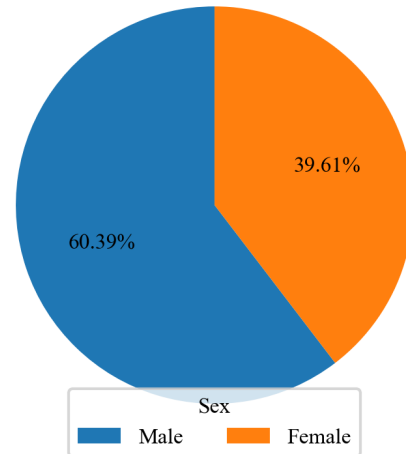


Fig. 4. Gender distribution among heart disease patients (combined dataset).

A comparative analysis of the BMI distribution was conducted for both the 2022 dataset and the combined dataset, as illustrated in Fig. 5 and Fig. 6. The results indicate that the overall distribution remains highly consistent across the two datasets.

In both cases, the majority of individuals fall within the overweight and obese categories, while the proportion of underweight individuals is minimal. The similarity in distributions suggests that combining the datasets from different years does not significantly alter the underlying BMI characteristics of the population. This consistency supports the validity of merging the datasets for further analysis.

2022 Dataset Gender Distribution Among Heart Disease Patients

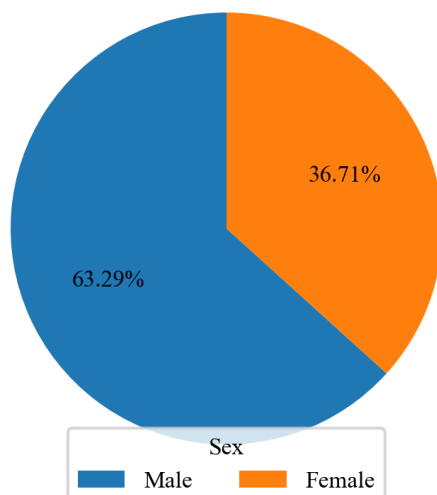


Fig. 3. Gender distribution among heart disease patients (2022 dataset).

2022 Dataset BMI Distribution

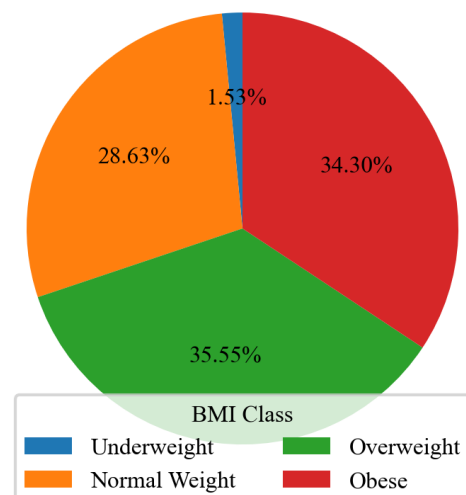


Fig. 5. BMI distribution (2022 dataset).

2020 + 2022 Dataset BMI Distribution

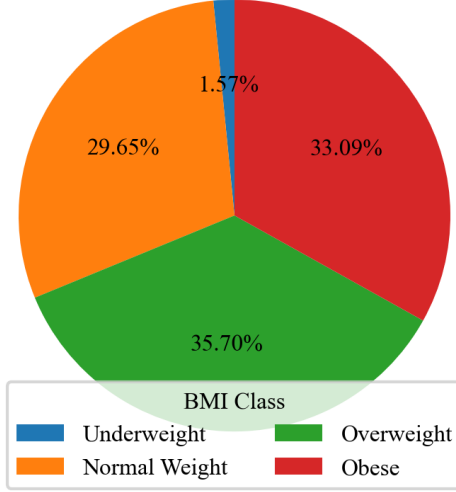


Fig. 6. BMI distribution (combined dataset).

Correlation Analysis

To quantify linear relationships and guide the feature selection process, a correlation matrix was generated for the combined dataset, as presented in Figure 7. The heatmap reveals several important insights for the purpose of the methodological approach.

The matrix highlights features most linearly correlated with heart disease. The feature with the largest correlation with the target variable *HeartDisease* is *AgeCategory*, indicating a positive linear correlation between age and the possibility of having a heart disease. Some other notable features that show correlation with the target variable are *GenHealth*, *Stroke*, *DiffWalking* and *Diabetic*.

The heatmap also reveals correlations between features as well. For example, there is a moderate correlation between *DiffWalking* and *GenHealth*, as well as between *Diabetic* and *KidneyDisease* or *AgeCategory*. The presence of multicollinearity can destabilize models that assume little or no conditional independence between features. These findings can also hint about the value of feature selection techniques that can potentially navigate these dependencies.

