# Asymmetric Error Control for Classification in Medical Disease Diagnosis

by

Wasif Bokhari

A Dissertation Presented in Partial Fulfillment
of the Requirement for the Degree
Doctor of Philosophy

Approved April 2021 by the
Graduate Supervisory Committee:

Ajay Bansal, Chair
Yu Zhang
Yezhou Yang
Faisal Bahadur

ARIZONA STATE UNIVERSITY

May 2021

# ABSTRACT

In classification applications, such as medical disease diagnosis, the cost of one type of error (false negative) could greatly outweigh the other (false positive) enabling the need of asymmetric error control. Due to this unique nature of the problem, traditional machine learning techniques, even with much improved accuracy, may not be ideal as they do not provide a way to control the false negatives below a certain threshold. To address this need, a classification algorithm that can provide asymmetric error control is proposed. The theoretical foundation for this algorithm is based on Neyman-Pearson (NP) Lemma and it is complemented with sample splitting and order statistics to pick a threshold that enables an upper bound on the number of false negatives. Additionally, this classifier addresses the imbalance of the data, which is common in medical datasets, by using Hellinger distance as the splitting criterion. This eliminates the need of sampling methods, which add complexity and the need for parameter selection. This approach is used to create a novel tree-based classifier that enables asymmetric error control.

Applications, such as prediction of the severity of cardiac arrhythmia, require classification over multiple classes. The NP oracle inequalities for binary classes are not immediately applicable for the multiclass NP classification, leading to a multi-step procedure proposed in this dissertation to extend the algorithm in the context of multiple classes. This classifier is used in predicting various forms of cardiac disease for both binary and multi-class classification problems with not only comparable accuracy metrics but also with full control over the number of false negatives. Moreover, this research allows us to pick the threshold for the classifier in a data adaptive way. This dissertation also shows that this methodology can be extended to non medical applications that require classification with asymmetric error control.

# ACKNOWLEDGMENTS

First of all, I would like to thank all of my family for their support and encouragement throughout this journey. This includes my parents, Dr. Wajahat Bokhari and Fiona Wajahat, who encouraged me to pursue my research and get enrolled in the PhD program. I would like to thank my wife, Shumail Abbas, who always supported me and allowed me to put in the extra hours towards my PhD. Thank you for helping with everything. I would also like to thank my sister, Shazeen Bokhari, for proofreading my writing and always being available for support.

Secondly, I would like to thank my committee chair, Dr. Ajay Bansal, who was always available with his guidance and direction. This research would not have been possible without his expert guidance throughout the past few years. I would like to thank my committee members Dr. Yu Zhang and Dr. Yezhou Yang for their supervision throughout this process. I would like to specially thank Dr. Faisal Bahadur, who took time away from his critical role as a cardiologist to guide this research.

Finally, I would like to thank my co-workers and managers at Intel and Citizens Bank who allowed me to pursue my PhD while still working full time with them. Thanks a lot for the flexibility that allowed me to pursue my PhD. I would especially thank Intel for helping me with the tuition costs for most of my courses during my PhD.

## TABLE OF CONTENTS

APPENDIX

# LIST OF TABLES

Table Page

## LIST OF FIGURES

Figure Page

# Chapter 1

# INTRODUCTION

This chapter starts with the overview of asymmetric error control for classification in medical disease diagnosis. Next, it discusses the research objectives and challenges that this dissertation is trying to address. Finally, it goes over the organization of the rest of the dissertation.

## 1.1 Overview

Machine learning involves the use of statistics to recognize patterns in data. Data can include numbers, images, clicks, words, and anything that can be digitally stored and used as input to a machine learning algorithm [32]. Supervised learning is the machine learning task where the data is labeled to guide the algorithms on what patterns it should look for. Classification [44; 45], a part of supervised learning [46], aims to automatically recognize and predict discrete outcomes for new observations after being trained on labeled data and patterns. Some of the most well known examples of classification include disease diagnosis, image classification as well as recognizing if the email is spam or not. Binary classification [47; 48] is the most common type of classification, in which the class labels can have only two values such as 0 or 1. Multi-class classification involves more than 2 class labels. Most classification models optimize for accuracy without providing control over the number of false negatives.

However, in medical disease diagnosis [49], where the cost of one error greatly outweighs the other, there is a need for asymmetric error control. Similarly, predicting if a patient is going to have a cardiac disease is a binary classification problem, where the cost of misclassifying a patient with high risk as no risk (False Negative) has a much bigger penalty than misclassifying a patient with no risk as high risk (False Positive). The former could cost a life whereas the latter may only cause medical costs and stress to the patient.

Traditional machine learning models [51; 52] may not be ideal in this scenario as they do not provide a mechanism to control the number of false negatives below a certain threshold. Even if these models result in improved accuracy and reduced classification error, they may not be optimal since the cost of one error greatly outweighs the other.

To address this need, we have created a tree-based classifier that can control the number of false negatives below a specified threshold value. This classifier is able to provide control over one type of error. The theoretical foundation for this model is based on Neyman-Pearson (NP) Lemma [53], which shows that the likelihood ratio test is the most powerful test in hypothesis testing. Based on the NP Classification umbrella implementation [13], this concept is expanded to create this tree-based classifier that can control the false negatives to a given value while still providing comparable accuracy and F1 score on different datasets [59].

Another challenge in medical disease diagnosis is the uneven distribution of positive and negative classes. This is because a positive case is typically a rare event compared to a negative test for a particular disease, leading to an imbalanced dataset. To address this problem without the need of extensive sampling techniques, this dissertation proposes the use of Hellinger distance as the tree splitting criterion. Hellinger distance addresses the imbalance by taking the difference of the two probability distributions, which is explained later in this dissertation.

Since this classifier can have the biggest impact in medical disease diagnosis, we use the problem of predicting various aspects of cardiovascular disease (CVD) as our case study to test out the classifier. This classifier predicts CVD with asymmetric error control enabling us to not only limit the number of false negatives for binary classification, but also for multi-class classification.

## 1.2 Research Objectives

This dissertation addresses the following research objectives for classification in medical disease diagnosis with asymmetric error control:

### Research Objective 1 (RO1)

To achieve asymmetric error control in binary classification with high probability that the population false negatives will not exceed a predefined user specified threshold $\alpha$. Additionally, this objective seeks a data adaptive way to select this $\alpha$ value.

### RO1 Challenges

Controlling the number of false negatives is a common requirement for medical disease diagnosis. In classification terms, false negative is often referred as type 2 error, and controlling type 2 error allows asymmetric error control.

The term “high probability” in the research objective implies that the classifier is able to control the type 2 error below a certain user specified value. For example, we can specify this value as 95 percent, which means there is a 95 percent probability that the type 2 error will be controlled. The last part of the research objective mentions a “predefined threshold”, which implies another user specified value that is referred as alpha later in this dissertation. This value is the maximum threshold percentage of false negatives which is acceptable for the classifier. For example, an alpha value of 0.1 implies that the false negatives will be controlled under 10 percent.

It is important to highlight the differences from cost based learning where we can adjust the weights depending on the higher cost of a false negative. The cost based learning does not provide a consensus way to assign costs, and provides no theoretical control on the population type 2 error. Moreover, with cost based learning, there is no easy way to compare population error control of different classifiers.

Another challenge will be in using the Neyman-Pearson (NP) Lemma from hypothesis testing in the classification space and showing how this approach can help resolve some of these issues. Evaluation of this approach should also include findings of any new issues that may arise with the use of NP in classification.

### Research Objective 2 (RO2)

To choose a classification threshold, independently from the training of the classifier, that controls the population type 2 error.

### RO2 Challenges

This goal of this research objective is to choose a classification threshold that controls population type 2 error under a user specified value with high probability. One challenge is to differentiate between population type 2 error versus empirical type 2 error. While controlling the type 2 error, we have to make sure that the control is being achieved over the population type 2 error. Appendix C provides a simulation study that showcases how controlling the empirical error does not correspond to controlling the population error. Additionally, another challenging task is to ensure that this threshold is not used in training of the resulting classifier to prevent bias from training data and prevent overfitting.

**Research Objective 3 (RO3)**

To extend asymmetric error control for binary classification to multi-class classification with high probability that the population false negatives will not exceed a predefined user specified threshold $\alpha$.

**RO3 Challenges**

The challenging tasks here involve ensuring that any theoretical properties on the type 2 error achieved using NP lemma for binary classification also holds for multi-class classification, and we still have a suitable method to evaluate our classifier. The NP oracle inequalities do not directly apply to multi class, so one of the main challenges is to achieve asymmetric control in multi-class classification problems using this approach.

**Research Objective 4 (RO4)**

To improve the quality of prediction of cardiac disease using classification.

**RO4 Challenges**

The challenge is to outperform the current state of art prediction techniques used in the hospitals using machine learning. For an accurate evaluation, it is important to not only get validation from an active cardiologist, but also to perform an evaluation over the same dataset to compare the results. The motivation of this research objective arises because in predicting disease, the cost of one error could be much higher than the other and data imbalance is common. The challenge in disease diagnosis, where one error outweighs the other and imbalance exists, is to create a classifier that enables asymmetric error control and allows classification without the need of extensive sampling.

## 1.3 Organization

The main parts of this dissertation are present in the next four chapters. Chapter 2 shows how this research is able to predict cardiac disease better than the current state of art systems being used in hospital settings. Chapter 3 shows how asymmetric error control is achieved for binary classification across diverse datasets. Chapter 4 shows how this approach is expanded to multi class classification problems not only in medical disease diagnosis but also in other domains. Chapter 5 compares asymmetric error control from handling data imbalance point of view and compares our approach with the state of art imbalance handling techniques. Chapter 6 summarizes the work and proposes future research.

Figure 1.1 summarizes the introduction section and shows how different aspects of asymmetric error control are connected to each other. The figure shows how imbalanced datasets and unequal cost of the errors led to the creation of AEC tree classifier, which uses hellinger distance as the tree splitting criterion and NP Lemma based approach. The figure shows how this is further expanded to predict cardiac disease and solve multi-class classification problems.

