THOMPSON RIVERS UNIVERSITY

Data-Driven Heart Disease Prediction Using Neural Networks

By

Reza Heidari

A PROJECT SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF

Master of Science in Data Science

KAMLOOPS, BRITISH COLUMBIA

April, 2024

SUPERVISOR

Mateen Shaikh

ABSTRACT

This research leverages machine learning to tackle the challenge of binary classification in heart disease, aiming to enhance both public health and medical research. It employs the Multi-Layer Perceptron (MLP) algorithm, renowned for its predictive capabilities, and focuses on two key methodologies: feature selection through Information Gain (IG) and the Synthetic Minority Over-sampling Technique (SMOTE) to manage imbalanced datasets. The study utilizes data from the CDC's Behavioural Risk Factor Surveillance System (BRFSS), which conducts annual health surveys in the United States. Despite initial expectations, the efficacy of the SMOTE algorithm was somewhat underwhelming. Nevertheless, upon conducting feature selection, it became evident that employing only 8 predictors out of the total 17 confers advantages, notably reducing complexity for this dataset. The most successful model, based on AUR-PR criteria, incorporates two hidden layers with neurons counts of 16, 18 respectively. This configuration achieved an accuracy of 91.55% and an AUC-PR of 0.321 on test dataset, demonstrating its efficacy in predicting heart disease.

Key Words: Neural Networks; Heart Disease Classification; Multi-Layer Perceptron (MLP); Feature Selection; Information Gain; SMOTE Algorithm.

Contents

1 Introduction **1**

1.1 Data Source and Description 2

1.2 Organization of the Report 3

2 Literature Review **6**

3 Methodology **22**

3.1 Data Preprocessing 22

3.1.1 Feature Transformation 23

3.1.2 Normalization 23

3.1.3 Feature Selection 24

3.1.4 Training-Test Split 24

3.2 Handling Imbalanced Data 24

3.3 Evaluation Metrics 25

3.4 Optimizing Predictive Modeling 28

4 Discussion **31**

4.1 Model and Dataset Overview 31

4.2 SMOTE Evaluation 35

4.3 Feature Selection via IG 38

4.4 Model Scaling 41

5 Conclusion **44**

5.1 Future Work 46

A **51**

List of Figures

2.1 A taxonomy of neural network architectures [Gardner and Dorling, 1998]. 7

2.2 A multilayer perceptron with two hidden layers [Gardner and Dorling, 1998]. 8

2.3 Artificial neural networks [Krogh, 2008]. 9

2.4 The logistic function $\frac{1}{1+e^{-x}}$ [Gardner and Dorling, 1998]. . . . 9

2.5 Structure of a single neuron. 10

2.6 Visualizing the effects of network complexity on model fit [Krogh, 2008]. 12

2.7 Methods to handle the imbalanced data [Spelmen and Porkodi, 2018]. 18

2.8 Baseline Sensitivity in PRC Plot: Class Ratio Impact with 1:1 ratio and 9:1 ratio of P:N. 21

3.1 Precision-Recall: AUC-PR Analysis. 27

4.1 Top 258 Neural Network Models by AUC-PR. 32

4.2 PRC Plot on 1% Dataset and 17 Predictors for training data (16-18-0 architecture). 33

4.3 PRC Plot on 1% Dataset and 17 predictors for Test Data (16-18-0 architecture). 34

4.4 The Number of Neurons' Impact on AUC-PR. 35

4.5 The Number of Hidden Layers' Impact on AUC-PR. 36

4.6 PRC Plot on 1% Dataset and 17 Predictors for training data (16-18-0 architecture) by using SMOTE. 37

4.7 PRC Plot on 1% Dataset and 17 Predictors for Test Data (16-18-0 architecture) by using SMOTE. 38

4.8 Impact of Predictor Count on Evaluation Metrics. 39

4.9 PRC Plot on 1% Dataset and 8 Predictors for training data (16-18-0 architecture). 41

4.10 PRC Plot on 1% Dataset and 8 Predictors for Test Data (16-18-0 architecture). 42

4.11 PRC Plot on Entire Dataset and 8 Predictors for training data (16-18-0 architecture). 42

4.12 PRC Plot on Entire Dataset and 8 Predictors for Test Data (16-18-0 architecture). 43

List of Tables

1.1 Summary of Variables 4

2.1 Information Gain for each feature from the analysis on the heart disease dataset [Khemphila and Boonjing, 2011]. 15

2.2 Accuracy of ANN Classifier Used Feature Selection [Khemphila and Boonjing, 2011]. 16

3.1 Confusion Matrix 26

4.1 Information Gain Values 40

5.1 Evaluation Metrics on Different Approaches. 46

A.1 Confusion Matrix on 1% Dataset and 17 predictors for training data (16-18-0 architecture). 51

A.2 Confusion Matrix on 1% Dataset and 17 Predictors for Test Data (16-18-0 architecture). 52

A.3 Confusion Matrix on 1% Dataset and 17 Predictors for training data (16-18-0 architecture) by using SMOTE. 52

A.4 Confusion Matrix on 1% Dataset and 17 Predictors for Test Data (16-18-0 architecture) by using SMOTE. 52

A.5 Confusion Matrix for 1% Dataset and 8 Predictors for training data (16-18-0 architecture). 52

A.6 Confusion Matrix for 1% Dataset and 8 Predictors for Test Data (16-18-0 architecture). 53

A.7 Confusion Matrix for Entire Dataset and 8 Predictors for training data (16-18-0 architecture). 53

A.8 Confusion Matrix for Entire Dataset and 8 Predictors for Test Data (16-18-0 architecture). 53

Chapter 1

Introduction

Cardiovascular diseases (CVDs) pose a significant challenge to global public health, affecting individuals, families, and healthcare systems worldwide. Among CVDs, heart disease stands out as a primary cause of morbidity and mortality, highlighting the urgent need for proactive risk assessment, prevention, and management strategies [WHO, 2021]. Heart disease includes various disorders affecting the heart and blood vessels, ranging from blood vessel disease to heart rhythm problems. These conditions necessitate thorough diagnosis using patient history, clinical examinations, and advanced medical analyses such as electrocardiograms (ECGs), echocardiograms, and imaging techniques. With heart disease causing approximately 17.9 million deaths annually, it remains a leading cause of global mortality. Early detection and appropriate treatment are crucial in preventing fatalities and reducing the risks associated with heart disease [Ozcan and Peker, 2023]. This research endeavors to harness the power of machine learning, specifically neural networks, to advance our understanding and prediction of heart disease. This

approach involves feature selection through information gain (IG) to address critical aspects in the prediction of heart disease.

1.1 Data Source and Description

The Behavioral Risk Factor Surveillance System (BRFSS), a program established by the Centers for Disease Control and Prevention (CDC) to observe and evaluate health-related behaviors and outcomes among U.S. adults, provides the data utilized in this study. Originally established in 1984 with 15 states, the BRFSS now collects data from all 50 states, the District of Columbia, and three U.S. territories, gathering a diverse range of variables including demographic characteristics, lifestyle behaviors, medical histories, and clinical indicators. This dataset, comprising 319,795 records and 18 variables, serves as an invaluable resource for studying determinants of heart disease and informing predictive modeling endeavors [CDC, 2023, Pytlak, 2023].

In Table 1.1, summarizing various variables, each variable is categorized by its type and accompanied by a brief comment. Additional context not directly included in the table, such as the levels of categorical variables, is provided here. For instance, the "AgeCategory" variable represents the respondent's age in ranges of 18-24, 25-29, 30-34, 35-39, 40-44, 45-49, 50-54, 55-59, 60-64, 65-69, 70-74, 75-79, to 80 or older, encompassing various stages of adulthood and elderly demographics. Similarly, the "Race" variable signifies six distinct race/ethnicity groups: "White", "Black", "Asian", "American Indian/Alaskan Native", "Other", and "Hispanic", reflecting the diverse racial backgrounds of respondents. The "Diabetic" variable denotes

if the respondent has had a test for high blood sugar or diabetes in the past three years, with options including “Yes”, “No”, “No, borderline diabetes”, and “Yes (during pregnancy)”. “GenHealth” indicates the respondent’s general health status with levels ranging from “Very good”, “Fair”, “Good”, “Poor”, to “Excellent”, offering a spectrum of subjective health assessments. Furthermore, “PhysicalActivity” represents engagement in physical activity or exercise in the past 30 days, apart from regular job duties, with responses of “Yes” or “No”, highlighting participation in leisure-time physical pursuits [CDC, 2021].

Additionally, “PhysicalHealth” asks respondents to “Now thinking about your physical health, which includes physical illness and injury, for how many days during the past 30 days was your physical health not good?” There is no further information about what “not good” means. “MentalHealth” similarly asks respondents to consider their mental health, including stress, depression, and problems with emotions, and report the number of days during the past 30 days that their mental health was not good. “KidneyDisease” queries respondents on whether they were ever told they had kidney disease, not including kidney stones, bladder infection, or incontinence, with possible responses being “Yes” or “No” [CDC, 2021].

1.2 Organization of the Report

This report is structured into several chapters, each delving into specific facets of the research literature review, methodology, discussion, and conclusion chapters.

Variable	Type	Comment
HeartDisease	Binary	Ever diagnosed with a heart attack?
BMI	Ratio	Body Mass Index (BMI) in $\frac{kg}{m^2}$.
Smoking	Binary	Smoked 100+ cigarettes in lifetime?
AlcoholDrinking	Binary	5+ drinks for men or 4+ for women in past 30 days?
Stroke	Binary	Ever diagnosed with a stroke?
PhysicalHealth	Ratio	Days of physical health feeling unwell in the past 30 days.
MentalHealth	Ratio	Mental health days unwell in the past 30 days.
DiffWalking	Binary	Difficulty walking or climbing stairs?
Sex	Nominal	Gender(Female, Male).
AgeCategory	Ordinal	Respondent's age group.
Race	Nominal	Race/ethnicity group.
Diabetic	Nominal	Tested for high blood sugar/diabetes in past 3 years?
PhysicalActivity	Binary	Days physical health not good in past 30 days.
GenHealth	Ordinal	General health status.
SleepTime	Ratio	Average hours of sleep per day.
Asthma	Binary	Ever diagnosed with asthma?
KidneyDisease	Binary	Ever diagnosed with kidney disease?
SkinCancer	Binary	Ever diagnosed with skin cancer?