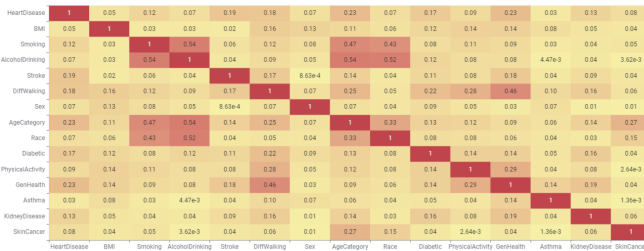


Fig. 7. Correlation matrix of features and target variable (combined dataset).

V. APPROACH DESCRIPTION

This section describes how each of the methods introduced in the background section was configured and adapted to suit the specific characteristics of the heart attack prediction problem. All methodological decisions were informed by the findings of the exploratory data analysis, most notably the severe class imbalance identified in the target variable and the multicollinearity observed among several features. All four models were implemented using the sklearn libraries except for the XGBoost, which used the "XGBoost" library. All models were trained with the same training and testing split to make sure the comparison was fair. All SMOTE variants were also used on each model.

A. Handling Class Imbalance

As established in the data analysis section, the target variable exhibits a pronounced class imbalance, with the positive class (heart disease) representing only 7.21% of the total dataset. To prevent the models from developing a bias toward the majority class, four distinct oversampling strategies were evaluated: SMOTE, Borderline-SMOTE1, ADASYN, and SMOTE-ENN.

A common preprocessing step was applied uniformly across all four strategies prior to the oversampling procedure itself. Rather than applying oversampling to the full dataset, a majority class under sampling step was first performed, reducing the majority class to three times the size of the minority class. This combined under- and oversampling approach was adopted for two reasons: first, to reduce the computational cost of the subsequent oversampling step given the large scale of the dataset; and second, to provide a more balanced starting distribution from which synthetic samples could be generated more effectively.

Each oversampling technique was then applied to this reduced combined dataset. Standard SMOTE was used as the baseline oversampling method, generating synthetic minority samples through linear interpolation between a minority instance and one of its k -nearest neighbors. As discussed in the background, a limitation of vanilla SMOTE is its potential to generate noisy samples near the decision boundary. To address this, Borderline-SMOTE1 was applied as a variant that restricts synthetic sample generation to borderline minority instances, interpolating exclusively between a borderline sample and its minority-class neighbors, thereby producing a cleaner decision boundary.

ADASYN (Adaptive Synthetic Sampling) was additionally evaluated as an alternative to SMOTE. Unlike SMOTE, which generates synthetic samples uniformly across the minority class, ADASYN adaptively generates more samples in regions where the minority class is harder to learn, i.e., where it is surrounded by a higher proportion of majority class instances. This makes ADASYN particularly suitable for datasets with complex decision boundaries.

Finally, SMOTE-ENN was evaluated as a hybrid technique. This method first applies standard SMOTE to generate synthetic minority samples and subsequently applies

Edited Nearest Neighbours (ENN) as a cleaning step. The ENN step removes any instance, from either class, whose class label differs from the majority class label among its nearest neighbors. This combination aims to both increase minority class representation and refine the class boundaries by removing potentially noisy or ambiguous instances. In the implementation, the ENN step was parallelized across 8 CPU cores to improve runtime efficiency.

All oversampling operations were applied exclusively to the training data, ensuring that no synthetic samples were present in the held-out test set, which preserves the integrity of the evaluation.

B. Decision Tree

The Decision Tree was configured as the primary interpretable baseline model. Given the known tendency of decision trees to overfit, particularly as tree depth increases, several constraints were imposed to promote generalization. The maximum tree depth was set to 5, limiting the number of consecutive splits and preventing the model from learning overly specific patterns in the training data. A minimum of 10 samples was required to allow a node to be split (`min_samples_split=10`), and leaf nodes were required to contain at least 5 samples (`min_samples_leaf=5`). These constraints were motivated by the large size of the dataset, where unrestricted growth would readily produce a tree that memorizes training examples rather than generalizing to unseen patients.

The splitting criterion was set to entropy (information gain) rather than Gini impurity. In the context of heart disease prediction, where the accurate identification of the minority positive class is clinically important, entropy-based splitting provides a slightly more sensitive measure of class separation. A fixed random state of 67 was used across all models to ensure reproducibility of results.

C. Logistic Regression

Logistic Regression was configured as the statistical linear baseline. As noted in the background, the model relies on Maximum Likelihood Estimation, which benefits from a sufficiently large sample size. Given that the combined 2020/2022 dataset satisfies this requirement, logistic regression is a well-suited baseline for this problem.