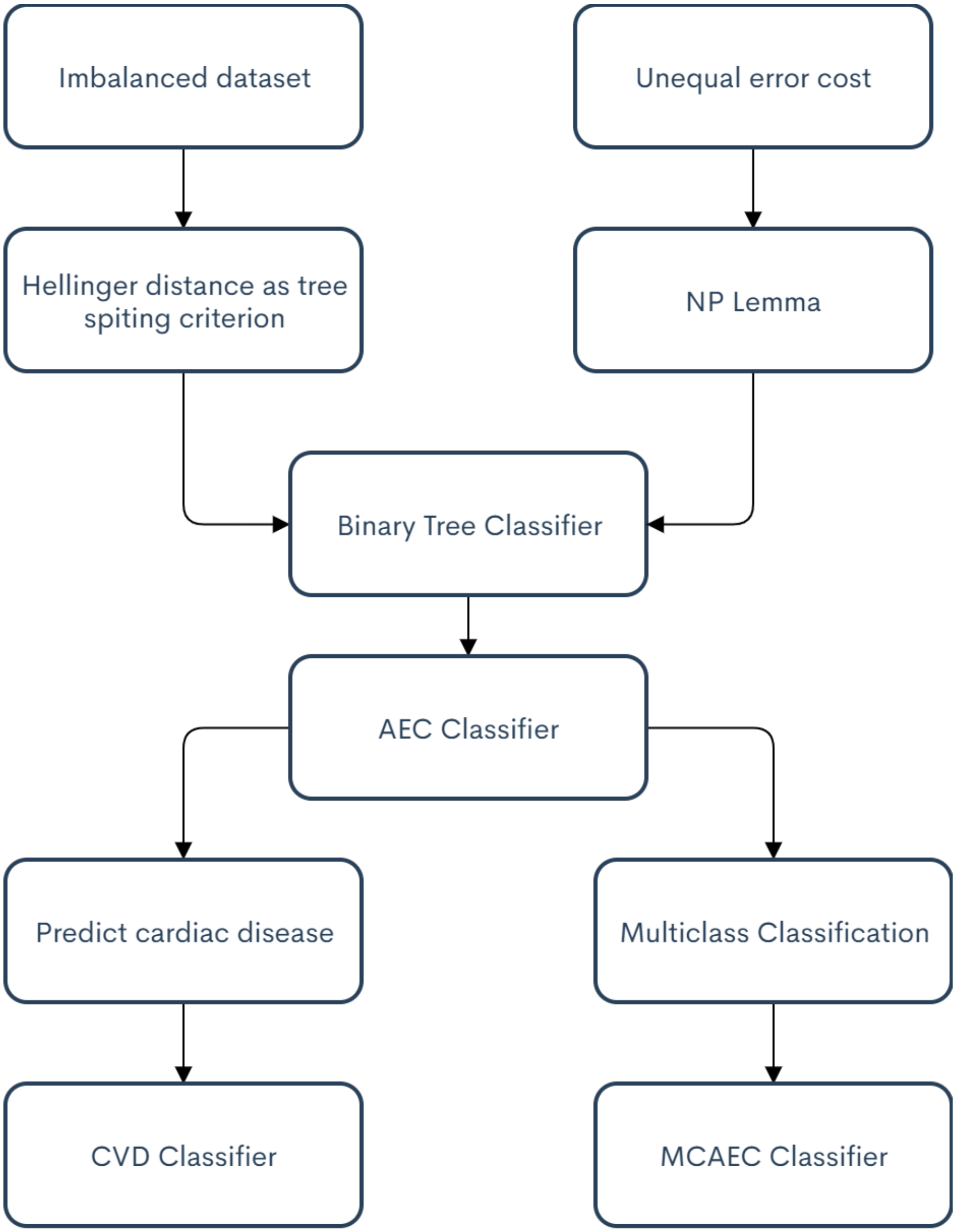


Figure 1.1: Overview of Asymmetric Error Control Classifiers

Chapter 2

# PREDICTING CARDIAC DISEASE WITH ASYMMETRIC ERROR CONTROL

This chapter goes over how our classifier is able to predict cardiac disease better than the existing methods being used in the hospital settings. The chapter starts with some background on this problem followed by the relevant literature in this domain. This chapter then shows the methodology used to construct the classifier and then concludes with the evaluation and results.

## 2.1 Background

Cardiovascular diseases (CVDs) are a group of disorders related to the heart and blood vessels. CVD is the number one cause of death worldwide as more people die from CVD than any other cause [2]. Globally, an estimated 17.9 million deaths are attributable to CVD on average every year [3]. According to the American Health Association, ”475,000 Americans die from cardiac disease in a given year and globally, cardiac disease claims more lives than breast cancer, prostate cancer, influenza, pneumonia, auto accidents, HIV, firearms, and house fires combined [1].”

Despite the alarming numbers, most premature deaths with CVD can be prevented with early detection. Individuals with high risk of CVD need to be identified early so they can take necessary preventive measures by addressing behavioural risk factors such as unhealthy diet, lack of physical activity and excessive use of alcohol and tobacco [2].

This chapter is focused on predicting the likelihood of a patient developing cardiovascular disease within the next 10 years accurately using machine learning. It is inspired by my previous study to improve cardiac resuscitation outcomes at the emergency cardiac center, Mayo Clinic (4, 21, 22). During the study, it was alarming to note that 90% of the patients did not survive the resuscitation attempts during cardiac arrest. One way to mitigate this is to identify patients who have a high risk of developing cardiac disease early so they can take preventive measures.

The next section discusses the related work and the existing methods being used to predict the risk of 10-Year CVD. In this section, the dissertation also covers the impact machine learning has made on the field of cardiology. Next, we propose the methodology and the theoretical foundation behind the Neyman Pearson Tree Classifier that we have created. The evaluation section explains the experimental setup used to compare our classifier. Finally, the results from the evaluation are presented and compared with the existing models to conclude that our tree-based classifier outperforms the current state-of-art 10-Year CVD risk prediction models.

## 2.2 Relevant Literature

This section is broken down into 5 subsections. Firstly, we discuss the risk scores that are being used to predict cardiac disease currently. The following two subsections discuss how AI is revolutionizing cardiac care and how AI can detect early stage heart disease. Next, this section discusses how machine leaning is used in code blue cardiac events and then finally concludes by elaborating the challenges for machine learning in the medical domain.

### *2.2.1 Risk Scores*

The nine most widely used risk models for predicting 10-year risk of CVD are listed in table 2.1, in order of popularity. The most popular risk model to predict the risk of cardiac disease is a common scoring mechanism called the Framingham Risk Score [20]. This is a gender-specific algorithm used to estimate the 10-year cardiovascular risk of an individual. The Framingham Risk Score was first developed based on data obtained from the Framingham Heart Study, to estimate the 10-year risk of developing coronary heart disease. This score has an underestimation of over 44 percent according to the Renfrew/Paisley Study [7]. This number worsens if the participants are chosen from deprived areas.

Table 2.1: Most Commonly Used 10 year CVD Prediction Scores

| # | **CVD Prediction Method** |
|---|---|
| 1 | Framingham Risk Score |
| 2 | Framingham Coronary Heart Disease Risk Score |
| 3 | Framingham Atp-III |
| 4 | ASCVD Coronary Heart Disease Risk Prediction |
| 5 | Reynolds Risk Score |
| 6 | Procam Score |
| 7 | QRisk Score |
| 8 | Cuore |
| 9 | Assign |

The other score, which is commonly used is called Atherosclerosis Cardiovascular Disease (ASCVD) Risk score. This risk score takes into consideration different races and gives 10 year and lifetime risk of cardiovascular death. However, this score overestimates the risk significantly in adults without diabetes [8].

Many studies have been performed to evaluate the performance of these risk models. After extensive evaluation by the American College of Cardiology (ACC), it was concluded that none of the most widely cited Framingham and ASCVD coronary heart disease risk predictions (table 2.1) are able to predict accurately [5]. Similarly, another study on the performance of risk models to predict 10-year CVD risk published in BMC Medicine journal last year concluded that all models overestimate the 10-year risk of CVD [6]. These studies also highlighted a need for an accurate prediction model and the massive impact it could have. This research aims to address this need by proposing a more accurate tree-based classifier that can not only predict more accurately but also control the number of false negatives under a controlled threshold value.

### *2.2.2 AI Revolutionizing Cardiac Care*

Cardiovascular diseases are not only the number one cause of death resulting in 31 percent of global deaths but also one of the most expensive medical conditions to treat, according to Forbes research. There are four main ways Artificial intelligence (AI) can revolutionize cardiovascular care.

First and foremost, AI can have a big impact on diagnosing cardiovascular diseases. Typical diagnosis involves three stages. The first stage is measuring ECG at rest and looking for anomalies. If anomalies are found, it leads to semi-invasive tests such as stress test, chest CT scan and echo-cardiography. AI is already being used to predict the anomalies quickly without using the third invasive step. This can be extended to potentially diagnose diastolic dysfunction in patients [96].

Secondly, AI aided cardiac imaging can enhance live visualization of the heart and can vastly improve the efficiency of clinical workflow of cardiologists. Zebra [97] a medical technology company has created a Coronary Calcium Scoring algorithm which can provide early detection of people at high risk of severe cardiovascular events based on chest CTs.

Third way involves AI based therapy sessions. KenSci [98] uses machine learning to predict the risk of a patient acquiring heart disease. “KenSci was built by doctors and data scientists to help providers and payers intervene earlier, at lower costs. KenSci's risk prediction platform helps uncover clinical, operational and financial risks by aggregating data from existing sources such as EMR, ADT, Claims and Financial data." [98]

Lastly, AI can play an important role in continuous monitoring using devices such as fitbit to predict early warning signs of lifestyle diseases. For example, Cardiogram's DeepHeart [99] works with Apple Watch as a semi-supervised AI learning for cardiovascular risk prediction. Consumer wearables generate two trillion health measurements a year according to research done at the University of California [100] which is too many for any human doctor to review. However, a novel deep neural network tested in multiple rigorous clinical studies similar to DeepHeart can get trained with this data.

### *2.2.3 AI and Early Stage Heart Disease*

Artificial intelligence can be used to detect early stage heart disease. "Asymptomatic left ventricular dysfunction (ALVD) is characterised by the presence of a weak heart pump with a risk of overt heart failure. It is present in three to six percent of the general population and is associated with reduced quality of life and longevity. However, it is treatable when found." [101]

Currently, there is no inexpensive, noninvasive, painless screening tool for ALVD available for diagnostic use. However, a trained neural network can reliably detect ALVD with reasonable accuracy rate. Paul Friendman, chair of the Midwestern Cardiovascular Department at Mayo Clinic, led the research using only 12-lead ECG and echo-cardiogram data to identify patients with ventricular dysfunction. After being trained from 45000 patients at Mayo Clinic, the network was tested on an independent set of 53000 patients and the algorithm predicted patients with ventricular dysfunction with an accuracy of 85 percent.