Table 1.1: Summary of Variables

Chapter 2 (Literature Review) describes neural networks, particularly multilayer perceptrons (MLPs), focusing on their architecture and training processes. It explains how learning occurs through back-propagation, which iteratively adjusts weights to minimize error. Additionally, it discusses how to address imbalanced datasets through oversampling techniques and reduce complexity and computational load through feature selection.

Chapter 3 (Methodology) encompasses data preprocessing, handling imbalanced data, evaluation metrics, and optimizing predictive modeling. Data preprocessing involves transforming features, normalizing data, and selecting relevant features. Imbalanced data is addressed using the SMOTE technique. Model performance is evaluated using the AUC-PR metric. Predictive modeling optimization entails exploring neural network architectures, applying SMOTE, and selecting features to identify an effective model.

Chapter 4 (Discussion) covers four main areas: Model and Dataset Overview, exploring neural network configurations with 1% of the dataset; SMOTE Evaluation, assessing its impact on model performance; Feature Selection using Information Gain to reduce predictors; and Model Scaling, evaluating performance across the entire dataset with detailed metrics.

Chapter 5 (Conclusion) is a summary of the key findings, contributions, and implications of the study, along with avenues for future research and practical applications.

Chapter 2

Literature Review

Neural networks (NN), or more precisely artificial neural networks (ANN), represent a significant branch of artificial intelligence [Gardner and Dorling, 1998]. The concept of artificial neural networks draws inspiration from the initial frameworks of sensory processing in the brain. Constructing an ANN involves emulating a network of simulated neurons within a computer system. By employing algorithms that emulate the processes observed in biological neurons, these networks can "learn" to solve a diverse array of problems [Krogh, 2008].

Among the various types of neural networks, multilayer perceptrons stand out prominently. Illustrated in the taxonomy provided in Figure 2.1, multilayer perceptrons offer distinct advantages. A multilayer perceptron implements without needing any prior assumptions about the data distribution. Additionally, they excel at modeling highly nonlinear functions and can be trained to generalize accurately when presented with previously unseen data. These attributes position multilayer perceptrons as compelling alternatives

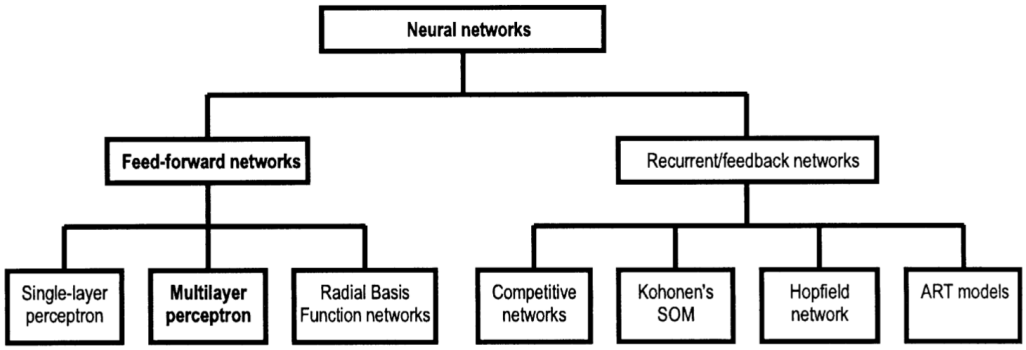

Figure 2.1: A taxonomy of neural network architectures [Gardner and Dorling, 1998].

for both numerical modeling and selecting appropriate statistical methodologies [Gardner and Dorling, 1998].

The architecture of a multilayer perceptron consists of interconnected neurons, as know as nodes, as depicted in Figure 2.2. A multilayer perceptron consists of several layers: an input layer, one or more hidden layers, and an output layer, with each layer comprising disjoint sets of nodes. The output from a node is multiplied by the connecting weight and then forwarded as input to all the nodes in the subsequent layer of the network. This directional flow of information processing characterizes the multilayer perceptron as a feed-forward neural network. The input layer does not engage in computations. This model serves to represent a nonlinear mapping between an input vector and an output vector. Interconnecting these nodes are weighted edges, which are used to determine the inputs to nodes of the next layers through a weighted summation. The input to the node is then passed to an activation function as describe below [Gardner and Dorling, 1998].

Figure 2.3 depicts the operation of a single node, which accepts input

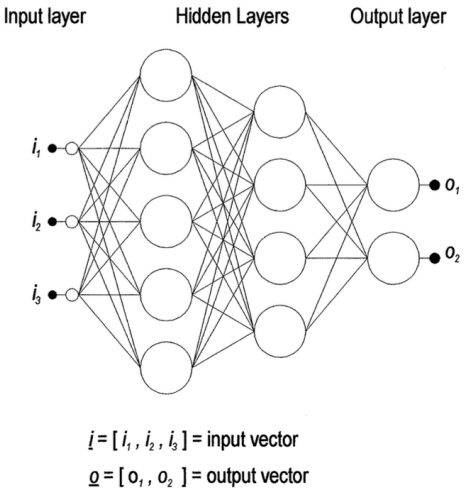

Figure 2.2: A multilayer perceptron with two hidden layers [Gardner and Dorling, 1998].

from N nodes, labeled from 1 to N. Each input, denoted as x_i, is paired with a corresponding weight, referred to as w_i. The unit's total input is determined by the sum of all weighted inputs, as shown in Equation (2.1). After that, it is sent to a function g, which represents the transfer function. Here, two kinds of transfer functions are shown. The black line represents a step function with a threshold: the output is one if the input exceeds the threshold and zero otherwise. The red curve illustrates a sigmoid function (also known as the logistic function) [Krogh, 2008].

$$\sum_{i=1}^{N} w_i x_i = w_1 x_1 + w_2 x_2 + \ldots + w_N x_N \tag{2.1}$$

The multilayer perceptron approximates nonlinear functions through the combination of numerous simple nonlinear transfer functions. These functions are organized in layers within the neural network, enabling them to

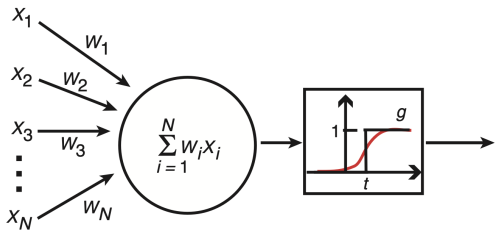

Figure 2.3: Artificial neural networks [Krogh, 2008].

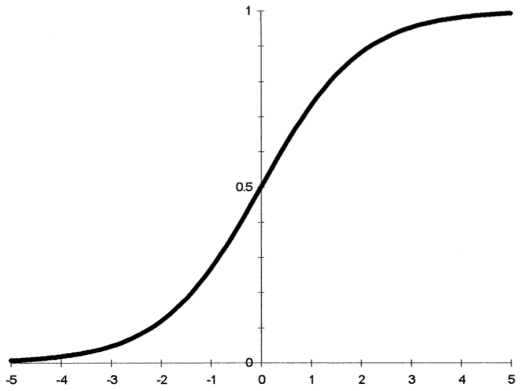

Figure 2.4: The logistic function $\frac{1}{1+e^{-x}}$ [Gardner and Dorling, 1998].

accurately model complex data patterns. Assuming the transfer function is linear, the multilayer perceptron would solely have the ability to model linear functions. Consequently, the logistic function, depicted in Figure 2.4, is often utilized not only because of its easily computed derivative, but also to approximate highly complex non-linear functions. The logistic function maps values from negative infinity to positive infinity onto the range between zero and one [Gardner and Dorling, 1998]. The calculation in a single node, shown in Figure 2.5 and demonstrated with an arbitrary numeric example in Equations 2.2 and 2.3, illustrates the process.

$$s = (0.2) \times (0.1) + (0.7) \times (0.6) + (0.4) \times (0.3) = 0.56 \tag{2.2}$$

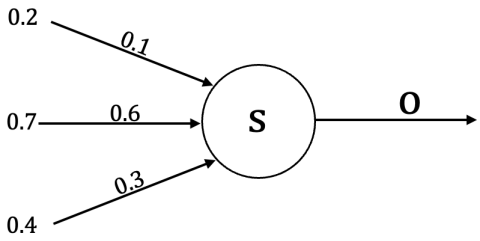

Figure 2.5: Structure of a single neuron.

$$o = \frac{1}{1+e^{-s}} = \frac{1}{1+e^{-0.56}} \approx 0.64 \tag{2.3}$$

Back-propagation is an algorithm used to adjust the weights during the training of a multilayer perceptron. This process involves iteratively presenting examples with known classifications to the network. This method is called learning or training due to its similarity to how humans learn. In a computer simulation, learning involves making small adjustments (either increases or decreases) to the weights to improve classification performance. Training begins by initializing all network weights to small random values. Each input example during training generates an output, which is then compared to the target. The goal of training is to minimize the error function, as indicated in Equation 2.4. A smaller error signifies better model performance, with the best model achieved when the error function reaches zero [Krogh, 2008, Gardner and Dorling, 1998].