To address the multicollinearity identified in the correlation analysis between features such as `DiffWalking` and `GenHealth`, as well as `Diabetic` and `KidneyDisease`, an ElasticNet regularization penalty was applied. ElasticNet combines both L1 and L2 penalties, where the L1 component encourages sparsity by driving some feature weights to zero (effectively performing feature selection), and the L2 component handles correlated features by distributing weight among them. The mixing ratio was set to `l1_ratio=0.5`, applying equal weight to both penalties. The regularization strength was set to `C=1.0`, representing the default inverse regularization strength.

The `saga` solver was selected, as it is the only solver in scikit-learn’s logistic regression implementation that supports the ElasticNet penalty and is efficient for large datasets. The maximum number of iterations was set to 1000 to ensure convergence.

D. Neural Network (MLP)

The Multilayer Perceptron was configured with two hidden layers of 64 and 32 neurons respectively. This progressively narrowing architecture compresses the learned representation at each layer, encouraging the network to extract increasingly abstract features from the input. As introduced in the background, ReLU was selected as the activation function for the hidden layers due to its computational efficiency and its ability to mitigate the vanishing gradient problem, which can be problematic with sigmoid activations in deeper networks.

The Adam optimizer was used for training, selected for its adaptive learning rate properties which generally lead to faster and more stable convergence than standard gradient descent on tabular data. A maximum of 300 training iterations was set, alongside an L2 regularization penalty of `alpha=0.0001` to reduce overfitting. The neural network was given a higher iteration budget compared to simpler models due to the increased complexity of learning non-linear decision boundaries across the relatively high-dimensional feature space.

E. XGBoost

XGBoost was configured as the primary high-performance benchmark model. An ensemble of 100 gradient-boosted trees was built (`n_estimators=100`), with each individual tree constrained to a maximum depth of 5, consistent with the depth restriction applied to the standalone Decision Tree. The learning rate was set to 0.1, providing a moderate step size during the boosting iterations to balance convergence speed and generalization.

To further reduce overfitting and improve generalization, two stochastic subsampling strategies were employed. Row subsampling (`subsample=0.8`) means that each tree is trained on a random 80% sample of the training instances, introducing variance that reduces correlation between trees. Column subsampling (`colsample_bytree=0.8`) similarly restricts each tree to a random 80% subset of features, which is particularly beneficial given the multicollinearity identified in the data analysis, as it prevents the model from repeatedly relying on the same dominant features. The evaluation metric was set to log-loss, which directly penalizes confident incorrect predictions, making it appropriate for the class-imbalanced binary classification task.

It is worth noting that the `scale_pos_weight` parameter, which instructs XGBoost to upweight the minority class during training, was deliberately omitted. Since class imbalance is instead addressed through the oversampling pipeline described above, using both mechanisms simultaneously would risk over-correcting the class distribution.

VI. EXPERIMENTAL DESIGN

To identify the most optimal pipeline for the given datasets, a modular experimental framework was used. Instead of testing each individual component in isolation, the method used employs a combinatorial approach, evaluating every possible permutation of feature selection, sampling techniques, and classification models for each dataset.

A. Experimental Workflow

The workflow is structured as a directed sequential pipeline consisting of four primary stages:

- 1) **Dataset Selection:** Experiments were conducted on two distinct datasets: the standalone **2022 dataset** and a combined **2020+2022 dataset**.
- 2) **Feature Selection:** Five strategies were evaluated to refine the input data:
 - **None (Baseline):** No features were removed to provide a performance baseline.
 - **Manual:** Selection based on relevance and clinical significance.
 - **Near Constant:** Removal of features with low variance that contribute minimal information.
 - **Chi-Square:** Statistical selection based on independence tests between features and the target.
 - **No BMI:** A targeted exclusion of the Body Mass Index (BMI) feature to test its specific impact on model bias or performance.
- 3) **Sampling Technique:** To address class imbalance, four oversampling methods were compared against a baseline of no sampling: **SMOTE**, **Borderline SMOTE**, **ADASYN**, and **SMOTE-ENN**.
- 4) **Model Selection:** Four diverse machine learning architectures were deployed:
 - **Decision Trees**
 - **Logistic Regression**
 - **Neural Networks**
 - **XGBoost**

The total search space consists of $2 \times 5 \times 5 \times 4 = 200$ unique pipeline combinations, as demonstrated in Fig. 8. Each pipeline was evaluated using a stratified 10-fold cross-validation to ensure the robustness of the results.

B. Evaluation Metrics and full pipeline

Each combination was assessed based on three key performance indicators: **F1-Score**, **Recall**, and **ROC-AUC**. Given the imbalanced dataset, more emphasis was placed on Recall and F1-Score to ensure the model effectively identifies the minority class while maintaining a balance between precision and sensitivity.