Deep-learning algorithms can be applied to large datasets of electrocardiograms, are capable of identifying abnormal heart rhythms and mechanical dysfunction, and could aid healthcare decisions [102]. Moreover, there are many more aspects or variables that have a correlation in cardiovascular disease such as: [103]

- Smoking status and lifetime exposure
- Age
- Diet
- Alcohol consumption
- Ethnicity
- Immigration status
- Stress
- Sense of belonging
- Physical activity
- Education
- High blood pressure
- Diabetes
- Socioeconomic status of the neighbourhood

Researchers at the Ottawa Hospital collected this type of data from over 100,000 Canadians from the community health surveys and trained their neural network to predict the risk of death or hospitalization from cardiovascular disease within the next 5 years. Once the danger of CVD has been identified, individuals can work on their lifestyle factors to help lessen the risk [103].

### *2.2.4 Code Blue and Resuscitation*

Code Blue is an emergency code that is used in hospitals to indicate when a patient goes into cardiac arrest and needs resuscitation. A medical team is paged and rushes in to attempt to save the patient's life when a Code Blue is called. According to research by NorthShore University HealthSystem, the survival rates are less than 20 percent and it remains a very intense, resource intensive, expensive and chaotic process. However, medical research shows that the patients actually start displaying clinical signs of deterioration for some time before actually going into cardiac arrest which makes early prediction and intervention possible.

Research indicates that patients actually start showing clinical signs of deterioration some time before going into cardiac arrest, making early prediction, and possibly intervention, feasible. Researcher at NorthShore University HealthSystem have developed machine learning classification models using support vector machine (SVM) and logistical regression, that preemptively flags patients who are likely to go into cardiac arrest, using signals extracted from demographic information, hospitalization history, vitals and laboratory measurements in patient-level electronic medical records [104]. They found that early prediction of Code Blue is possible and when compared with state of the art existing method used by hospitals (MEWS - Modified Early Warning Score), their methods perform significantly better. Based on these results, this system is now being considered for deployment in hospital settings in Chicago. [9]

The International Research Journal Of Engineering And Technology also tried to predict the code blue event specifically cardiac arrhythmia. It was performed using tensor flow along with keras for training the neural network. The dataset used was taken from the UCI machine learning repository [10] and consisted of 452 patient records and 279 attributes. The dataset was divided into 70 percent training and 30 percent testing set but it only yielded an accuracy of 58 percent. They concluded, "There exists scope for improvement since the existing system is taking into account the reading interpreted from ECG ratings. So, better pre-processing techniques will help us to remove more redundant data plus at the same time more patient records will be required so as to train the model for various different scenarios". (11, 23)

This research is motivated by my background in developing software applications for code blue outcomes for Mayo Clinic. While researching the code blue resuscitation process, I was amazed to see such a high death rate. Dr Ayan Sen, emergency care specialist at Mayo Clinic, recommended that an early diagnosis would massively improve patient outcomes. The current systems used by hospitals(MEWS) are not very accurate. Machine learning techniques have not been applied extensively in the cardiac arrest domain for prediction of outcomes. I also observed that at the time of code blue, it is often too late to save the life of a patient, as resuscitation is not always successful. This led me to explore ways to improve cardiac care so that cardiac disease can be identified earlier in the process before it leads to a possible code blue event.

### *2.2.5 Machine Learning Challenges in Medical Domain*

Modern machine learning is data intensive. For example, to make speech recognition work on a smartphone, Google has to train a deep neural network on over 10k hours of annotated speech. Similarly, ImageNet contains more than a million hand-annotated images. These labels are the key components to make deep learning successful.

In the medical domain, each label represents a human life. An example by the cardiogram journal states, “In our study with UCSF Cardiology, labeled examples come from people visiting the hospital for a procedure called cardioversion, a 400-joule electric shock to the chest that resets your heart rhythm. It can be a scary experience to go through. Many of these patients are gracious enough to wear a heart rate sensor (e.g., an Apple Watch) during the whole procedure in the hope of making life better for the next generation, but we know we’ll never get one million participants, and it would be unconscionable to ask.” [106]

The challenge here is to make AI work with fewer labels than it is used to. We can aim to address this problem using unsupervised techniques which can find trends and structures in unlabelled data. Hybrid techniques such as semi supervised sequence learning and one shot learning are other techniques that can make fairly accurate predictions with less labelled data as long as there is a lot of unlabeled data present. Generation of unlabeled data has become easier now with the advance of sensors and wearable devices such as Apple Research Kit and Google fit. This allows us to collect health data at large scale which can be translated to useful data that can enable clinicians and patients to take real actions based on the results of the deep learning algorithm.

Other non technical challenges that are unique to the medical domain include deployment in hospital EMR systems and the many regulations which serve as a barrier to entry for machine learning to be adopted in mainstream hospitals. For example, a misdiagnosis in hospitals will lead to follow up tests and visit, which brings more revenue for the hospital. A better algorithm could actually reduce the revenue of the hospitals which becomes a bad business decision from return on investment point of view. The solution to these challenges as mentioned by the cardiogram research blog states, “Enable outside-in approaches to healthcare: build up a user base outside the core of the healthcare system (e.g., outside the EMR), but take on risk for core problems within the healthcare system, such as re-hospitalizations. Together, these two factors let startups solve problems end-to-end, much the same way Uber solved transportation end-to-end rather than trying to sell software to taxi companies."

To be very specific towards the challenges in Cardiac disease, one big one is that it remains a very rare event. To account for the imbalance of the two class labels, as positive cardiac disease is very rare versus non cardiac disease events, we will have to sample the data to make our training set less skewed. Still the extreme imbalance in class labels in the data has caused some difficulties in creating test and training sets. A good plan would be to combine data from different hospital systems, so the system can be trained on more positive examples, which should improve predictions further.

Another unique challenge in this domain for Code Blue events is identifying the negative event. The positive event will be when patients go into Code Blue. However, for patients who do not go into Code Blue, we have their blood level information at time for admission and time when they were discharged. Most patients will be healthier at time of discharge compared to time of admission which makes using time of discharge as a comparable event a bad choice as our classification problem will become artificially easy. Using time close to discharge will not give us enough comparable information to build an accurate model. Instead, we can vary the time of event for control patients, as the 25th, 50th and 75th percentile of their stay time in the hospital, and select the value that performs best in the training data [104].

Most of AI Success has come from problems where the goal state is easily and uncontroversial quantified. However, for biomedical domain, the end goal is not always well defined. Research from University of California, San Diego poses some interesting questions, "What precisely is the goal of a doctor? How can we distill success of a medical professional to a single reward that is dolled out as outcomes become known? How precisely do we measure quality of life? What is the trade-off between limbs and longevity? How much should the doctor value revenue vs patient health (if the value of life is infinite then all patients should be seen for free). Sometimes the objective function for the doctor might vary from patient to patient depending on their preferences. Human doctors implicitly evaluate these trade-offs constantly, but before we learn to canonize our objectives current AI may remain confined to more isolated, low-level pattern recognition problems." [105]

Currently, machine learning has been remarkable in recognizing patterns, images as well as deriving meaning from certain sentences. However, it is not able to show that it can abstract concepts from limited experience and transferring knowledge between domains. Both of these traits are very useful in the medical field that requires diagnosing and treating novel conditions. As the medical futurist remarks, "Although data, measurements and quantitative analytics are a crucial part of a doctor's work setting up a diagnosis and treating a patient are not linear processes where AI is lacking at the moment." [106]

Typically, machine learning algorithms are optimized to obtain peak accuracy. However, in medical diagnosis, the control of false negatives may even be more important than improving accuracy as a false negative could mean failing to diagnose a disease that could be life threatening. This is especially important in cardiology as the heart remains the most important organ in the human body (12, 24). Through this research, we propose a classifier in the cardiology field that is optimized to control the false negatives.

## 2.3 Methodology

Predicting if a patient is going to have cardiac disease is a binary classification problem, where the cost of misclassifying a patient with high risk as no risk (False

Negative) has a much bigger penalty than misclassifying a patient with no risk as high risk (False Positive). This is because the former could cost a life, whereas the latter may only cause medical costs and stress to the patient. Due to this unique nature of the problem, where one error greatly outweighs the other, traditional machine learning techniques, even with much improved accuracy, may not be ideal as they do not provide a way to control the false negatives below a certain threshold.

The current state of the art binary classifiers algorithms are optimized to minimize the classification error instead of providing asymmetric error control. The classification error [28] is calculated as:

$$\text{Classification Error} = FP + FN * 100/N \tag{2.1}$$

where,
FP is the number of false positives (type 1 error)
FN is the number of false negatives (type II error)
N is the total number of samples

Equation 2.1 shows that we can control the classification error using the traditional classification techniques but there is no way to limit one type of error. This is because the false negatives and false positives are grouped together in this equation.

To address this need, a classification algorithm that can control the false negatives to a certain threshold is proposed. The theoretical foundation for this algorithm is based on Neyman-Pearson (NP) Lemma, which shows that the likelihood ratio test is the most powerful test in hypothesis testing. Based on the NP Classification Umbrella Implementation [13], this concept is applied to create a tree-based classifier, called CVD Tree Classifier, that can control the false negatives to a given value while still providing improved accuracy on the Framingham dataset.

This CVD Tree Classifier is based on decision tree classifiers [25; 26]. It consists of a large number of individual decision trees that operate together as an ensemble. Each tree in the classifier produces a class prediction and the model's predicted class is selected based on the class that has received majority of votes.

Similar to popular ensemble decision trees such as Random Forest and extra trees, our classifier is trained over various sub-samples of the data and each tree is grown to its largest [27; 28]. However, the most critical point of decision tree induction algorithms is the choice of the splitting criteria [29] of a node. We use Hellinger distance as the splitting criteria because it addresses the imbalance in the dataset by quantifying the difference between two probability distributions. This eliminates the need of extensive sampling techniques and hyper tuning that are required using traditional splitting criterion's such as Gini Index and Entropy [28]. Hellinger distance as a function is calculated for all attributes and it provides the highest value of split measure for our feature set. This algorithm is then adapted to control the false negatives below a certain threshold using the NP classification paradigm.

Specifically, we seek an efficient way to choose a threshold for the classification scores predicted by our tree classifier so that the threshold leads to classifiers with false negatives below the user specified upper bound .This algorithm is needed because the naïve approach, which simply picks a threshold by setting the empirical type I error to no more than alpha, fails to satisfy the type I error constraint, as demonstrated in the simulation study [13] conducted by American Association for Sciences. The major aspects of the NP algorithm are listed below.