The objective of back-propagation is to reduce the difference between the predicted probability and the true label. For binary classification, where the true label y is either 0 or 1 and the predicted probability $\hat{y}$ is a value between 0 and 1. Additionally, p represents the total number of training inputs in the dataset. Thus, minimizing the error function (E) of the network is defined

as follows [Günther and Fritsch, 2010, Rojas and Rojas, 1996]:

$$E = -\sum_{i=1}^{p} [y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i)] \tag{2.4}$$

Gradient descent is a numerical optimization technique employed in back-propagation to adjust the weights [Gardner and Dorling, 1998, Krogh, 2008]. Equation 2.5 defines the gradient of the error function with respect to the all l weights, where l represents the total number of weights in the neural network architecture.

$$\nabla E = \left(\frac{\partial E}{\partial w_1}, \frac{\partial E}{\partial w_2}, \ldots, \frac{\partial E}{\partial w_l} \right) \tag{2.5}$$

By employing an iterative process of gradient descent, wherein weights are updated based on the computed gradient by Equation 2.6, the network strives to minimize the error function, eventually reaching a minimum where the gradient equals zero [Rojas and Rojas, 1996]. Each weight undergoes updates through small adjustments, where γ represents the learning rate parameter, determining the step size during the iterative gradient descent learning process. Careful tuning of the learning rate is necessary to ensure effective training. If it is too large, the network error will fluctuate unpredictably because of large weight changes, potentially jumping over the global minima, while if it's too small, training will take a long time [Gardner and Dorling, 1998, Krogh, 2008].

$$\Delta w_i = -\gamma \frac{\partial E}{\partial w_i} \quad \text{for } i = 1, \ldots, l \tag{2.6}$$

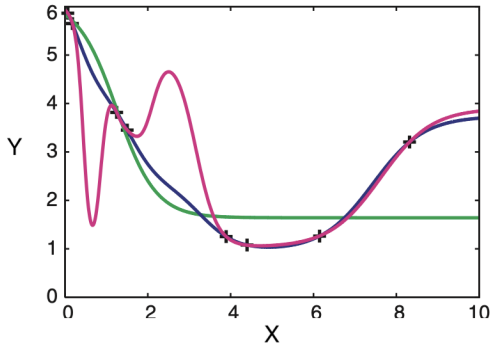

Figure 2.6: Visualizing the effects of network complexity on model fit [Krogh, 2008].

In Figure 2.6, the eight points on a parabola that are indicated by plusses and three different neural networks trained on a dataset are compared, showcasing the impact of the number of hidden units (neurons in hidden layers) on performance. While a network with one hidden unit struggles to capture the underlying function (green) illustrated by the pattern that is too simple, 20 hidden units lead to overfitting (purple) illustrated by a very complicated model, "a network with 10 hidden units (blue) approximates the underlying function remarkably well". These results emphasize the delicate balance required in network design.[Krogh, 2008]. Overfitting, defined as the model excessively fitting the training data but failing to generalize to unseen data, poses a significant challenge in neural network training [Krogh, 2008].

This paper addresses the challenge of determining the appropriate number of hidden layers and neurons in each hidden layer for neural networks, to reduce the risks of underfitting with insufficient neurons or overfitting with excessive ones [Karsoliya, 2012].

Rule-of-thumb methods suggest setting the number of neurons in each

hidden layer to be $\frac{2}{3}$ (or 70% to 90%) of the size of the input layer. Furthermore, the total number of neurons across all hidden layers should not exceed twice the number of input layer neurons. Deviation from this guideline can lead to increased complexity and total training time, potentially resulting in overfitting [Karsoliya, 2012].

For example, if there are three hidden layers, the total number of neurons in these layers would be approximately twice the number of neurons in the input layer, since $3 \times \frac{2}{3} = 2$ times the number of neurons in the input layer. Adding a fourth hidden layer would result in an excessive number of neurons, surpassing twice the number of neurons in the input layer, thus making the fourth hidden layer unnecessary. Experimental results presented in this paper indicate that maintaining near equality in the number of neurons in the first and second hidden layers contributes to efficient training. [Karsoliya, 2012].

In this paper, the dataset, comprising 13 numeric features and an indicator of heart condition, is divided into 60% for training and 40% for validation. Information gain is employed to determine feature weights for feature selection, and the ranking of features is based on descending information gain values, reflecting their informativeness about the class. Information gain for attribute X concerning the class attribute Y is defined as the reduction in uncertainty about the value of Y given the knowledge of X, denoted as $IG(Y;X)$. The formula for information gain is expressed in Equation 2.7 [Khemphila and Boonjing, 2011].

$$IG(Y;X) = H(X) + H(Y) - H(X,Y) \tag{2.7}$$

Here Y and X are discrete variables with values in $y_1, ..., y_n$ and $x_1, ..., x_n$

respectively. The entropy of Y and X, X, and Y are determined by the following equations:

$$H(X,Y) = -\sum_{i=1}^{n}\sum_{j=1}^{n} P(X=x_i, Y=y_j)\log_2(P(X=x_i, Y=y_j)) \quad (2.8)$$

$$H(X) = -\sum_{i=1}^{n} P(X=x_i)\log_2(P(X=x_i)) \quad (2.9)$$

$$H(Y) = -\sum_{j=1}^{n} P(Y=y_j)\log_2(P(Y=y_j)) \quad (2.10)$$

The entropy of Y and X, $H(X,Y)$, is calculated as the proportion of instances where both X takes on the value x_i and Y takes on the value y_j out of the total number of instances. And also the entropy of X, $H(X)$, is calculated as the proportion of instances where X takes on the value x_i out of the total number of instances. The entropy of Y, $H(Y)$, denoted by $P(Y=y_j)$, is calculated similarly [Cover, 1999].

In order to calculate information gain, it is imperative that the input be comprised of discrete numbers. However, given that the inputs in their experiment consist of continuous real numbers, we addressed this issue by discretizing the continuous-valued attributes through partitioning the range of values into a finite number of subsets [Khemphila and Boonjing, 2011].

Table 2.1 displays the ranking of information gains for 13 attributes in [Khemphila and Boonjing, 2011]. Initially, an Artificial Neural Network (ANN) was employed without incorporating any feature selection. The corresponding results, depicted in Table 2.2, indicate the accuracy in the training dataset is 88.46%, and in the validation dataset, it is 80.17%. Subsequently, the researcher sequentially removed features starting with the lowest information gain and utilized ANN for classification. If the classification accuracy

Item	Features	Information Gain
11	Blood Pressure	0.2187
1	Thal	0.217395
2	Chest Pain Type	0.204599
12	Cholesterol	0.20316
3	Number Colored Vessels	0.190442
4	Old Peak	0.167595
5	Maximum Heart Rate	0.151654
6	Induced Angina	0.14221
7	Slope	0.116834
8	Age	0.072551
9	Sex	0.059138
10	Resting ECG	0.024075
13	Fasting Blood Sugar	0.000566

Table 2.1: Information Gain for each feature from the analysis on the heart disease dataset [Khemphila and Boonjing, 2011].

Features	Training	Validation
13	88.46%	80.17%
8	89.56%	80.99%

Table 2.2: Accuracy of ANN Classifier Used Feature Selection [Khemphila and Boonjing, 2011].

matched or exceeded the accuracy without considering the feature, it remained omitted and they continued this process by eliminating the feature with the second lowest information gain, iterating until the classification accuracy fell below that achieved without information gain. Table 2.2 presents the feature numbers utilized in the experiment. Features were gradually eliminated until only 8 remained. The resulting accuracy in the training dataset was 89.56%, and in the validation dataset, it was 80.99%. This study underscores the significance of feature selection in enhancing computational efficiency. Lower healthcare costs associated with checklists, and minimize the number of attributes required from patients.[Khemphila and Boonjing, 2011].

Suppose a dataset has been partitioned into classes. The term "majority class" refers to a class within a dataset that contains a greater number of instances compared to another class. Conversely, the "minority class" pertains to the class with fewer instances within the same dataset. These datasets, characterized by such class imbalances, are termed "imbalanced datasets" [Spelmen and Porkodi, 2018].

An issue in imbalanced data becomes evident when considering an example: imagine a dataset where 90% of instances belong to the majority class, leaving only 10% for the minority class. If a classification rule predicts all

instances as the majority class, it achieves an accuracy of 90%. However, this accuracy metric fails to adequately represent classification performance, as none of the minority class instances are correctly classified. Furthermore, classifiers or classification algorithms often perceive minority class instances as noisy data, leading to their elimination. Addressing the imbalance in data is crucial for enhancing the accuracy of classification techniques and ensuring precise predictions [Spelmen and Porkodi, 2018]. Various real-world applications, including fraud detection (e.g., credit card, phone calls, insurance), medical diagnosis, network intrusion detection, fault monitoring, pollution detection, biomedical research, bioinformatics, and remote sensing (e.g., land mines, underwater mines), grapple with the challenge of imbalanced datasets [Spelmen and Porkodi, 2018].

Numerous methods have been proposed to mitigate the issue of imbalanced datasets, with this section focusing on the techniques. Figure 2.7 illustrates these methods which are categorized into three types: Data level methods, Algorithmic level methods, and Hybrid methods [Spelmen and Porkodi, 2018].

Data level methods involve preprocessing steps to balance the dataset. These methods encompass oversampling, undersampling, and feature selection techniques. Oversampling methods involve increasing minority class instances to balance class distribution. However, a common issue with oversampling is the potential for overfitting, as it does not introduce new instances or information. Conversely, undersampling methods entail removing majority class instances to achieve balance, though this approach disregards potentially valuable information contained in the deleted instances. To address these challenges, researchers have proposed heuristic approaches [Spel-

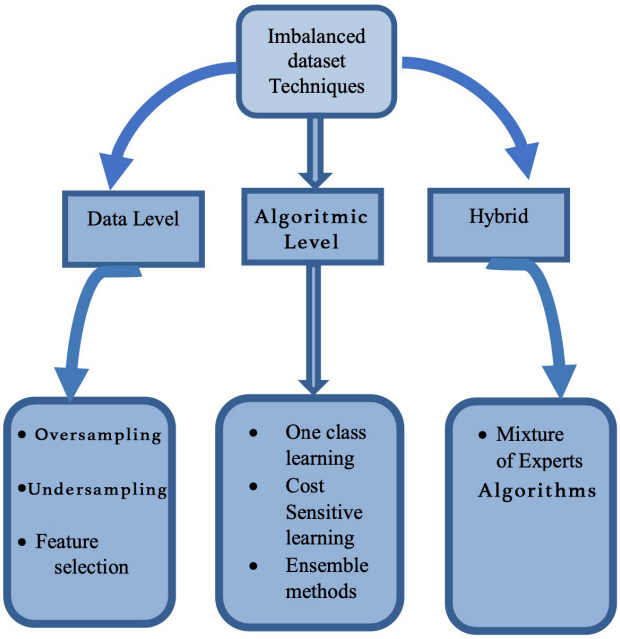

Figure 2.7: Methods to handle the imbalanced data [Spelmen and Porkodi, 2018].

men and Porkodi, 2018].