The complete experimental workflow, encompassing dataset selection, feature engineering, sampling, and model training, is illustrated in Fig. 8.

VII. RESULTS

A. Baseline Results

The baseline experiments conducted reveal a pattern across both datasets, where all four models achieve extremely high accuracy and relatively high precision as shown in Fig. 9 and Fig. 10 below. However, they fail to give a useful score on recall and F1 for the minority class; this confirms the initial theory that a standard classifying model that is heavily trained on an imbalanced dataset, develops a strong bias towards the majority class.

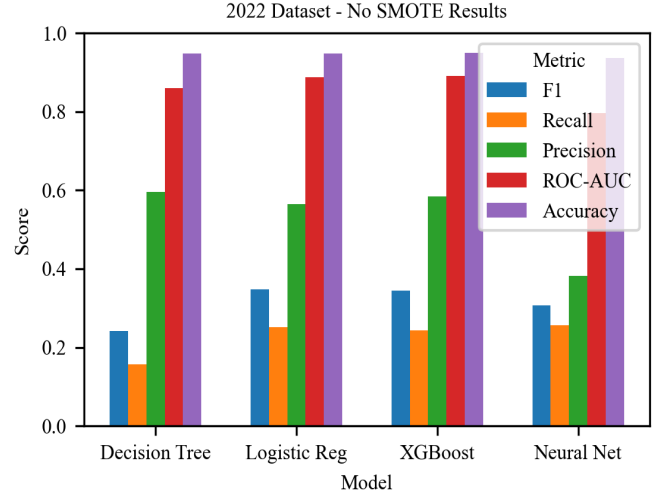


Fig. 9. Baseline Results for 2022 dataset, using no SMOTE technique.

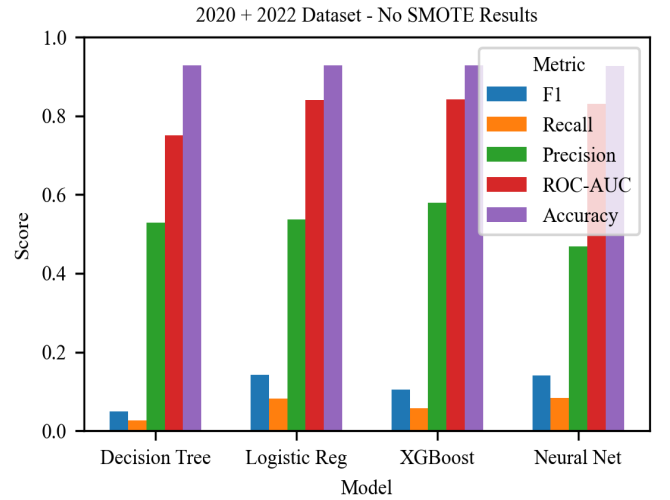


Fig. 10. Baseline Results for combined dataset, using no SMOTE technique.

B. Optimal SMOTE Results

The results in Fig. 11 and Fig. 12 demonstrate a significant improvement in model performance after applying SMOTE-ENN technique. Compared to the baseline results, there is a

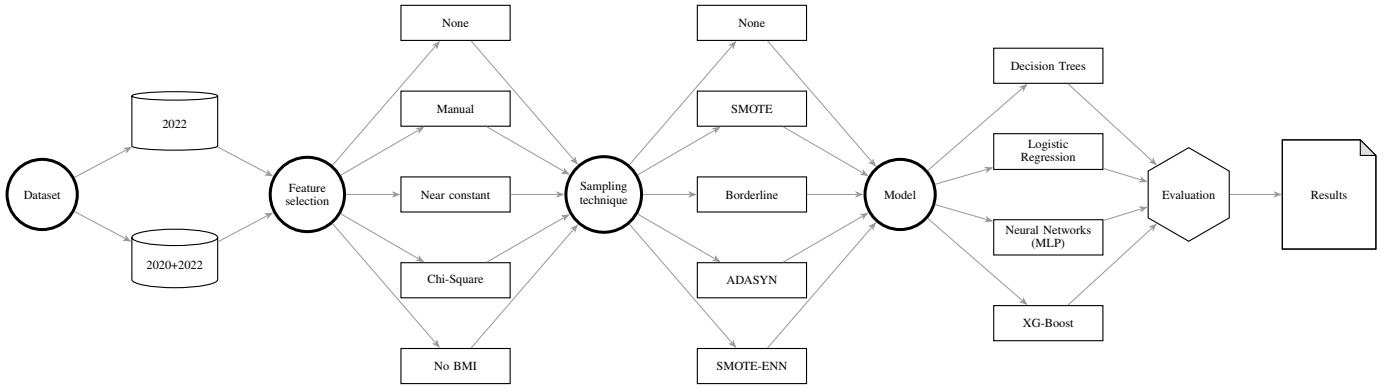


Fig. 8. Full experimental pipeline encompassing all $2 \times 5 \times 5 \times 4 = 200$ unique combinations of dataset, feature selection strategy, oversampling technique, and classifier.