### *2.3.1 Theoretical Foundation*

The NP classification is based on the NP Lemma. The lemma states that the likelihood test is the most powerful hypothesis testing among all other tests [23]. The

NP Lemma says:

$$X \to X \sim P(X, H_0)$$

$$X \to X \sim P(X, H_1)$$

where,

X is the data or the observation

$H_0$ is null hypothesis.

$H_1$ is alternative hypothesis

$P(X, H_0)$ is probability distribution of X given $H_0$

$P(X, H_1)$ is probability distribution of X given $H_1$.

This leads us to the likelihood ratio test L(X) which states:

$$L(X) = \frac{P(X_1, H_1)}{P(X_1, H_0)} > \gamma$$

where $\gamma$ (gamma) is threshold gamma. To maximize $P_D$ for given $P_{FA}= \alpha$ where $P_D$ is probability of detection and $P_{FA}$ is probability of false alarm. The given alpha $\alpha$ is the upper bound threshold value. $P_{FA}$ is then calculated as:

$$P_{FA} = P_r L(X) > \gamma, H_0 = \alpha$$

which is the probability that the likelihood ratio is greater than gamma under the null hypothesis $H_0$. These equations show how the NP Lemma is able to bound the probability of false alarm, which is equivalent to a false negative, with a threshold value setting the foundation for asymmetric error control.

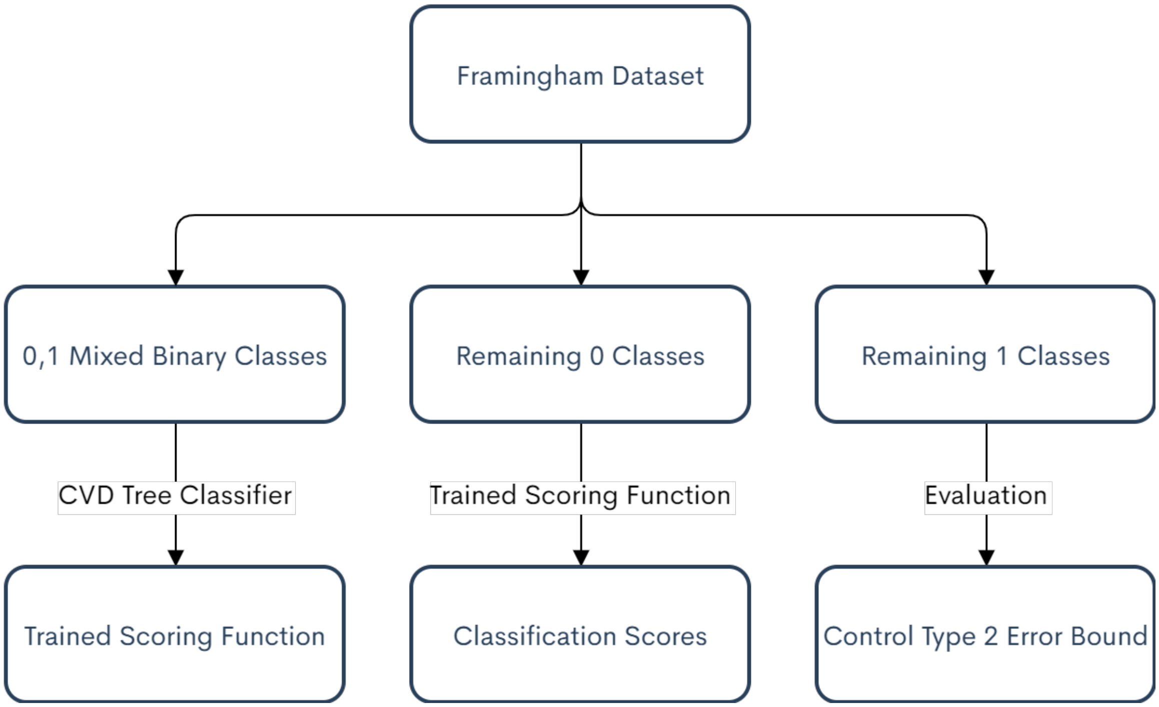


Figure 2.1: Sample Splitting on Framingham Data

### *2.3.2 Sample Splitting*

Sample splitting is the first step of incorporating the NP lemma in the classifier. It involves splitting the training data into three parts as explained in figure 2.1 below. The figure shows mixed classes of 0 and 1 samples outputting a trained scoring function and remaining class 0 samples producing our classification scores. The remaining class 1 samples are evaluated to control false positives error bound.

### *2.3.3 Threshold Search and Order Statistics*

We chose the smallest value of the threshold (alpha) on the classification scores, that are obtained from previous step, such that violation rate remains minimized. To find the threshold, brute force and bagging were tried initially but order statistics was proven to be the best in finding this threshold value from the classification scores

[16].

The three steps above are incorporated in our decision tree to create the CVD Tree Classifier that is able to control the false negatives below a certain threshold with high probability such that the upper bound on the violation rate is:

$$V(k) = \sum_{j=k}^{n} (j^n)(1-\alpha)^j \alpha^{n-j}$$

where V(k) is violation rate which is the probability that the false negatives exceed the threshold value and n is the sample size. $\alpha$ is the threshold which is the upper bound on the percentage of false negatives.

This produces a classifier that has threshold optimized during learning and training process. This CVD Tree classifier can be adapted to be used in other areas of medical diagnosis as well where the cost of one type of error greatly outweighs the other.

## 2.4 Identifying high-risk Features for CVD

We have obtained the dataset of the patients involved in the Framingham Heart Study, which includes patient records of over 4200 patients. The features of this dataset are shown in table 2.2.

| **Attributes** | **Description** |
|---|---|
| Sex | Male or Female |
| Age | Age of the patient |
| Education | 1 = Some High School;<br>2 = High School or GED;<br>3 = Some College or Vocational School;<br>4 = College |
| Current Smoker | Whether or not the patient is a current smoker |
| Cigs Per Day | The number of cigarettes that the person smoked on average in one day |
| BPMeds | Whether or not the patient was on blood pressure medication |
| PrevalentStroke | Whether or not the patient had previously had a stroke |
| PrevalentHyp | Whether or not the patient was hypertensive |
| Diabetes | Whether or not the patient had diabetes |
| TotChol | Total cholesterol level |
| SysBP | Systolic blood pressure |
| DiaBP | Diastolic blood pressure |
| BMI | Body Mass Index |
| HeartRate | Heart Rate |
| Glucose | Glucose Level |

Table 2.2: Dataset Features used as input to train the CVD Tree Classifier

We have performed a causality analysis on these features to not only identify the relations between them but also to gauge which attributes contribute most significantly to the onset of CVD. The results from this analysis are shown in table 2.3. This table shows the difference in percentage of risk of 10-year CVD based on presence and absence of each attribute. For example, the first row from this table corresponds to:

$$P(X = CVD \mid Y = \text{ Diabetes }) = 37\%$$
$$P(X = CVD \mid Y \neq \text{ Diabetes }) = 14.6\%$$

The last column in the first row from table 2.3 shows the difference between these two percentages as 22.4 percentage. A larger percentage difference implies that the attribute is a significant factor leading to CVD. The difference of each attribute is plotted in figure 2.2.

The first line shows the probability in percentage a person has CVD, given that they have diabetes as 37 percent. The second line shows the probability that the person has CVD, given that they do not have diabetes as 14.6 percent.

| **Attribute** | **Present** | **Absent** | **Difference** |
|---|---|---|---|
| Diabetes | 37% | 14.6% | 22.4% |
| Male | 19% | 12.4% | 6.6% |
| Age | 27.7% | 12.7% | 15.0% |
| Smoker | 15.9% | 14.5% | 1.40% |
| Smoker with 10+ cigs daily | 17.9% | 11.6% | 6.30% |
| BP Med | 33% | 14.6% | 18.4% |
| Prev Stroke | 44% | 15% | 29.0% |
| Prevalent Hypertension | 24.7% | 10.9% | 13.8% |
| Cholesterol | 18.6% | 13.3% | 5.30% |
| Sys BP | 18.4% | 8.6% | 9.80% |
| Dia BP | 18.4% | 11.3% | 7.10% |
| BMI >25 | 17.4% | 12.2% | 5.20% |
| BMI >30 | 19.2% | 14.3% | 4.90% |
| Heart Rate >80 | 16.1% | 14.8% | 1.30% |
| Heart Rate >100 | 19% | 15% | 4.0% |
| Glucose >100 | 24.9% | 14.6% | 10.30% |
| Glucose >125 | 45.3% | 14.8% | 30.50% |
| High School | 18.6% | 15% | 3.60% |
| GED | 11.7% | 15% | -3.30% |
| Vocational School | 14% | 15% | -1.0% |
| College | 14.8% | 15% | -0.20% |

Table 2.3: Probability that you have high risk of CVD given the presence or absence of these attributes

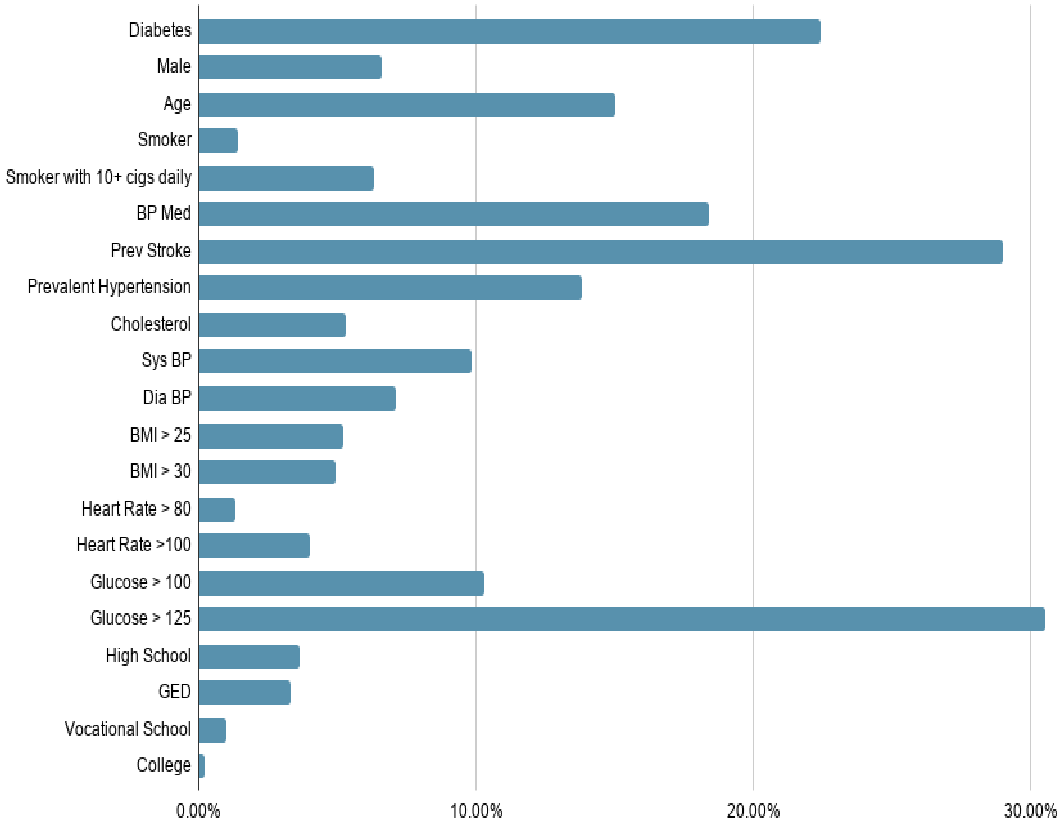


Figure 2.2: Percentage Difference (Significance) for each attribute that causes CVD

Figure 2.2 shows that the highest percentage difference is present for glucose greater than 125, presence of prevalence stroke and presence of diabetes respectively. This implies that these three are the most significant attributes for the prediction of CVD. Education level, which corresponds to the last 4 entries in figure 2.2 and table 2.3, has low percentage difference, which implies that education level may not be the most significant metric to predict CVD. This graph allows us to see impact of different attributes on the risk of 10-year CVD and identify the high risk features.