One notable contribution is the Synthetic Minority Oversampling Technique (SMOTE), which generates synthetic examples instead of replacing minority class instances [Spelmen and Porkodi, 2018]. SMOTE is an oversampling technique that generates synthetic samples from the minority class using information from the dataset. By increasing the minority class through SMOTE, a class-balanced training set is obtained. The synthetic SMOTE sample is defined in Equation 2.11 [Blagus and Lusa, 2013].

$$S = x + u \cdot (x^R - x) \tag{2.11}$$

In which:

- S: The synthetic sample.
- x: A sample from the minority class.
- x^R: A sample randomly chosen from the nearest neighbors of x in the minority class.
- u: A random number between zero and one.

The Precision-Recall Curve (PRC) plot (illustrated in Figure 2.8) is an essential tool for assessing the performance of classifiers, especially when class distributions differ. The PRC plot demonstrates the tradeoff between precision and recall, and it includes a single performance metric known as the Area Under the PRC Curve (AUC-PR) score. The AUC-PR scores are convenient for comparing the performance of classifiers [Saito and Rehmsmeier, 2015].

In PRC plots, classifiers with random performance are represented by a straight horizontal line, which can be defined as the baseline of the PRC. This baseline dynamically adjusts based on the ratio of positive (P) to negative (N) instances in the dataset. The baseline's position is given in Equation 2.12, where y indicates the baseline level and P and N are the numbers of positive and negative instances, respectively. For example, in a dataset with balanced class distribution, the baseline is at $y = 0.5$, while in an imbalanced datasets with a $P : N$ ratio of 1:9, the baseline moves to $y = 0.1$. The dynamic nature of the PRC plot highlights its usefulness in evaluating classifiers across various class distribution scenarios and enables meaningful comparisons among different classifiers. The baseline reveals the degree of dataset imbalance: a value of 0.5 denotes a balanced dataset, values below

0.5 signify a majority of negative instances, and values above 0.5 indicate a minority class of negative instances. [Saito and Rehmsmeier, 2015].

In Figure 2.8, the green line represents the AUC-PR for a dataset with a baseline of 0.5, while the red line represents the PRC plot for a dataset with a baseline of 0.1. The AUC-PR values are roughly the same, at 0.76 and 0.78 for the red and green lines, respectively. In this case, the model represented by the red line demonstrates better performance because the difference between its baseline and AUC-PR is higher than that of the green model.

$$y = \frac{P}{P + N} \tag{2.12}$$

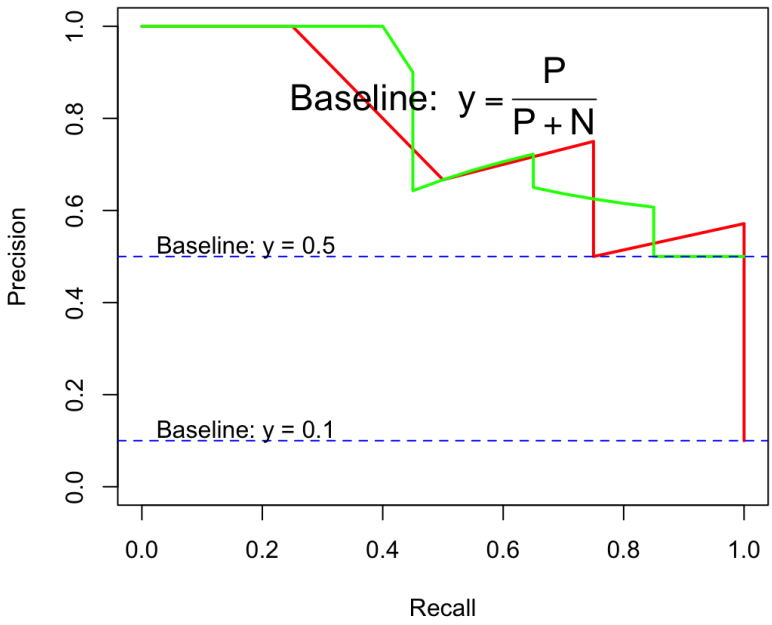

Figure 2.8: Baseline Sensitivity in PRC Plot: Class Ratio Impact with 1:1 ratio and 9:1 ratio of P:N.

Chapter 3

Methodology

The methodology employed in this project is designed to investigate the factors influencing the occurrence of heart disease and develop an accurate predictive model using a Multi-Layer Perceptron (MLP), also known as a Feed Forward Neural Network [Gardner and Dorling, 1998, FFNN]. This methodology is comprised of several stages, including data preprocessing, handling an imbalanced dataset, evaluation and optimizing predictive modeling.

3.1 Data Preprocessing

The Data Preprocessing phase, focus on ensuring the dataset is in a compatible format for model building, which involves several crucial steps.

3.1.1 Feature Transformation

In statistical modelling and regression analysis, dummy variables play a crucial role in representing categorical data numerically. They address the challenge of transforming non-numeric categorical attributes into binary values (0 or 1). When dealing with binary categories, a single dummy variable is employed. However, for variables with more than two categories, multiple dummy variables are created, with the number of dummies equal to the number of categories minus one. This approach allows for the effective inclusion of categorical information in regression models, facilitating the interpretation of relationships between different categories and the dependent variable [Faraway, 2016].

3.1.2 Normalization

Normalization of input data in artificial neural networks is as crucial as other preliminary data processes. It involves scaling data into a consistent range, thereby reducing bias and accelerating the learning process [Jo, 2019].

The normalization of input data, ensuring all values fall within the [0, 1] range, helps prevent bias toward certain inputs, with the normalized value calculated using Equation 3.1 [Patel and Joshi, 2013].

$$\text{Normalized Value} = \frac{X - \text{Min}(X)}{\text{Max}(X) - \text{Min}(X)} \tag{3.1}$$

3.1.3 Feature Selection

In the process of feature selection, the criterion employed is the concept of information gain (IG), which serves as the basis for attribute selection. Information gain is commonly utilized for the selection of feature sets, and the method implemented involves the calculation of IG for each attribute.

3.1.4 Training-Test Split

Before constructing models, the dataset is randomly divided into two distinct subsets: 60% is allocated for the training set and the remaining 40% designated for the test set [Khemphila and Boonjing, 2011].

3.2 Handling Imbalanced Data

In addressing the challenge of class imbalance in datasets, the Synthetic Minority Over-sampling Technique (SMOTE) has emerged as a vital solution. This technique, which involves generating synthetic samples from the minority class to achieve a more balanced dataset, operates by creating linear combinations of similar minority class samples. The research has shown that SMOTE consistently excels over other methods in mitigating class imbalance issues, a finding of particular relevance in areas such as medical diagnosis, fraudulent call detection, and telecommunications. Notably, the effectiveness of SMOTE varies across different fields and depends on factors like the dimensionality of the data and the type of classifiers used. For instance, in high-dimensional settings, the impact of SMOTE on classifiers becomes more

complex and demands careful implementation, including considerations for variable selection. These nuances highlight the importance of context-specific adaptations when applying SMOTE in diverse domains such as bioinformatics and other real-world applications [Spelmen and Porkodi, 2018, Blagus and Lusa, 2013].

Addressing imbalanced data is a pivotal aspect of this project, given the uneven distribution of "Yes" and "No" instances in the binary dependent variable "HeartDisease".

3.3 Evaluation Metrics

The effectiveness of the proposed classification models is evaluated using confusion matrix and accuracy [Desai et al., 2019]. The Area Under the Precision-Recall curve (AUC-PR) is a metric that evaluates a binary classification model's performance, particularly useful in imbalanced datasets or tasks where correctly identifying positive instances is crucial [Davis and Goadrich, 2006].

A confusion matrix, as shown in Table 3.1, is a tool which is used to assess the performance of a classification algorithm. "This matrix has four categories: True positives (TP) are examples correctly labeled as positives. False positives (FP) refer to negative examples incorrectly labeled as positive. True negatives (TN) correspond to negatives correctly labeled as negative. Finally, false negatives (FN) refer to positive examples incorrectly labeled as negative." Each cell plays a crucial role in illustrating how well the model has predicted the actual classifications [Davis and Goadrich, 2006]. The

	Predicted Positive	Predicted Negative
Actual Positive	True Positive (TP)	False Negative (FN)
Actual Negative	False Positive (FP)	True Negative (TN)

Table 3.1: Confusion Matrix

confusion matrices show in results use a fixed threshold of 0.5.

The accuracy is calculated using elements from the confusion matrix (TP, TN, FP, and FN), as demonstrated in Equation 3.2 [Khemphila and Boonjing, 2011].

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{3.2}$$

Precision: Precision represents the ratio of true positive predictions out of all positive predictions made by the model, which is a measure of how accurate the positive predictions are [Davis and Goadrich, 2006].

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{3.3}$$

Recall (Sensitivity): Recall is the ratio of true positive predictions to the total number of actual positive instances. It is a measure of the model's ability to capture all positive instances [Davis and Goadrich, 2006].

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{3.4}$$

The Area Under the Precision-Recall curve (AUC-PR) is a metric that evaluates a binary classification model's performance. It measures the area under the precision-recall curve, summarizing the trade-off between precision (accuracy of positive predictions) and recall (sensitivity to positive instances).