clear increase in both recall and F1-score across all models, indicating a more effective identification of the minority class. This confirms that addressing class imbalance through synthetic sampling and noise reduction leads to more balanced predictions. Although a slight decrease in overall accuracy can be observed, this tradeoff is expected, as the models are no longer biased toward the majority class. Overall, the application of SMOTE-ENN results in a substantial improvement in the model's ability to detect heart disease cases, which is particularly important in a medical context where minimizing false negatives is critical.

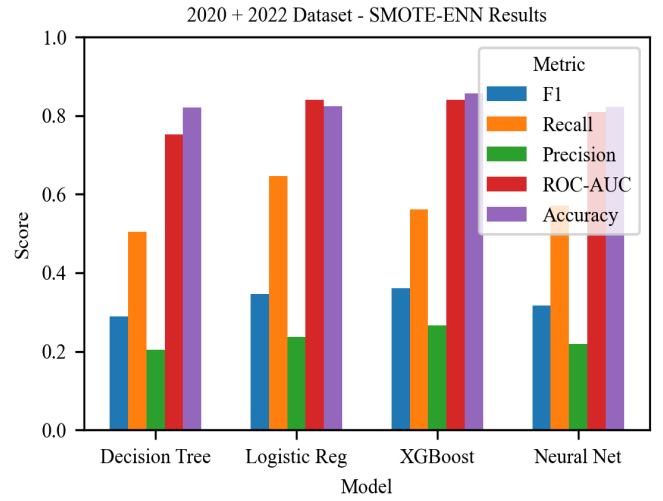


Fig. 12. Best SMOTE Results for the combined dataset, using the SMOTE-ENN technique.

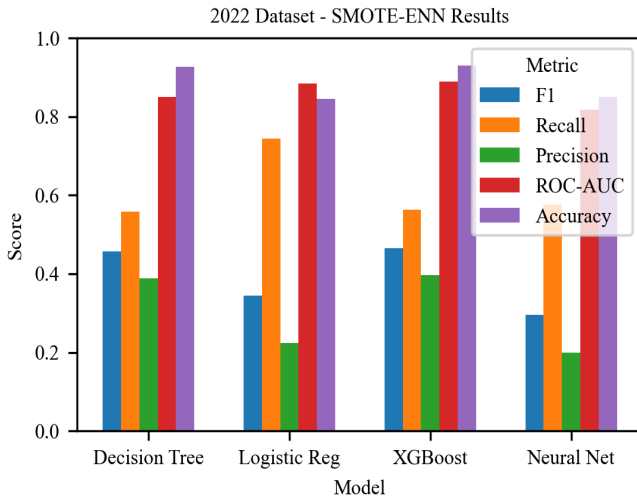


Fig. 11. Best SMOTE results for the 2022 dataset, using the SMOTE-ENN technique.

C. Top Performing Pipelines

The results in Table II show that XGBoost consistently achieves the highest F1-scores when combined with SMOTE-ENN, regardless of the feature selection method applied. This suggests that ensemble-based models benefit significantly from synthetic sampling techniques in imbalanced settings. The relatively small variation across feature selection strategies indicates that feature selection has a limited impact on XGBoost's performance, which is likely due to its inherent ability to prioritize relevant features during training. While ADASYN also produces strong results, SMOTE-ENN consistently yields higher F1-scores, indicating that the additional noise reduction step improves model performance.

TABLE II
2022 DATASET - RANKED ON F1

Rank	Model	SMOTE	Features	F1	Recall	AUC
1	XGBoost	SMOTE-ENN	Manual	0.467	0.557	0.883
2	XGBoost	SMOTE-ENN	No_BMI	0.465	0.565	0.888
3	XGBoost	SMOTE-ENN	Baseline	0.465	0.563	0.888
4	XGBoost	SMOTE-ENN	Chi-Square	0.464	0.566	0.885
5	XGBoost	ADASYN	Baseline	0.458	0.592	0.890
6	DecisionTree	SMOTE-ENN	Baseline	0.456	0.558	0.850
7	DecisionTree	SMOTE-ENN	Manual	0.455	0.552	0.839
8	XGBoost	Default	Baseline	0.455	0.595	0.890

The results in Table III show that the combined dataset is more challenging to model, as reflected by the generally lower F1-scores compared to the 2022 dataset. This is likely due to the reduced number of available features, which limits the model's ability to capture complex patterns. Similar to the 2022 dataset, SMOTE-ENN consistently demonstrates strong performance, reinforcing its effectiveness as a sampling strategy in imbalanced settings. XGBoost remains the best-performing model. This suggests the performance gap between models is smaller, with Logistic Regression and Neural Networks achieving comparable results. This suggests that model choice has a less pronounced impact when using the combined dataset.