## 2.5 Evaluation

This sections explains the experimental setup created to evaluate the CVD Tree Classifier. It is broken down into the following subsections.

### *2.5.1 Dataset and Hyper Parameters*

Using the Framingham heart study dataset, we have trained and optimized our machine learning model to predict the risk of a ten-year CVD with not only improved accuracy, but also reduced underestimation. 70 percent of the dataset was used in training and the remaining 30 percent was used in testing. The number of trees in each forest was chosen to be 300. The minimum number of samples to split the node was chosen as 2 and the minimum number of samples required to be at leaf node was 1. The rest of the parameters were kept the same as the default values of decision tree algorithms.

### *2.5.2 Class Imbalance*

There was an imbalance in the dataset with a lot more negative classes as compared to positive classes. A simple naive model that returns all samples as negative could have high accuracy. To address this, Synthetic Minority Oversampling Technique (SMOTE) was used to balance the dataset. With a more balanced dataset, the accuracies of different models is more meaningful. Extensive use of sampling techniques was not needed due to the use of Hellinger distance as a splitting criterion in our decision tree.

### *2.5.3 Missing Values*

There were some values missing from the dataset in certain rows. Typically, rows with missing values are removed from the dataset. However, in the medical domain, we often do not have the luxury of massive datasets as each row represents an actual patient's medical record. So in order to maintain the row count of the original sample, the average of each column was calculated. This average was replaced in each column in place of the missing value to complete the row.

### *2.5.4 Type 1 vs Type 2 Error Control*

The NP classification provides a way for asymmetric error control on type 1 error which is the number of false positives, However, in the prediction of CVD, we need to control the type 2 error which is the number of false negatives. Using the same theoretical foundation based on the np lemma, we flipped our predicted variable column in the training set. All the 1 classes were changed to 0 and all the 0 classes were changed to 1. This meant that our null and alternative hypothesis also swapped which enabled us to use the same algorithm to control the type 2 error instead of type 1.

## 2.6 Results

The results can be divided into four main phases. First, we compare our CVD Tree Classifier with other most common machine learning classifiers and show that our classifier performs the best. Secondly, we show the control of false negatives using our CVD Tree Classifier. Next, we compare our classifier with the most cited medical risk prediction scores. Finally, we perform a comparison of CVD Classifier with the Framingham risk score on the same dataset.

### *2.6.1 CVD Tree Classifier vs Popular Machine Learning Classifiers*

We compared the accuracy of our CVD Tree Classifier with other machine learning classifiers as shown in figure 2.3. CVD Tree Classifier and Random Forest perform the best with accuracy over 85 percent. These results can be explained by the ability of decision trees to perform better on an imbalanced dataset by balancing error in class populations.

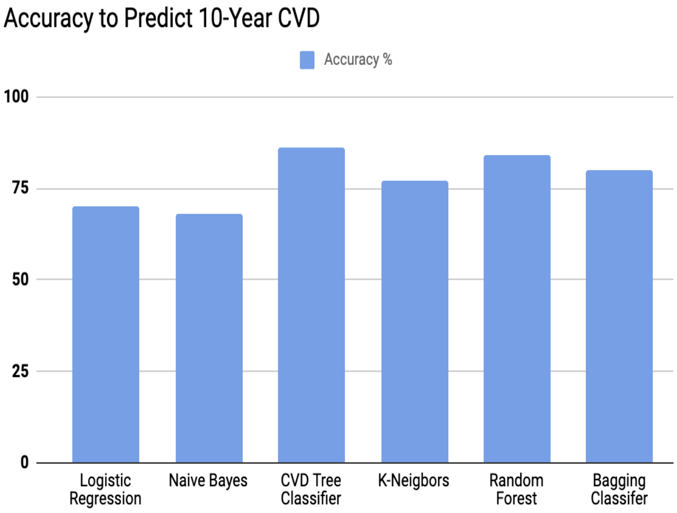


Figure 2.3: Accuracy of ML Classifiers vs CVD Tree Classifier

Next, we compare the "Area under Curve Receiver Operating Characteristics" (AUC-ROC) score. AUC-ROC is explained by the Data Science Journal as "It is a performance measurement for classification problems at various thresholds settings. ROC is a probability curve and AUC represents degree or measure of separability. It tells how much a model is capable of distinguishing between classes. Higher the AUC, better the model is at predicting 0s as 0s and 1s as 1s. By analogy, Higher the AUC, better the model is at distinguishing between patients with disease and no disease."

Figure 2.4 shows that CVD Tree Classifier outperforms other machine learning with a score of 0.82. Random Forest and K-Neighbors have second and third best AUC scores. This score is especially important validation for our CVD Tree Classifier because in our dataset with imbalanced classes, accuracy can be a little misleading. If we have an imbalanced dataset, then any algorithm that returns the majority class often will have a higher accuracy. This makes the AUC-ROC scores very important.

Table 2.4: CVD Tree Classifier Control Over False Negatives

| | **Accuracy%** | **AUC ROC%** |
|---|---|---|
| Logistic Regression | 71% | 58% |
| Naive Bayes Classifier | 68% | 56% |
| CVD Tree Classifier | 87% | 82% |
| K-Neighbors Classifier | 78% | 77% |
| Random Forest | 86% | 79% |
| Bagging Classifier | 82% | 77% |

Table 2.4 shows the values used to plot the figures 2.3 and 2.4. It shows the accuracy and AUC-ROC scores as percentages. The AUC-ROC typically ranges from 0 to 1 but the percentage has been calculated to normalize it for easier comparison with accuracy.

Finally, we compare the number of false negatives predicted by each classifier

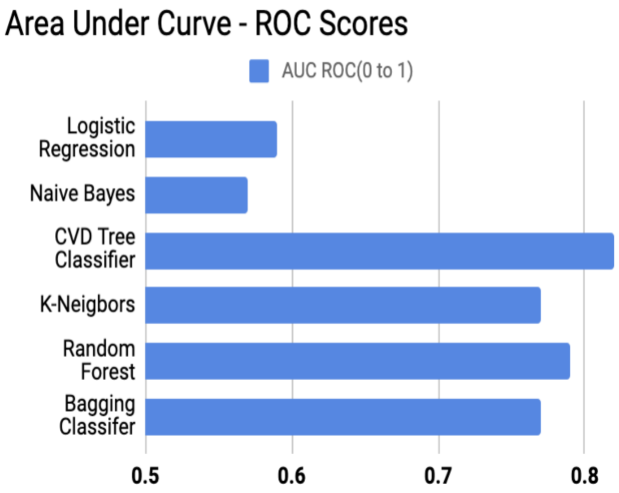


Figure 2.4: AUC-ROC of ML Classifiers vs CVD Tree Classifier

as shown in Figure 2.5 . This is one of the most important metrics for our CVD prediction as a false negative means that the classifier failed to diagnose a patient which had a high risk of CVD. This could potentially cause a loss of a human life as the patient will not get any treatment. Our CVD Tree Classifier was designed to keep the false negatives minimum using the theoretical approach from the NP Lemma which was explained previously.

The results in Figure 2.5 show that the CVD Tree Classifier greatly outperforms all other machine learning classifiers by having false negatives under 50 in a sample of over 4000 patients. Random Forest Classifier which was comparable to CVD Tree Classifier in terms of accuracy and AUC-ROC Scores falls behind here with close to 200 false negatives.

The results above clearly show that our CVD Tree Classifier outperforms all other machine learning classifiers in terms of accuracy, AUC-ROC scores and the least number of false negatives.

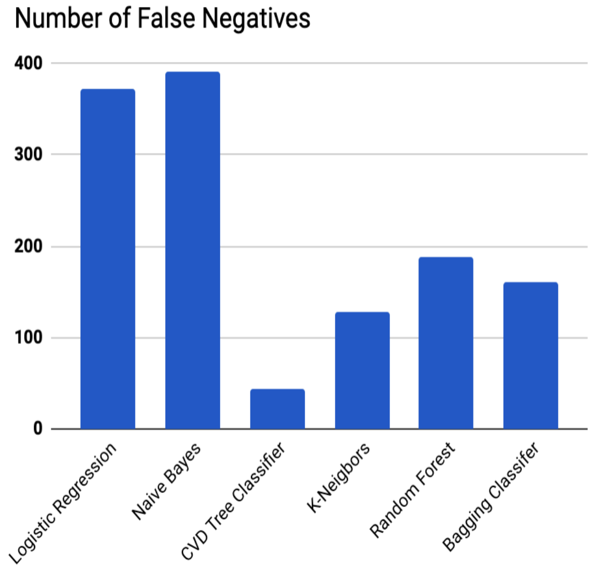


Figure 2.5: Number of False Negatives of ML Classifiers vs CVD Tree Classifier

### *2.6.2 CVD Tree Classifier Control Over False Negatives*

Figure 2.6 below shows how the accuracy varies with the upper bound on the false negatives. The graph shows that we are able to get an accuracy of 60 percent with false negatives close to 0 using our CVD Tree Classifier. Even though an accuracy of 60 percent is very low, it gives us the supreme advantage of keeping false negatives close to 0 which is not possible using traditional methods.