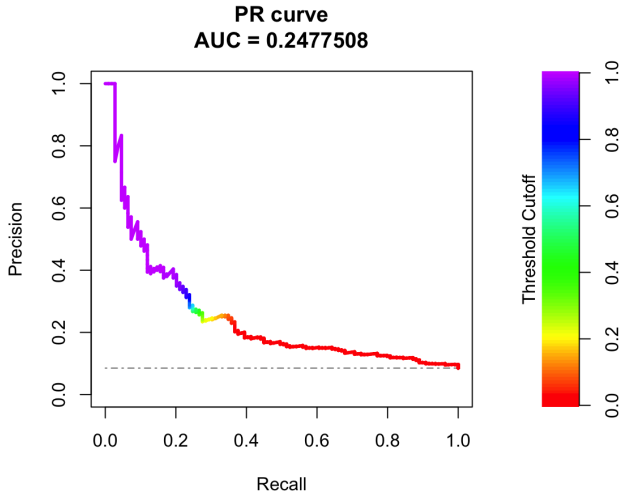

Figure 3.1: Precision-Recall: AUC-PR Analysis.

AUC-PR ranges between zero and one, with a higher value indicating better model performance [Davis and Goadrich, 2006].

The AUC-PR is derived from the PRC plot shown in Figure 3.1. The PRC plot includes the X-axis representing Recall, the Y-axis representing Precision, and the grey dashed line indicating the baseline. The legend on the right side of the PRC plot shows the colour scheme for the score thresholds, with the curve plotted in corresponding colours. The AUC-PR value is displayed at the top of the plot [Grau and Keilwagen, 2018].

The activation function in the output layer is sigmoid, so the output can be any number between zero and one. To calculate precision and recall, the output needs to be mapped to either zero or one to find the predicted values, which are then compared with the actual values to calculate TP, TN, FP, and FN. A threshold is used for this mapping, and it has the same range as the model output. For each threshold, precision and recall are calculated, determining a point on the plot. By connecting these points, the PRC plot

is created, and the precision at the end point always equals the baseline while recall equals one [Saito and Rehmsmeier, 2017]. The area under curve can be calculated different ways, for example, [Saito and Rehmsmeier, 2017] provide an example with trapezoids. However, the code that we will employ uses non-linear interpolation which is not fully describe in the documentation [Grau and Keilwagen, 2018].

3.4 Optimizing Predictive Modeling

This study employed the neuralnet package in the R Programming Language to conduct an examination, aimed at identifying the optimal model for our dataset. It involved an exploration of the potential benefits of implementing the Synthetic Minority Over-sampling Technique (SMOTE) algorithm. Additionally, the study tried to figure out the efficacy of Information Gain (IG) for feature selection, in order to decrease the model's complexity by reducing the number of predictors. In light of the expansive dataset comprising 319,795 entries, only 1% of the dataset was used in the initial phase of training, facilitating the development of 258 distinct models. These models are diverse in their complexity, varying from one to three hidden layers. The variation in the number of neurons in each layer is guided by specific recommendations: the number of neurons in each hidden layer was advised to be within the range of 70% to 90% of the size of the input layer. Additionally, it was recommended that the aggregate number of neurons across all hidden layers should not surpass twice the number of neurons in the input layer [Karsoliya, 2012].

Following the process of feature transformation, the number of neurons

in the input layer was 23. Consequently, this meant that the number of neurons in the hidden layers should range between 16 and 21. Moreover, it was recommanded that the total count of neurons in all hidden layers did not exceed 46, which is twice the number of input layer neurons [Karsoliya, 2012].

For a single hidden layer, varying the number of neurons between 16 and 21 resulted in 6 models. For two hidden layers, varying the number of neurons between 16 and 21 in both layers resulted in 36 models. For three hidden layers, varying the number of neurons between 16 and 21 in the first two layers and between 4 and 14 in the third layer (while ensuring the total did not exceed 46 neurons by removing excess neurons from third layer) resulted in 222 models. Therefore, a total of 258 models were created.

Each of these model structures was subjected to ten iterations of training. The selection of the optimal model was based on its performance, particularly measured by the highest AUC-PR. This criterion ensured that the model with the most effective predictive accuracy was chosen.

Subsequently, the optimal model selected in the previous phase, equipped with its initial weights, was employed to train the oversampled training dataset using the SMOTE algorithm, ensuring balanced representation without any manipulation of the test dataset. This step was crucial in evaluating the model's robustness and adaptability to a more evenly distributed dataset. Additionally, among the 17 predictors, when training the (optimal 16-18-0) model with all 17 predictors, we can initialize the weights obtained in the first phase. However, for models trained with 1 to 16 predictors, we cannot utilize the same initial weights due to the varying number of input neurons. To ensure a fair comparison, we iterated the training process 10 times across

each configuration and selected the best performance for comparison. This approach aimed to accurately determine the optimal number of predictors while maintaining consistency in model evaluation.

In the final phase of our methodology, after establishing the effectiveness of both the SMOTE algorithm and feature selection, these strategies are applied to a larger subset, encompassing 60% of the original dataset as training dataset. This step was fundamental in validating our findings and ensuring the scalability of our model under more data conditions.

Chapter 4

Discussion

4.1 Model and Dataset Overview

This study involved an exploration of various neural network configurations to determine the most effective model for predicting heart disease. In order to manage computational constraints and expedite the training process, our study deliberately restricted the scope of the training dataset to a mere 1% of the total dataset, amounting to 3199 instances. Consequently, the dataset was partitioned into a training set comprising 1,920 instances (60%) and a separate test set consisting of 1279 instances (40%). While this approach markedly reduced the duration of model training to approximately 24 hours across all 258 combinations.

This process evaluated 258 distinct combinations of neurons distributed across three hidden layers, with each configuration undergoing ten iterations of training. The result of this thorough exploration led to the identification

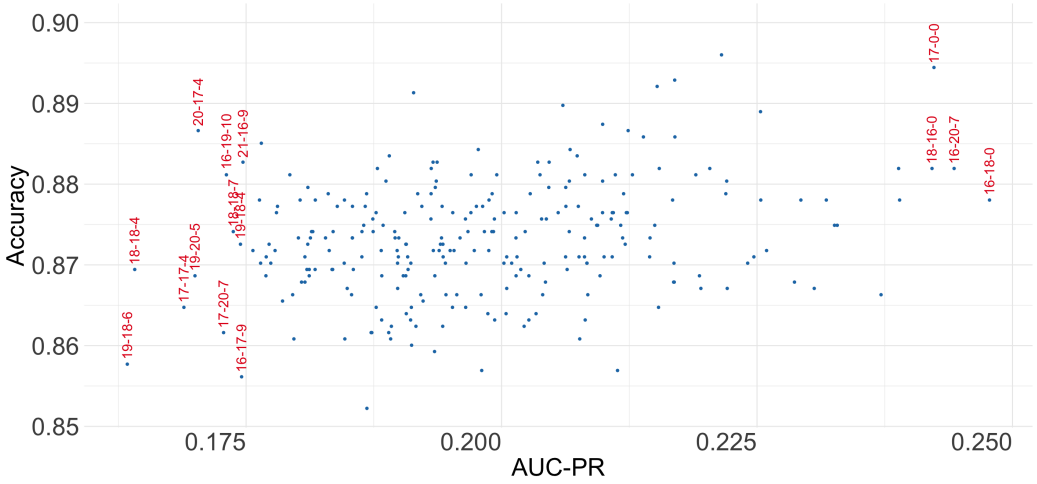

Figure 4.1: Top 258 Neural Network Models by AUC-PR.

of an optimal neural network model, graphically depicted in Figure 4.1.

In Figure 4.1, each dot represents a neural network configuration, specifying the number of neurons in each hidden layer along with corresponding accuracy and Area Under the Precision-Recall Curve (AUC-PR) metrics. We clearly note that accuracy is not strongly related to AUC-PR therefore assesses the result differently. As we describe in Chapter 3 (Methodology) we will be focusing in AUC-PR rather than accuracy. The 16-18-0 architecture was chosen as optimal due to its highest AUC-PR.

Some structures demonstrated competitive performance, though with varying levels of complexity. For instance, configurations such as 17-0-0, 18-16-0, and 16-20-7 showcased noteworthy performance compared to 16-18-0. This suggests that single-layer architectures or those with fewer hidden layers may also merit consideration, particularly in scenarios where computational resources are limited.

Furthermore, all models with an AUC-PR lower than 0.175 have three

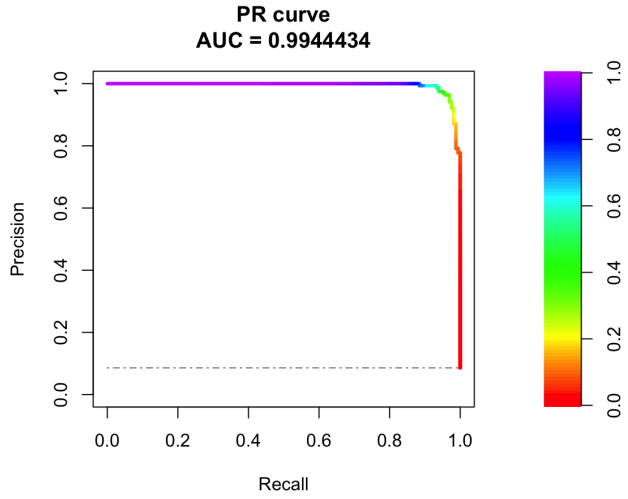

Figure 4.2: PRC Plot on 1% Dataset and 17 Predictors for training data (16-18-0 architecture).

hidden layers, indicating that increased complexity in a model does not lead to better performance. For example, architectures such as 19-18-6 and 18-18-4 exhibit the lowest performance. Figure 4.1 shows the trade-offs between model complexity and predictive efficacy.