TABLE III
COMBINED DATASET - RANKED ON F1

Rank	Model	SMOTE	Features	F1	Recall	AUC
1	XGBoost	SMOTE-ENN	Baseline	0.360	0.561	0.839
2	XGBoost	SMOTE-ENN	Manual	0.358	0.564	0.839
3	LogReg	SMOTE-ENN	No_BMI	0.353	0.520	0.835
4	LogReg	SMOTE-ENN	Baseline	0.346	0.647	0.839
5	XGBoost	Default	Baseline	0.331	0.706	0.839
6	NeuralNetwork	SMOTE-ENN	Manual	0.328	0.547	0.815
7	NeuralNetwork	SMOTE-ENN	Baseline	0.316	0.572	0.808
8	XGBoost	Borderline	Baseline	0.313	0.755	0.837

When ranked by recall, the results in Table IV present a different perspective, with Logistic Regression consistently achieving the highest recall values. In this setting, ADASYN and Borderline-SMOTE emerge as the most effective sampling techniques, likely due to their focus on generating synthetic samples in more difficult regions near the decision boundary. However, this improvement in recall comes at the cost of significantly lower F1-scores, indicating a trade-off where the models achieve higher sensitivity but reduced overall balance. This suggests an increase in false positives, as more negative instances are incorrectly classified as positive.

TABLE IV
2022 DATASET - RANKED BY RECALL

Rank	Model	SMOTE	Features	Recall	F1	AUC
1	LogReg	ADASYN	Manual	0.811	0.291	0.880
2	LogReg	Borderline	Manual	0.810	0.292	0.880
3	LogReg	Borderline	No_BMI	0.803	0.300	0.885
4	LogReg	Borderline	Baseline	0.802	0.302	0.885
5	LogReg	ADASYN	Baseline	0.802	0.299	0.884
6	LogReg	ADASYN	No_BMI	0.801	0.299	0.884
7	DecisionTree	Borderline	Baseline	0.786	0.253	0.847
8	DecisionTree	ADASYN	Baseline	0.784	0.254	0.850

The results in Table V are highly consistent with those observed in the 2022 dataset, with Logistic Regression combined with ADASYN and Borderline-SMOTE again achieving the highest recall values. This reinforces the observation that these sampling techniques are particularly effective for maximizing sensitivity in imbalanced classification tasks. In contrast to the previous dataset, XGBoost also appears among the top performing pipelines, although it does not surpass Logistic Regression in terms of recall. Additionally, the top ranked configurations are limited to manual and baseline feature selection, suggesting that more complex feature selection strategies provide limited benefit in this setting.

TABLE V
COMBINED DATASET - RANKED BY RECALL

Rank	Model	SMOTE	Features	Recall	F1	AUC
1	LogReg	ADASYN	Manual	0.827	0.288	0.838
2	LogReg	Borderline	Manual	0.820	0.288	0.836
3	LogReg	ADASYN	Baseline	0.819	0.290	0.838
4	LogReg	Borderline	Baseline	0.813	0.291	0.837
5	XGBoost	Borderline	Manual	0.813	0.290	0.836
6	XGBoost	ADASYN	Manual	0.802	0.297	0.837
7	LogReg	Default	Baseline	0.780	0.308	0.839
8	LogReg	Default	Manual	0.779	0.307	0.839

D. Most Optimal Pipeline

Based on the analysis of all the metrics on the pipeline combinations that were evaluated, the recommended configuration is XGBoost with SMOTE-ENN and manual feature selection on the 2022 data, as illustrated in Fig. 13. While Logistic Regression with ADASYN achieves a substantially higher recall, however the F1 score drops significantly, indicating that the model generates far more false positives upon which decreases the models precision. The SMOTE-ENN is vital to the optimum pipeline and is reflected in its consistently superior F1 score over the various models. In a medical context, that a large number of healthy patients would be incorrectly flagged with a heart disease and result in unnecessary follow up diagnostics. It should be known though that this approach is context dependent, thus in medical settings where resources are constrained, this is the optimal approach to reduce costs and patient anxiety.