However, the optimal is achieved with upper bound on false negatives close to 18 percent as it still gives us an accuracy over 80 percent. Table 2.5 shows the exact values used to plot figure 2.6. As seen from this table, at the threshold value of 0.2, the accuracy is close to 82 percent and the false negatives are only around 18 percent which indicates that this threshold is close to the optimal value.

It should be noted that even though we can increase our accuracy close to 86 percent with false negatives at around 40 percent threshold value, this may not be a good option as we want to minimize the false negatives. In other use cases, the

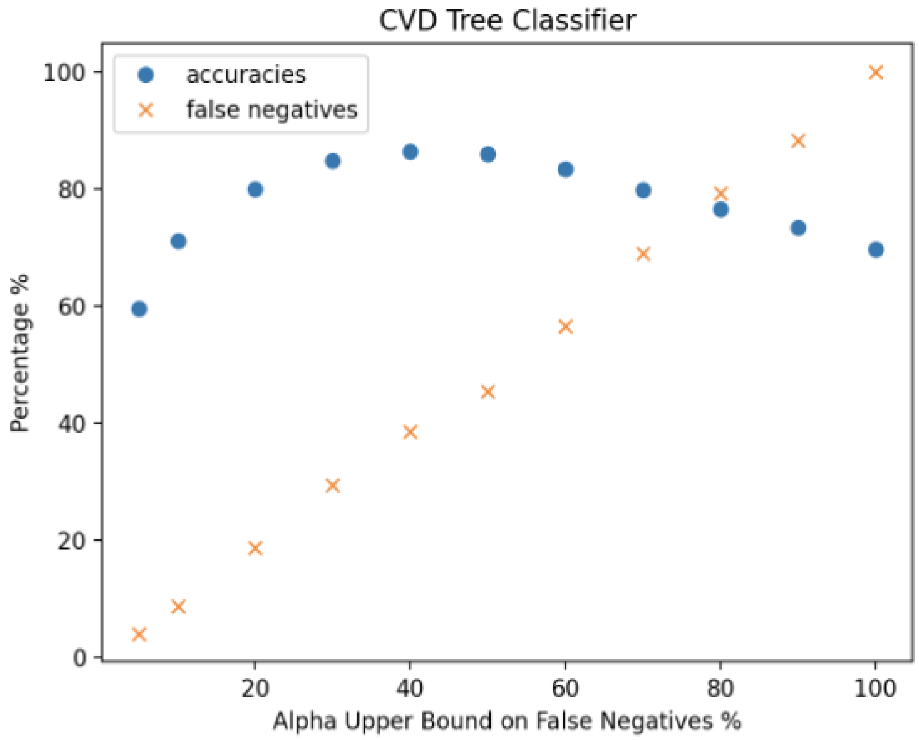


Figure 2.6: CVD Tree Classifier Control Over False Negatives with respect to Accuracy

threshold value should be chosen carefully depending on the potential consequences of false negatives. The results here also show that not all classifiers should be designed to be optimized for accuracy as other metrics such as false negatives may be equally important.

### *2.6.3 CVD Tree Classifier vs Current 10-Year CVD Risk Prediction Scores*

Based on the previous results, this dissertation has identified our CVD Tree Classifier as the best machine learning model to be used to predict 10-year CVD risk. In this section, we compare our classifier to the current state of art medical prediction scores which are not based on machine learning. To address the imbalance of dataset in the cardiac domain, we use AUC-ROC score for comparison. The AUC-

Table 2.5: CVD Tree Classifier Control Over False Negatives with Varying values of alpha ($\alpha$)

| **Alpha** | **Accuracy** | **False Negatives** |
|---|---|---|
| 0.05 | 61.3% | 1.7% |
| 0.1 | 72.1% | 5.2% |
| 0.2 | 81.9% | 17.8% |
| 0.3 | 86.0% | 29.1% |
| 0.4 | 86.3% | 39.1% |
| 0.5 | 85.0% | 49.7% |
| 0.6 | 81.4% | 58.5% |
| 0.7 | 79.1% | 68.7% |
| 0.8 | 75.3% | 78.7% |
| 0.9 | 72.0% | 89.1% |
| 1.0 | 66.2% | 96.8% |

ROC scores for the medical 10-Year CVD prediction risk scores are obtained from the Framingham heart study evaluation paper[20].

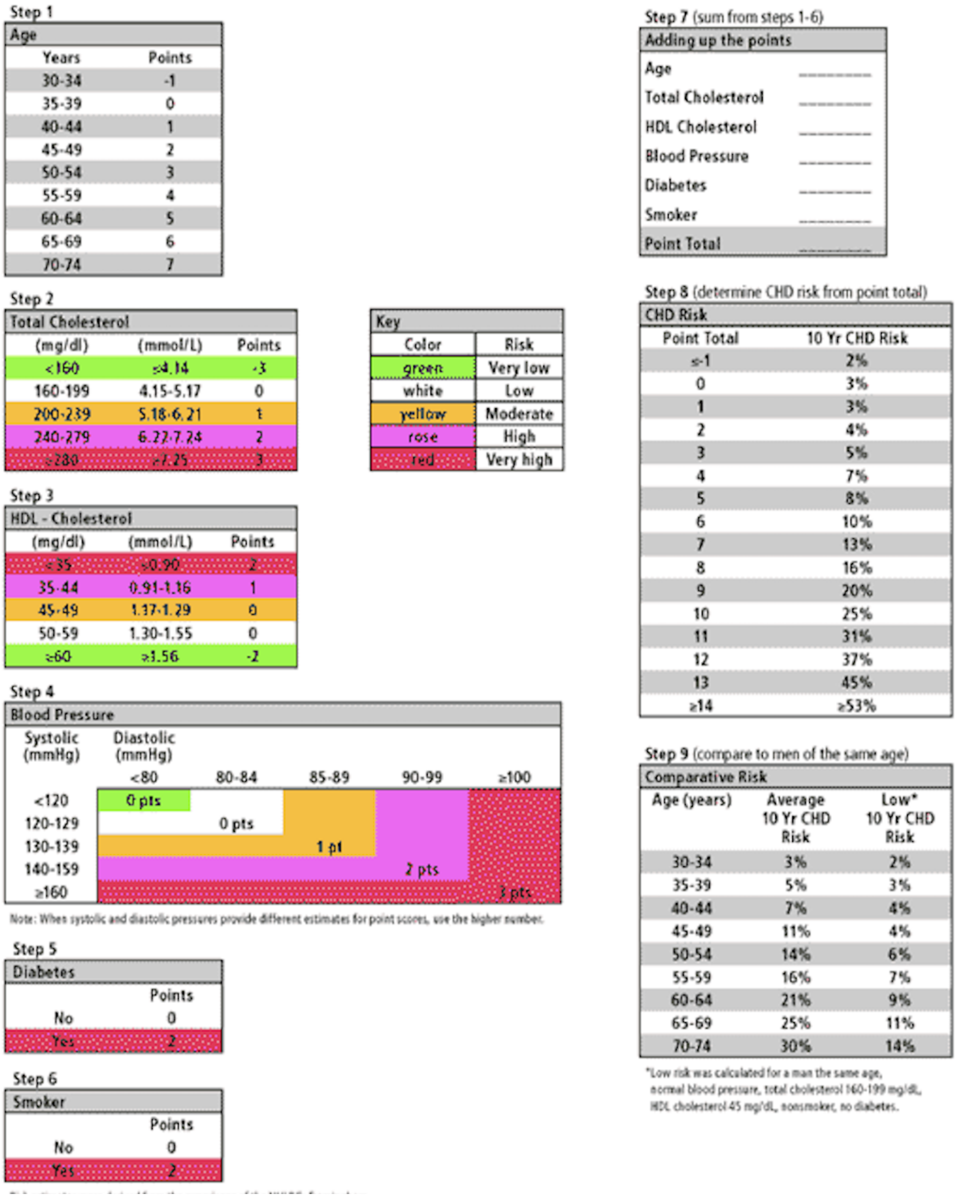
Table 2.6 shows that CVD Tree Classifier outperforms all the 10-Year CVD Risk prediction methods in the medical field with a AUC-Roc score of 0.82. Framingham Score, which is the most commonly used risk prediction score, only averages to about 0.73. Moreover, the biggest advantage of our CVD Tree Classifier over Framingham Risk Score is that we have full control over the false negatives as shown in figure 2.6 earlier. No other model allows this fine control over the number of false negatives.

Table 2.6: Comparison of CVD Tree Classifier vs Current 10-Year CVD Risk Prediction Scores

| # | **10-Year CVD Risk Prediction Models** | **AUC ROC Score** |
|---|---|---|
| 1 | Framingham (Prime Study France) | 0.68 |
| 2 | Framingham(Monica / Procam Study) | 0.78 |
| 3 | Framingham (CUORE Study) | 0.72 |
| 4 | CVD Tree Classifier | 0.82 |
| 5 | Score | 0.74 |
| 6 | Cuore | 0.74 |
| 7 | Assign | 0.73 |
| 8 | Qrisk | 0.76 |
| 9 | Procam(Prime Study France) | 0.64 |
| 10 | ACC/AHA Risk Calculator | 0.70 |

### *2.6.4 CVD Tree Classifier vs Framingham Risk Score on the same dataset*

This section compares the results of our CVD Tree Classifier with the Framingham risk score on the same dataset. The dataset described in table 2.2 is used to calculate the Framingham risk score for each row and then the accuracy and false negatives are calculated. The calculation used is based on the steps described in tables 2.7 through 2.14 [31] along with figure 2.8. These tables shows how each attribute from this dataset is used to calculate the risk score of a person. The combined steps from all these table are also listed in figure 2.7.