The selected model achieved an AUC-PR of 0.944, as shown in Figure 4.2, and an accuracy rate of 99.32% on the training dataset. Similarly, for the test dataset, the AUC-PR is 0.248, as shown in Figure 4.3, with an accuracy of 87.80%. The high AUC-PR on the training dataset and low AUC-PR on the test dataset indicate overfitting. However, it is essential to emphasize that the aim is to find the best model with a high AUC-PR on the test dataset. Additionally, these results were derived from a significantly reduced dataset.

Confusion matrices provide a depiction of the model's classification performance by detailing the counts of true positive, true negative, false positive,

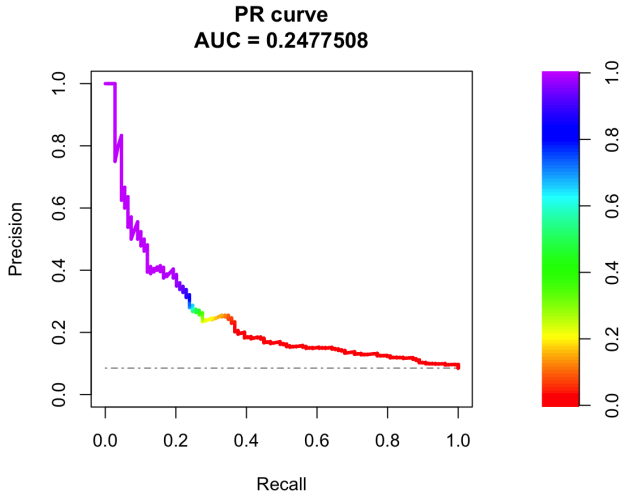

Figure 4.3: PRC Plot on 1% Dataset and 17 predictors for Test Data (16-18-0 architecture).

and false negative predictions. In Table A.1, the confusion matrix for the training dataset and Table A.2 for the test dataset.

Figure 4.4 presents a box plot illustrating the relationship between the number of neurons and AUC-PR values. Different architectures can comprise the same number of neurons. For example, 16-18-0, 17-17, 18-16 architectures are represented in the box for 34 neurons including all 10 iteration of each. The highest AUC-PR value is achieved with a network of 34 neurons (16-18-0 architecture). The second and third highest AUC-PR values are observed with networks of 43 and 17 neurons, respectively. In contrast, the lowest AUC-PR value is seen with 45 neurons. Moreover, the highest average AUC-PR value occurs with 20 neurons. This highlights the sensitivity of model performance to architecture choices and emphasizes the need for careful consideration and experimentation in neural network design.

Figure 4.5 presents the impact of varying the number of hidden layers in

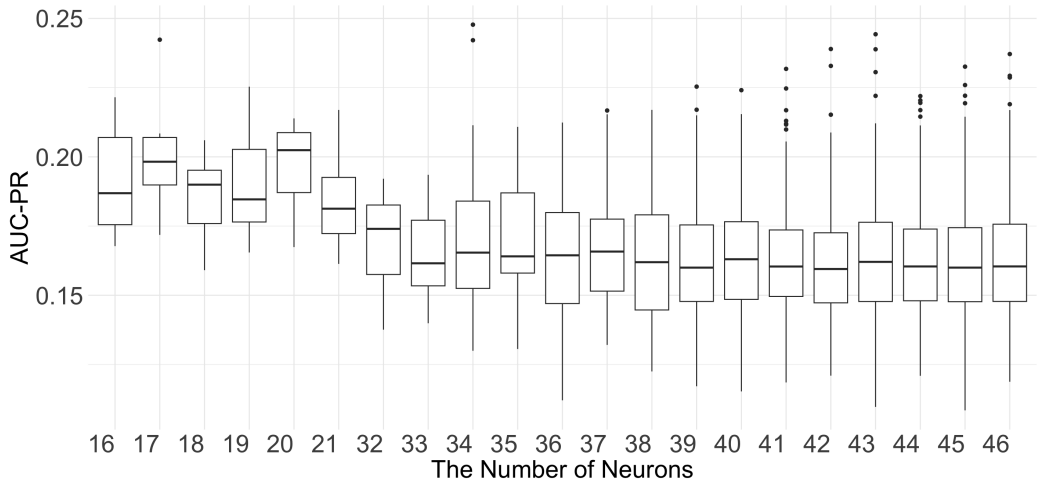

Figure 4.4: The Number of Neurons' Impact on AUC-PR.

a neural network on its performance, measured by AUC-PR. Three configurations were tested: networks with 1, 2, and 3 hidden layers. The results show that the AUC-PR fluctuates across these configurations. The network with 2 hidden layers achieved the highest AUC-PR, indicating optimal performance. Conversely, the network with 3 hidden layers yielded the lowest AUC-PR. The mean AUC-PR values suggest diminished performance with increased complexity. These findings underscore the importance of balancing model complexity to optimize performance in binary classification tasks.

4.2 SMOTE Evaluation

To address the imbalanced dataset, the Synthetic Minority Over-sampling Technique (SMOTE) was implemented on training dataset. SMOTE endeavors to synthetically augment the minority class, thereby striving for a more balanced dataset. The initial distribution of the training dataset, where "No" constitutes 91.44%, while "Yes" accounts for only 8.56%.

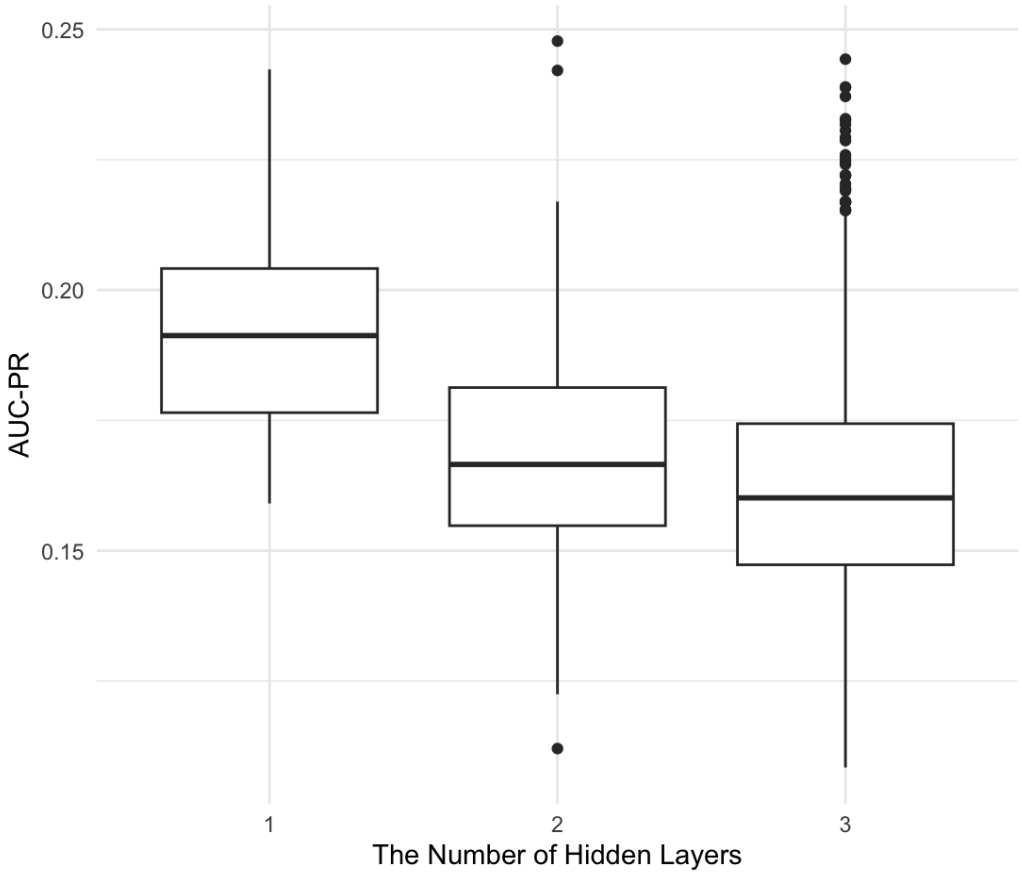

Figure 4.5: The Number of Hidden Layers' Impact on AUC-PR.

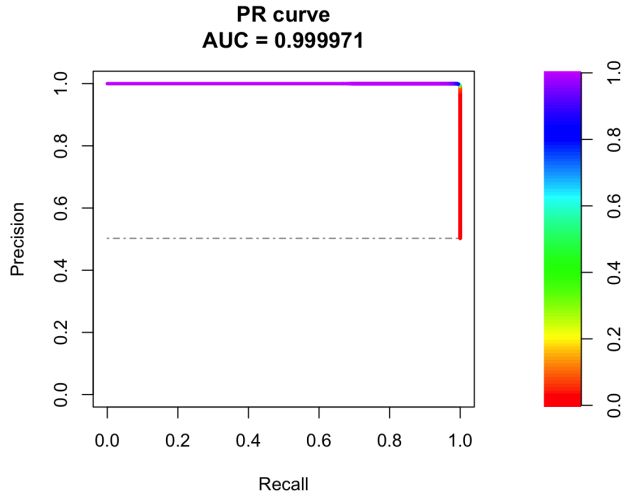

Figure 4.6: PRC Plot on 1% Dataset and 17 Predictors for training data (16-18-0 architecture) by using SMOTE.

Following the application of SMOTE, the distribution of the dataset underwent a significant transformation. Consequently, the modified training dataset distribution post-SMOTE application displayed a closer balance, with "No" representing 49.74% and "Yes" 50.26% of the dataset.

Table A.3 presents the confusion matrix for the training dataset, revealing distinct trends in predictive performance compared to the test dataset. The accuracy is 99.63%, with an AUC-PR of 0.999 as shown in Figure 4.7. The high AUC-PR on the training dataset and low AUC-PR on the test dataset indicate overfitting. However, it is essential to emphasize that the aim is to find the best model with a high AUC-PR on the test dataset. Despite the highly favourable evaluation metrics observed for the training dataset, the test dataset exhibits contrasting results.