VIII. CONCLUSION

This paper showed a systematic overview of 200 machine learning pipelines for binary heart disease prediction on CDC's two health survey data. The results demonstrated that class imbalance handling is the most impactful aspect in performance. All four models defaulted to predicting the majority class when applied without oversampling, which would be an unacceptable in any real world health application. XGBoost emerged as the strongest classifier across the datasets and the various sampling methods. Furthermore, Logistic Regression showed the highest recall scores but suffered slightly on precision which resulted on a lower F1 score.

Based on these findings, the recommended pipeline for prediction is XGBoost with SMOTE-ENN and manual selection

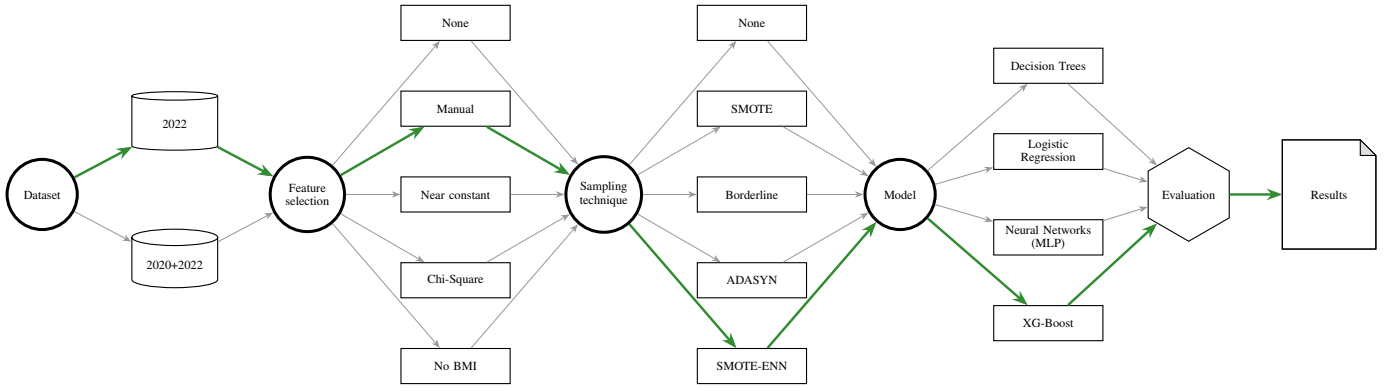


Fig. 13. Complete experimental pipeline encompassing all $2 \times 5 \times 5 \times 4 = 200$ unique combinations of dataset, feature selection strategy, oversampling technique, and classifier. Green arrows indicate the recommended optimal pipeline: 2022 dataset with SMOTE-ENN oversampling and XGBoost classifier (Recall = 0.557, F1 = 0.467).

or the baseline configuration. For applications where maximizing the the sensitivity is the primary objective, Logistic Regression with ADASYN with manual feature selection was the most viable alternative.

Although some acknowledgments should be mentioned in regards to the limitations of this study, such as that the target variable is based on self reported diagnosis which preferably should be reexamined by a medical professional. The datasets are additionally drawn exclusively from the American population, which in general may not correspond to other healthcare contexts around the world. Future work could explore and expand on fine tuning the hyper-parameters of the models as well as utilize more advanced feature selection methodologies. Furthermore, the models would potentially benefit from training on other datasets for richer clinical precision, for example health records or lab results.