Step 1

| Age | |
|---|---|
| Years | Points |
| 30-34 | -1 |
| 35-39 | 0 |
| 40-44 | 1 |
| 45-49 | 2 |
| 50-54 | 3 |
| 55-59 | 4 |
| 60-64 | 5 |
| 65-69 | 6 |
| 70-74 | 7 |

Step 2

| Total Cholesterol | | |
|---|---|---|
| (mg/dl) | (mmol/L) | Points |
| <160 | ≤4.14 | -3 |
| 160-199 | 4.15-5.17 | 0 |
| 200-239 | 5.18-6.21 | 1 |
| 240-279 | 6.22-7.24 | 2 |
| ≥280 | ≥7.25 | 3 |

| Key | |
|---|---|
| Color | Risk |
| green | Very low |
| white | Low |
| yellow | Moderate |
| rose | High |
| red | Very high |

Step 3

| HDL - Cholesterol | | |
|---|---|---|
| (mg/dl) | (mmol/L) | Points |
| <35 | ≤0.90 | 2 |
| 35-44 | 0.91-1.16 | 1 |
| 45-49 | 1.17-1.29 | 0 |
| 50-59 | 1.30-1.55 | 0 |
| ≥60 | ≥1.56 | -2 |

Step 4

| Blood Pressure | | | | | |
|---|---|---|---|---|---|
| Systolic (mmHg) | Diastolic (mmHg) | | | | |
| | <80 | 80-84 | 85-89 | 90-99 | ≥100 |
| <120 | 0 pts | | | | |
| 120-129 | | 0 pts | | | |
| 130-139 | | | 1 pt | | |
| 140-159 | | | | 2 pts | |
| ≥160 | | | | | 3 pts |

Note: When systolic and diastolic pressures provide different estimates for point scores, use the higher number.

Step 5

| Diabetes | |
|---|---|
| | Points |
| No | 0 |
| Yes | 2 |

Step 6

| Smoker | |
|---|---|
| | Points |
| No | 0 |
| Yes | 2 |

Step 7 (sum from steps 1-6)

| Adding up the points | |
|---|---|
| Age | ________ |
| Total Cholesterol | ________ |
| HDL Cholesterol | ________ |
| Blood Pressure | ________ |
| Diabetes | ________ |
| Smoker | ________ |
| Point Total | ________ |

Step 8 (determine CHD risk from point total)

| CHD Risk | |
|---|---|
| Point Total | 10 Yr CHD Risk |
| ≤-1 | 2% |
| 0 | 3% |
| 1 | 3% |
| 2 | 4% |
| 3 | 5% |
| 4 | 7% |
| 5 | 8% |
| 6 | 10% |
| 7 | 13% |
| 8 | 16% |
| 9 | 20% |
| 10 | 25% |
| 11 | 31% |
| 12 | 37% |
| 13 | 45% |
| ≥14 | ≥53% |

Step 9 (compare to men of the same age)

| Comparative Risk | | |
|---|---|---|
| Age (years) | Average 10 Yr CHD Risk | Low* 10 Yr CHD Risk |
| 30-34 | 3% | 2% |
| 35-39 | 5% | 3% |
| 40-44 | 7% | 4% |
| 45-49 | 11% | 4% |
| 50-54 | 14% | 6% |
| 55-59 | 16% | 7% |
| 60-64 | 21% | 9% |
| 65-69 | 25% | 11% |
| 70-74 | 30% | 14% |

*Low risk was calculated for a man the same age, normal blood pressure, total cholesterol 160-199 mg/dL, HDL cholesterol 45 mg/dL, nonsmoker, no diabetes.

Figure 2.7: Framingham Risk Score Calculations

Table 2.7 is the first step in calculating the Framingham risk score. It shows the points scored for different age groups. The risk score starts from ages 30 and over as shown from the table.

Table 2.8 and Table 2.9 correspond to the second and third steps of the risk score calculation. These two tables show the number of points for different levels of

Table 2.7: Framingham Risk Score - Step 1 - Age

| Age in Years | Points |
|---|---|
| 30-34 | -1 |
| 35-39 | 0 |
| 40-44 | 1 |
| 45-49 | 2 |
| 50-54 | 3 |
| 55-59 | 4 |
| 60-64 | 5 |
| 65-69 | 6 |
| 70-74 | 7 |

Table 2.8: Framingham Risk Score - Step 2 - Total Cholesterol

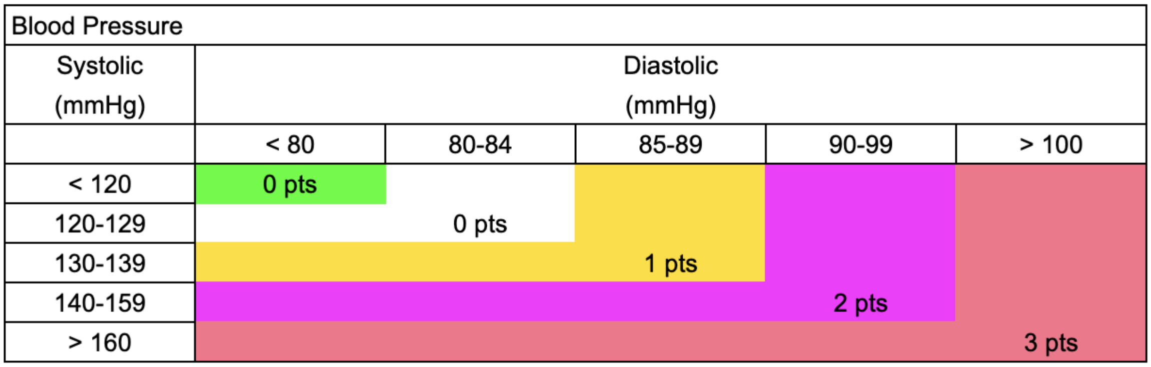
| (mg/dl) | (mmol/L) | Points |
|---|---|---|
| <160 | <4.14 | -3 |
| 160-199 | 4.15-5.17 | 0 |
| 200-239 | 5.18-6.21 | 1 |
| 240-279 | 6.22-7.24 | 2 |
| >280 | >7.25 | 3 |

cholesterol scores for total and HDL cholesterol levels.

The fourth step is listed in figure 2.8, which shows the points calculations at different levels of diastolic and systolic blood pressure levels. Table 2.10 and 2.11 show the points total for having diabetes and being a smoker respectively. These tables shows addition of 2 points each for the presence of diabetes and if the person is a smoker.

Table 2.9: Framingham Risk Score - Step 3 - HDL Cholesterol

| (mg/dl) | (mmol/L) | Points |
|---|---|---|
| <35 | <0.90 | 2 |
| 35-44 | 0.91-1.16 | 1 |
| 45-49 | 1.17-1.29 | 0 |
| 50-59 | 1.30-1.55 | 0 |
| >60 | >1.56 | -2 |



| Blood Pressure | | | | | |
|---|---|---|---|---|---|
| Systolic<br>(mmHg) | Diastolic<br>(mmHg) | | | | |
| | < 80 | 80-84 | 85-89 | 90-99 | > 100 |
| < 120 | 0 pts | | | | |
| 120-129 | | 0 pts | | | |
| 130-139 | | | 1 pts | | |
| 140-159 | | | | 2 pts | |
| > 160 | | | | | 3 pts |

Figure 2.8: Framingham Risk Score - Step 4 - Blood Pressure

The seventh step shown in table 2.12 shows the how all the points from the previous steps are added to come up with the points total. Table 2.13 shows how these points total correspond to the risk percentages. Finally, table 2.14 shows the comparative risk for different age groups for the risk percentage obtained from step 8. These steps are also summarized in figure 2.7.

Table 2.10: Framingham Risk Score - Step 5 - Diabetes

| Diabetes | Points |
|---|---|
| No | 0 |
| Yes | 2 |

Table 2.11: Framingham Risk Score - Step 6 - Smoker

| Smoker | Points |
|---|---|
| No | 0 |
| Yes | 2 |

Table 2.12: Framingham Risk Score - Step 7 - Adding up the points

| | |
|---|---|
| Age | Points from Age (step 1) |
| Total Cholesterol | Points from Total Cholesterol (step 2) |
| HDL Cholesterol | Points from HDL Cholesterol (step 3) |
| Blood Pressure | Points from Blood Pressure (step 4) |
| Diabetes | Points from Diabetes (step 5) |
| Smoker | Points from Smoker (step 6) |
| Points Total | Sum of column (step 7) |

Table 2.13: Framingham Risk Score - Step 8 - CVD Risk

| Point Total | 10 Yr CHD Risk |
|---|---|
| <-1 | 2% |
| 0 | 3% |
| 1 | 3% |
| 2 | 4% |
| 3 | 5% |
| 4 | 7% |
| 5 | 8% |
| 6 | 10% |
| 7 | 13% |
| 8 | 16% |
| 9 | 20% |
| 10 | 25% |
| 11 | 31% |
| 12 | 37% |
| 13 | 45% |

Table 2.14: Framingham Risk Score - Step 9 - Comparative Risk

| Age (years) | Average Risk | Low Risk |
|---|---|---|
| 30-34 | 3% | 2% |
| 35-39 | 5% | 3% |
| 40-44 | 7% | 4% |
| 45-49 | 11% | 4% |
| 50-54 | 14% | 6% |
| 55-59 | 16% | 7% |
| 60-64 | 21% | 9% |
| 65-69 | 25% | 11% |
| 70-74 | 30% | 14% |

The steps used to calculate the Framingham risk score explained in the previous tables were used on the Framingham Dataset. Table 2.15 shows the results from the current state of art, Framingham risk score, on the same dataset that the CVD Classifier was tested on. The calculations are performed at four different points total as shown in table 2.15. Each points total score (7,8,9 or 10) corresponds to a certain CVD Risk at which the person is predicted as high risk of CVD. This corresponds to a prediction of ”Yes” for high risk of 10-year CVD by the CVD Classifier.

The results from table 2.15 show that as points total increase, the accuracy increases but so does the number of false negatives. The best control over the number of false negatives by the Framingham risk score is at 34.3 percent with accuracy of 60.5 percent. The CVD Classifier is able to achieve accuracy of 82 percent at only 5.2 percent of false negatives, which shows it is able to easily outperform the current risk score setup. Even at points total of 10, the Framingham risk score accuracy is only

Table 2.15: Framingham Risk Score (FRC) Results

| **Scoring Technique** | **Accuracy (%)** | **False Negatives** | **Sensitivity (Recall)** | **Specificity** | **Precision** |
|---|---|---|---|---|---|
| FRC at Points cutoff 10 | 80.5 | 483 | 0.25 | 0.9 | 0.32 |
| FRC at Points cutoff 9 | 76.5 | 409 | 0.36 | 0.84 | 0.29 |
| FRC at Points cutoff 8 | 70.9 | 301 | 0.53 | 0.74 | 0.27 |
| FRC at Points cutoff 7 | 62.7 | 205 | 0.68 | 0.62 | 0.24 |
| CVD Classifier | 83.8 | 94 | 0.9 | 0.75 | 0.84 |
| Delta of last 2 rows | 21.1 | -111 | 0.22 | 0.13 | 0.6 |

78.8 percent with a huge 80 percent of false negatives, which pales in comparison to the numbers produced by the CVD Classifier.