Contrary to the anticipated outcome, the application of SMOTE did not yield an improvement in the model's performance. The metrics post-

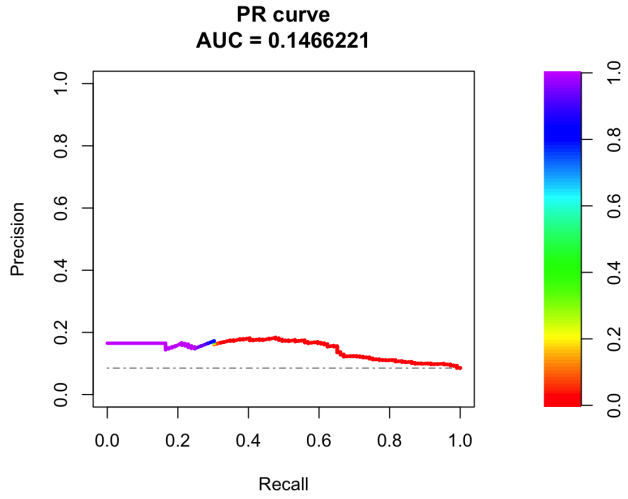

Figure 4.7: PRC Plot on 1% Dataset and 17 Predictors for Test Data (16-18-0 architecture) by using SMOTE.

SMOTE application, detailed in Figure 4.7, reveal a decreased AUC-PR of 0.147 and an accuracy of 81.47% for the test dataset. Table A.4 illustrates the confusion matrix for the test dataset, providing insight into the model's predictive performance after SMOTE application.

Based on the observed decrease in model performance, SMOTE does not appear to be beneficial for improving the performance of this dataset.

4.3 Feature Selection via IG

In the third phase of our study, we focused on incorporating Information Gain (IG) as a feature selection technique to less complexity our model for predicting heart disease. The objective was to identify and retain the most informative predictors, thus simplifying the model while maintaining its pre-

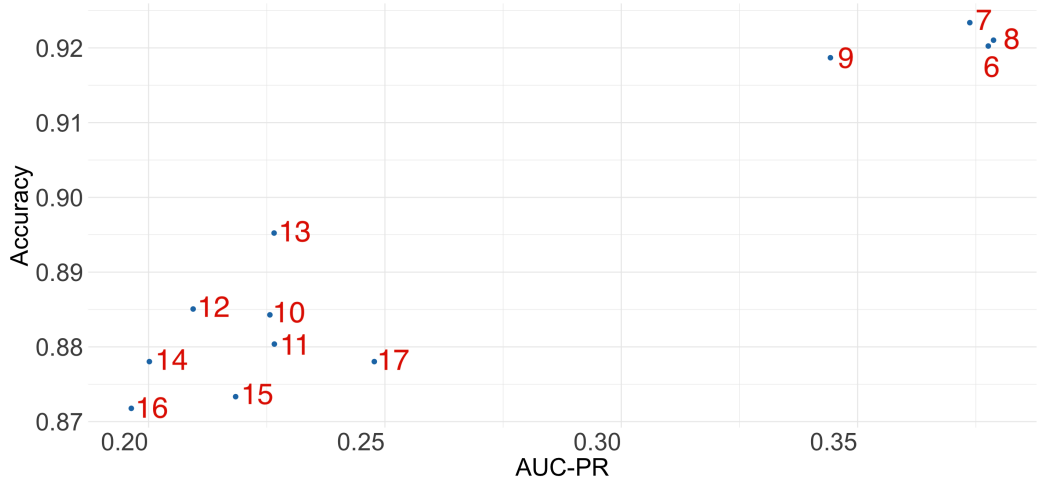

Figure 4.8: Impact of Predictor Count on Evaluation Metrics.

dictive performance. Table 4.1 presents the IG values for each predictor, revealing the degree of informativeness associated with each feature.

Analysis of the IG values highlighted that a significant number of predictors were crucial for maintaining the accuracy and reliability of the model. Notably, utilizing 8 predictors resulted in the highest AUC-PR, indicating optimal model performance.

Figure 4.8 illustrates the relationship between the number of features used and the corresponding AUC-PR values on the test dataset. It demonstrates that the AUC-PR values plummet to zero when the number of features is five or fewer, emphasizing the importance of feature selection in maintaining predictive performance.

Moreover, in Figure 4.9 and Figure 4.10, AUC-PR plots for the training and test datasets respectively are depicted. Notably, the AUC-PR value for the training dataset is at 0.365 when utilizing eight predictors, corroborating the findings from Table 4.1. Similarly, the AUC-PR value for the test dataset

Variable	**Information Gain Value**
AgeCategory	0.045999
GenHealth	0.040552
DiffWalking	0.022499
Diabetic	0.019223
PhysicalHealth	0.017188
Stroke	0.017180
KidneyDisease	0.010097
Smoking	0.008223
PhysicalActivity	0.006518
SkinCancer	0.005171
SleepTime	0.004594
Sex	0.003543
BMI	0.002294
Race	0.002205
MentalHealth	0.002143
Asthma	0.001146
AlcoholDrinking	0.000844

Table 4.1: Information Gain Values

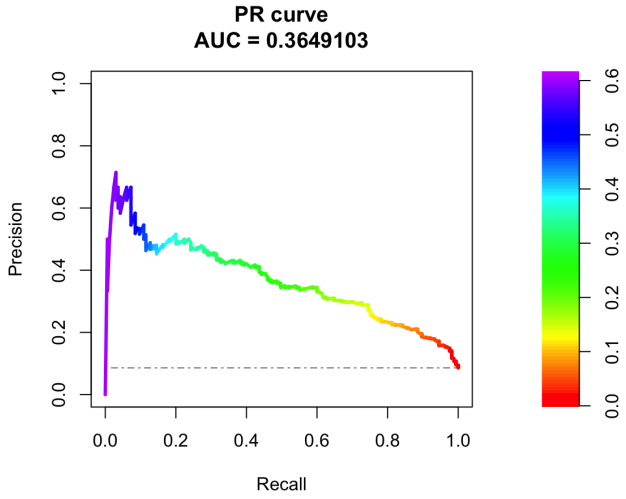

Figure 4.9: PRC Plot on 1% Dataset and 8 Predictors for training data (16-18-0 architecture).

reaches 0.379 under the same feature selection configuration.

Furthermore, Tables A.6 and A.5 present the confusion matrices for the test and training datasets, respectively, offering a detailed breakdown of predicted positive and negative instances compared to actual outcomes.

4.4 Model Scaling

To evaluate the performance of our model, we conducted an analysis using the entire dataset, yielding insightful metrics. Figure 4.11 shows the corresponding Area Under the Precision-Recall Curve (AUC-PR) for the training dataset is 0.344. And also the AUC-PR for the test dataset is 0.321 in Figure 4.12. Tables A.7 and A.8 presents the confusion matrix for the training and test dataset, respectively.

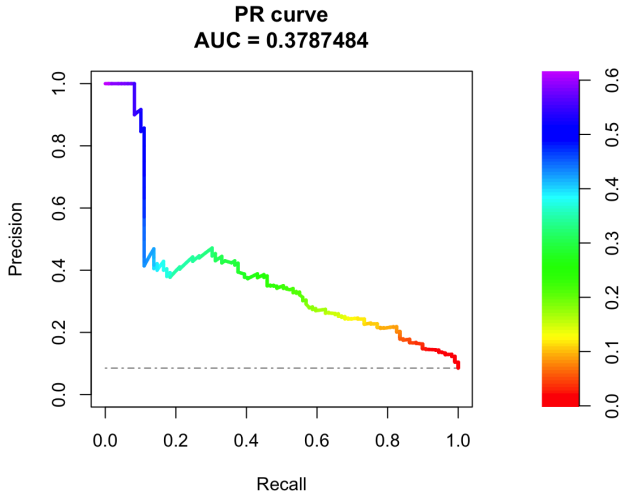

Figure 4.10: PRC Plot on 1% Dataset and 8 Predictors for Test Data (16-18-0 architecture).

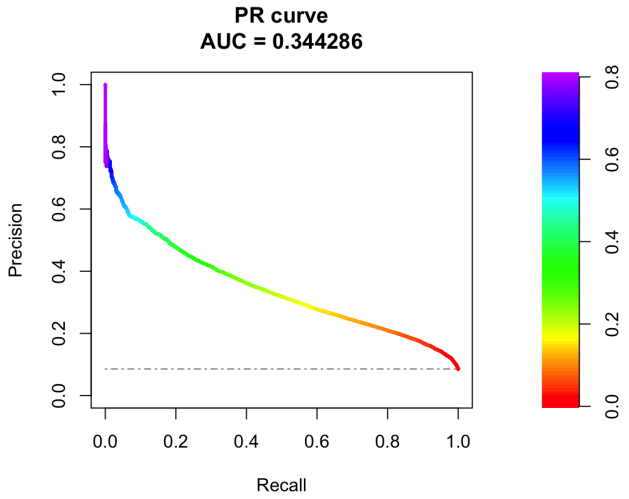

Figure 4.11: PRC Plot on Entire Dataset and 8 Predictors for training data (16-18-0 architecture).

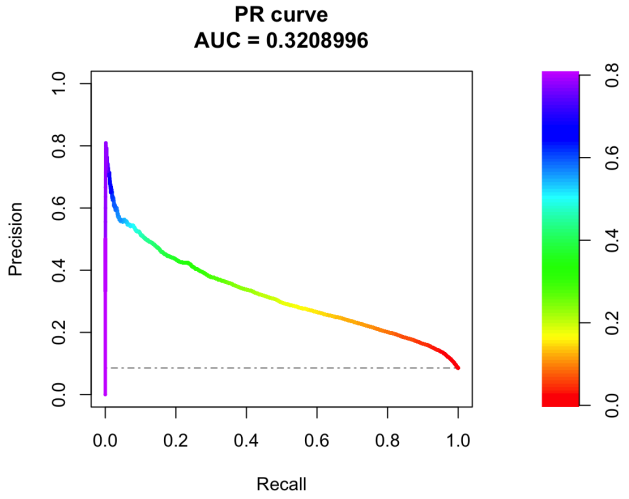

Figure 4.12: PRC Plot on Entire Dataset and 8 Predictors for Test Data (16-18-0 architecture).