REFERENCES

- [1] Centers for Disease Control and Prevention, "About Heart Disease," 2024. [Online]. Available: <https://www.cdc.gov/heart-disease/about/index.html>
- [2] World Health Organization, "Cardiovascular Diseases (CVDs)," 2024. [Online]. Available: [https://www.who.int/news-room/factsheets/detail/cardiovascular-diseases-\(cvds\)](https://www.who.int/news-room/factsheets/detail/cardiovascular-diseases-(cvds))
- [3] "The economic burden of CVD in Europe," *Eur. Heart J.*, 2009. [Online]. Available: <https://pmc.ncbi.nlm.nih.gov/articles/PMC2722492/>
- [4] European Society of Cardiology, "Understanding the burden of CVD: facts and figures," 2024. [Online]. Available: <https://www.escardio.org/>
- [5] "Projected growth of cardiovascular disease burden due to aging," *PubMed*, 2024. [Online]. Available: <https://pubmed.ncbi.nlm.nih.gov/39270739/>
- [6] S. Mohan, C. Thirumalai, and G. Srivastava, "Effective Heart Disease Prediction Using Hybrid Machine Learning Techniques," *IEEE Access*, vol. 7, pp. 81542–81554, 2019.
- [7] D. Shah, S. Patel, and S. K. Bharti, "Heart Disease Prediction using Machine Learning Techniques," *SN Comput. Sci.*, vol. 1, no. 6, p. 345, 2020.
- [8] Centers for Disease Control and Prevention (CDC), "Behavioral Risk Factor Surveillance System (BRFSS) 2020 Annual Data," [Online]. Available: https://www.cdc.gov/brfss/annual_data/annual_2020.html
- [9] Centers for Disease Control and Prevention (CDC), "Behavioral Risk Factor Surveillance System (BRFSS) 2022 Annual Data," [Online]. Available: https://www.cdc.gov/brfss/annual_data/annual_2022.html
- [10] N. Chen *et al.*, "Development of a risk assessment model for multi-morbidity of diabetes, hypertension, and coronary heart disease with XGBoost in primary care in Shanghai, China," *BMC Cardiovasc. Disord.*, vol. 26, no. 1, p. 103, 2026.
- [11] H. Wang *et al.*, "Development, validation, and visualization of a machine learning-based predictive model for depression risk in sleep disorder patients," *BMC Psychiatry*, vol. 26, no. 1, p. 91, 2026.
- [12] H. He and E. A. Garcia, "Learning from imbalanced data," *IEEE Transactions on Knowledge and Data Engineering*, vol. 21, no. 9, pp. 1263–1284, 2009.
- [13] B. T. Jijo and A. M. Abdulazeez, "Classification Based on Decision Tree Algorithm for Machine Learning," *Journal of Applied Science and Technology Trends*, vol. 2, no. 1, pp. 20–28, 2021.
- [14] C. Wang, C. Deng, and S. Wang, "Imbalance-XGBoost: Leveraging weighted and focal losses for binary label-imbalanced classification with XGBoost," *Pattern Recognition Letters*, vol. 136, pp. 190–197, 2020.
- [15] E. Boateng and D. Abaye, "A Review of the Logistic Regression Model with Emphasis on Medical Research," *J. Data Anal. Inf. Process.*, vol. 7, no. 4, pp. 190–207, 2019.
- [16] D. W. Hosmer Jr., S. Lemeshow, and R. X. Sturdivant, *Applied Logistic Regression*, 3rd ed. Hoboken, NJ, USA: Wiley, 2013.
- [17] L. Hardesty, "Explained: Neural networks," *MIT News*, Apr. 14, 2017. [Online]. Available: <https://news.mit.edu/2017/explained-neural-networks-deep-learning-0414>
- [18] M. C. Popescu, V. E. Balas, L. Perescu-Popescu, and N. Mastorakis, "Multilayer perceptron and neural networks," *WSEAS Trans. Circuits Syst.*, vol. 8, no. 7, pp. 579–588, 2009.
- [19] J. Singh and R. Banerjee, "A Study on Single and Multi-layer Perceptron Neural Network," in *Proc. 3rd Int. Conf. Comput. Methodologies Commun. (ICCMC)*, Erode, India, 2019, pp. 35–40.
- [20] S. Raschka, "What is the derivative of the ReLU activation function?" *sebastianraschka.com*, 2017. [Online]. Available: <https://sebastianraschka.com/faq/docs/relu-derivative.html>
- [21] R. Rojas, "The Backpropagation Algorithm," in *Neural Networks*. Berlin, Heidelberg, Germany: Springer, 1996, pp. 149–182.
- [22] J. S. Almeida, "Predictive non-linear modeling of complex data by artificial neural networks," *Curr. Opin. Biotechnol.*, vol. 13, no. 1, pp. 72–76, 2002.
- [23] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic Minority Over-sampling Technique," *J. Artif. Intell. Res.*, vol. 16, pp. 321–357, 2002.
- [24] H. Han, W.-Y. Wang, and B.-H. Mao, "Borderline-SMOTE: A New Over-Sampling Method in Imbalanced Data Sets Learning," in *Proc. Int. Conf. Intelligent Computing (ICIC)*, vol. 3644, 2005, pp. 878–887.
- [25] H. Liu and R. Setiono, "Chi2: Feature selection and discretization of numeric attributes," in *Proceedings of the 7th IEEE International Conference on Tools with Artificial Intelligence*, 1995, pp. 388–391.
- [26] M. Cilimkovic, "Neural networks and back propagation algorithm," *Inst. Technol. Blanchardstown*, Dublin, Ireland, vol. 15, no. 1, p. 18, 2015.
- [27] C. M. Bhatt, P. Patel, T. Ghetia, and P. L. Mazzeo, "Effective Heart Disease Prediction Using Machine Learning Techniques," *Algorithms*, vol. 16, no. 2, p. 88, 2023. [Online]. Available: <https://doi.org/10.3390/a16020088>