Chapter 5

Conclusion

This research aims to enhance the prediction of heart disease using machine learning techniques, focusing on neural network models trained on data from the CDC's Behavioral Risk Factor Surveillance System (BRFSS). It involves a systematic exploration of model architectures, feature selection methods, and techniques for handling imbalanced data.

The evaluation of various neural network configurations highlighted the significance of model complexity and architecture in achieving optimal predictive performance. While the chosen 16-18-0 architecture demonstrated promising accuracy and AUC-PR metrics, other configurations showcased competitive performance, underscoring the importance of thorough exploration.

The utilization of imbalanced datasets poses challenges for predictive modeling, particularly in the context of heart disease prediction. The application of the Synthetic Minority Over-sampling Technique (SMOTE) aimed

to address data imbalance but resulted in unexpected performance degradation.

Feature selection proved instrumental in streamlining model complexity while preserving predictive accuracy. The identification of informative predictors through Information Gain analysis facilitated the construction of more efficient models, enhancing understanding of the underlying factors contributing to heart disease prediction.

The AUC-PR (Area Under the Precision-Recall curve) being low at 0.321 indicates that the model's precision-recall performance is not particularly strong. In this case, the baseline is defined $y = \frac{P}{P+N}$ and is calculated as $y = 0.0856$. This indicates that the dataset is highly imbalanced, with a much larger number of negative instances compared to positive instances. Even though the AUC-PR is low, it's still considered promising because it's higher than the random baseline, which is at $y = 0.0856$. This suggests that the model is performing better than a random guess and might be capturing some signal from the data.

In imbalanced scenarios, achieving a high AUC-PR is inherently challenging due to the precision-recall curve's high sensitivity to the number of True Positives when the minority class is the positive class.

The evaluation metrics presented in Table 5.1 show the performance of different approaches utilized in this study for predicting heart disease. Initially, training the model on only 1% of the dataset yielded promising results, with a test AUC-PR of 0.248 and an accuracy of 87.80%. However, used 1% training dataset and SMOTE attempts to address dataset imbalance through SMOTE resulted in a notable decrease in performance, underscoring the im-

Approach	**AUC-PR$_{\text{Test}}$**	**AUC-PR$_{\text{Training}}$**	**Acc$_{\text{Test}}$**	**Acc$_{\text{Training}}$**
1% Train	0.248	0.994	87.80%	99.32%
1% Train, SMOTE	0.147	0.999	81.47%	99.63%
1% Train, IG	0.379	0.365	92.10%	91.56%
Entire Train, IG	0.321	0.344	91.55%	91.62%

Table 5.1: Evaluation Metrics on Different Approaches.

portance of carefully selecting and evaluating data preprocessing techniques. Then applied, Feature selection via Information Gain proved to be a crucial step in improving model performance, as evidenced by the significant enhancement in AUC-PR to 0.379 and accuracy to 92.10% when utilizing only 8 predictors. Finally, the model was trained on the entire dataset using the selected 8 predictors. The model achieved a AUC-PR of 0.321 with 8 predictors, indicating robustness across the entire dataset, with an accuracy of 91.55%.

5.1 Future Work

Accurately detecting whether a person has heart disease is crucial, minimizing errors where a person with heart disease is incorrectly identified as healthy. Therefore, high True Positive rates and low False Negative rates should be achieved by the model, prioritizing a high Recall ($Recall = \frac{TP}{TP+FN}$). Two key areas need to be explored when considering future research directions.

Firstly, the Oversampling Data Level technique was employed in this

study. Additionally, exploring techniques such as ensemble methods like bagging or boosting, and hybrid approaches combining different classifiers, might yield superior results.

Secondly, exploring the use of different non-linear transfer functions while retaining the sigmoid function in the output layer. This approach aims to assess how variations in transfer functions such as Tanh, ReLU, and Leaky ReLU affect model performance, deepening our understanding of their influence on predictions.

Bibliography

Matt W Gardner and SR Dorling. Artificial neural networks (the multilayer perceptron)—a review of applications in the atmospheric sciences. *Atmospheric environment*, 32(14-15):2627–2636, 1998.

Anders Krogh. What are artificial neural networks? *Nature biotechnology*, 26(2):195–197, 2008.

Vimalraj S Spelmen and R Porkodi. A review on handling imbalanced data. In *2018 international conference on current trends towards converging technologies (ICCTCT)*, pages 1–11. IEEE, 2018.

Anchana Khemphila and Veera Boonjing. Heart disease classification using neural network and feature selection. pages 406–409, 2011.

World Health Organization: WHO. Cardiovascular diseases (CVDs). 2021. Accessed on May 28, 2024. https://www.who.int/news-room/fact-sheets/detail/cardiovascular-diseases-(cvds).

Mert Ozcan and Serhat Peker. A classification and regression tree algorithm for heart disease modeling and prediction. *Healthcare Analytics*, 3:100130, 2023.

CDC. Know your risk for heart disease. 2023. Accessed on Oct 11, 2023. https://www.cdc.gov/heartdisease/risk_factors.htm.

Kamil Pytlak. Indicators of heart disease (2022 update). 2023. Accessed on Oct 11, 2023. https://www.kaggle.com/datasets/kamilpytlak/personal-key-indicators-of-heart-disease.

CDC. Llcp 2020 codebook report. 2021. Accessed on May 28, 2024. https://www.cdc.gov/brfss/annual_data/2020/pdf/codebook20_llcp-v2-508.pdf.

Frauke Günther and Stefan Fritsch. Neuralnet: training of neural networks. *R J.*, 2(1):30, 2010.

Raul Rojas and Raúl Rojas. The backpropagation algorithm. *Neural networks: a systematic introduction*, pages 149–182, 1996.

Saurabh Karsoliya. Approximating number of hidden layer neurons in multiple hidden layer bpnn architecture. *International Journal of Engineering Trends and Technology*, 3(6):714–717, 2012.

Thomas M Cover. *Elements of information theory*. John Wiley & Sons, 1999.

Rok Blagus and Lara Lusa. Smote for high-dimensional class-imbalanced data. *BMC bioinformatics*, 14:1–16, 2013.

Takaya Saito and Marc Rehmsmeier. The precision-recall plot is more informative than the roc plot when evaluating binary classifiers on imbalanced datasets. *PloS one*, 10(3):e0118432, 2015.

Julian J Faraway. *Extending the linear model with R: generalized linear, mixed effects and nonparametric regression models*. CRC press, 2016.

Jun-Mo Jo. Effectiveness of normalization pre-processing of big data to the machine learning performance. *The Journal of the Korea institute of electronic communication sciences*, 14(3):547–552, 2019.

Ankeeta R Patel and Maulin M Joshi. Heart diseases diagnosis using neural network. In *2013 Fourth International Conference on Computing, Communications and Networking Technologies (ICCCNT)*, pages 1–5. IEEE, 2013.

Shrinivas D Desai, Shantala Giraddi, Prashant Narayankar, Neha R Pudakalakatti, and Shreya Sulegaon. Back-propagation neural network versus logistic regression in heart disease classification. pages 133–144, 2019.

Jesse Davis and Mark Goadrich. The relationship between precision-recall and roc curves. pages 233–240, 2006.

Jan Grau and Jens Keilwagen. Package 'prroc'. 2018. Accessed on Jun 18, 2024. https://cran.r-project.org/web/packages/PRROC/PRROC.pdf.

Takaya Saito and Marc Rehmsmeier. Classifier evaluation with imbalanced datasets. "introduction to the precision-recall plot". 2017. Accessed on Jun 19, 2024. https://classeval.wordpress.com/introduction/introduction-to-the-precision-recall-plot/.

Appendix A

	Predicted Positive	Predicted Negative
Actual Positive	155	10
Actual Negative	3	1,752

Table A.1: Confusion Matrix on 1% Dataset and 17 predictors for training data (16-18-0 architecture).

	Predicted Positive	Predicted Negative
Actual Positive	27	82
Actual Negative	74	1096

Table A.2: Confusion Matrix on 1% Dataset and 17 Predictors for Test Data (16-18-0 architecture).

	Predicted Positive	Predicted Negative
Actual Positive	1,644	6
Actual Negative	6	1627

Table A.3: Confusion Matrix on 1% Dataset and 17 Predictors for training data (16-18-0 architecture) by using SMOTE.

	Predicted Positive	Predicted Negative
Actual Positive	33	76
Actual Negative	161	1009

Table A.4: Confusion Matrix on 1% Dataset and 17 Predictors for Test Data (16-18-0 architecture) by using SMOTE.

	Predicted Positive	Predicted Negative
Actual Positive	14	151
Actual Negative	11	1,744

Table A.5: Confusion Matrix for 1% Dataset and 8 Predictors for training data (16-18-0 architecture).

	Predicted Positive	Predicted Negative
Actual Positive	12	97
Actual Negative	4	1,166

Table A.6: Confusion Matrix for 1% Dataset and 8 Predictors for Test Data (16-18-0 architecture).

	Predicted Positive	Predicted Negative
Actual Positive	1,380	15,044
Actual Negative	1,042	174,412

Table A.7: Confusion Matrix for Entire Dataset and 8 Predictors for training data (16-18-0 architecture).

	Predicted Positive	Predicted Negative
Actual Positive	864	10,085
Actual Negative	726	116,242

Table A.8: Confusion Matrix for Entire Dataset and 8 Predictors for Test Data (16-18-0 architecture).