## 2.7 Causal Analysis: Identifying high-risk Features for Diabetes

This section analyses the cause and effect relationship for diabetes using the Framingham Dataset. Even though this dataset is primarily used to predict cardiac disease, it can further help advance the medical field by some useful insight into diabetes. The analysis here includes identifying how different attributes from this data have an effect on diabetes similar to the analysis performed for cardiac disease in section 2.4.

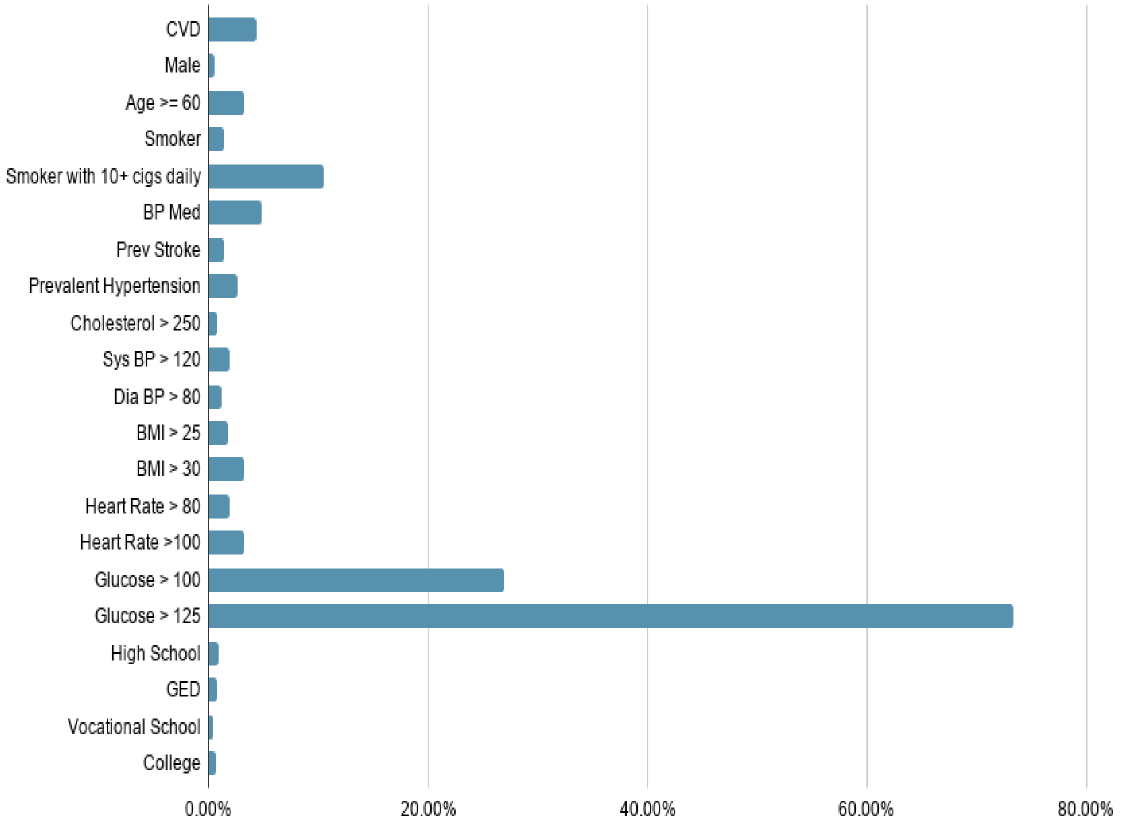


Figure 2.9: Percentage Difference (Significance) for each attribute that causes Diabetes

Table 2.16 shows the results from this analysis. This table shows that the biggest factor, as shown from the difference column, is on glucose levels over 100 and 125. The glucose direct co-relation with diabetes is expected, however it is interesting to see the jump in percentage difference from 26.94 percent to 73.33 percent as the glucose level jumps from 100 to 125. The next big attribute is ”smoker” with over 10 cigarettes per day. It is interesting to note that a smoker who smokes less than 10 cigarettes per day only has a 1.4 increased risk of diabetes. Another important observation to notice is that even though the presence of diabetes increased the risk of CVD by

22.4 percent as observed in table 2.3, the presence of CVD only increased the risk of diabetes by 4.3 percent. Figure 2.9 represents this visually with bar graphs making it easier to identify the most significant attributes responsible for diabetes based on this dataset.

Table 2.16: Probability that you have high risk of Diabetes given the presence or absence of these attributes

| **Attribute** | **Present** | **Absent** | **Difference** |
|---|---|---|---|
| CVD | 6.2% | 1.9% | 4.3% |
| Male | 2.86% | 2.36% | 0.5% |
| Age >= 60 | 5.36% | 2.06% | 3.3% |
| Smoker | 1.86% | 3.26% | 1.4% |
| Smoker with 10+ cigs daily | 1.95% | 12.40% | 10.45% |
| BP Med | 7.26% | 2.41% | 4.85% |
| Prev Stroke | 4.0% | 2.56% | 1.44% |
| Prevalent Hypertension | 4.40% | 1.74% | 2.66% |
| Cholesterol >250 | 3.11% | 2.30% | 0.81% |
| Sys BP >120 | 3.19% | 1.30% | 1.89% |
| Dia BP >80 | 3.07% | 1.97% | 1.1% |
| BMI >25 | 3.36% | 1.56% | 1.8% |
| BMI >30 | 5.41% | 2.12% | 3.29% |
| Heart Rate >80 | 3.69% | 1.85% | 1.84% |
| Heart Rate >100 | 5.71% | 2.49% | 3.22% |
| Glucose >100 | 27.48% | 0.54% | 26.94% |
| Glucose >125 | 74.42% | 1.09% | 73.33% |
| High School | 3.49% | 2.6% | 0.89% |
| GED | 1.84% | 2.6% | 0.76% |
| Vocational School | 2.23% | 2.6% | 0.37% |
| College | 1.90% | 2.6% | 0.7% |

## 2.8 Summary and Contributions

One of the research objectives (RO4) proposed in this dissertation aimed to improve the quality of prediction of cardiac disease using classification. The discussion below shows that we have achieved this objective. The prediction results have also been compared on the same dataset of the CVD Tree Classifier versus Framingham risk score, which show that our classifier is able to outperform the risk score.

The results show that CVD Tree Classifier outperforms not only other machine learning classifiers but also all the current state-of-art 10-Year CVD Prediction Risk scores. We have trained and optimized our CVD tree classifier to predict the risk of a 10-Year CVD with not only improved accuracy and AUC-ROC score, but also reduced underestimation as compared to other methods. This NP Lemma based approach is able to predict cardiac disease with over 85 percent accuracy. Additionally, it cuts down the false negatives to under 10 percent. It also has the ability to easily reduce the false negatives further at a cost of reduced accuracy, which may be acceptable depending on this use case.

The methods used to create this CVD Tree Classifier can be easily expanded to work with any classification problem where there is a need for asymmetric error control. This approach can have a massive impact in the medical domain, especially in disease diagnosis, where we typically need to control the number of false negatives.

To conclude, the accurate 10-Year CVD risk predictions by the CVD Tree Classifier can have a massive impact in the cardiac domain, since early prevention can save a lot of human lives. With an improved accuracy in predicting CVD, this tree-based classifier model with asymmetric error control can reduce the burden of CVD in populations and improve the quality of life as well as life expectancy in individuals with CVD.

# Chapter 3

# ASYMMETRIC ERROR CONTROL FOR BINARY CLASSIFICATION

## 3.1 Background

Asymmetric error control implies unequal distribution and weights assigned to each error. False positives and false negatives can have unequal costs based on the problem. This dissertation builds upon our work [32] on asymmetric error control with a more detailed evaluation across different large scale data-sets for classification. Classification [44], is a part of supervised learning [46] that aims to automatically predict and classify new data samples after training on labelled data. Disease diagnosis, image classification and automatically recognizing spam email are some examples of classification [45]. Binary classification [47], which is the most common sub type of classification, can only have two values as class labels. Most binary classification models [48] do not naturally provide control over the number of false negatives, as they are optimized for accuracy.

The focus of this section is to provide asymmetric error control for binary classification applications. There are two main challenges for binary classification in medical disease diagnosis [49]. Firstly, the cost of one error greatly outweighs the other, highlighting the need for asymmetric control. For example, predicting if a patient is going to have a cardiac disease is a binary classification problem where the cost of misclassifying a patient with high risk as no risk (False Negative) has a much bigger penalty than misclassifying a patient with no risk as high risk (False Positive). The former could cost a life whereas the latter does not have such severe consequences. Traditional machine learning models [51] may not be ideal in this scenario as they do not provide a mechanism to control the number of false negatives below a certain threshold with a theoretical guarantee [52]. Even if these models result in improved accuracy and reduced classification error, they may not be optimal since the cost of one error greatly outweighs the other.

As a proposed solution to this problem, we have created a tree-based classifier that can control the number of false negatives below a specified threshold value with a theoretical guarantee that the false negatives will not exceed this value. We name this classifier as Asymmetric Error Control (AEC) Tree Classifier as it is able to provide control over one type of error. The theoretical foundation for this model is based on Neyman-Pearson (NP) Lemma [53], which shows that the likelihood ratio test is the most powerful test in hypothesis testing. Based on the NP Classification umbrella implementation [44], this concept is expanded to create this tree-based classifier that can control the false negatives to a given value, while still providing comparable accuracy on different datasets.

The imbalance in the medical disease diagnosis datasets is the second main challenge for binary classification in medical disease diagnosis. This imbalance arises because a positive disease diagnosis is typically a rare event compared to a negative disease diagnosis. This leads to an imbalanced dataset where the negative class outweighs the positive class. This problem is usually addressed using Synthetic Minority Over-sampling Technique (SMOTE) [58] which can balance the dataset. However, oversampling the minority class can lead to over-fitting as well as increasing the learning time of an algorithm. Similarly, under sampling of the majority class can lead to the removal of some important data points. To avoid the overuse of SMOTE, we use Hellinger distance as the splitting criteria in our decision tree because it addresses the imbalance in the dataset by quantifying the difference between two probability distributions and eliminates the need of extensive sampling techniques in binary classification.

To solve these two challenges, greater cost of a false negative compared to false positive and imbalanced datasets in binary classification is the motivation in the creation of the AEC Tree classifier. This classifier is able to predict the risk of future and present cardiac disease with asymmetric error control. This enables us to limit the number of false negatives, where each false negative could be as costly as a human life. Moreover, the asymmetric error control is also tested on a large dataset from a different domain and it is able to achieve the same error control. The classifier is also comparable to the state of the art classifiers in terms of accuracy and AUC-ROC scores on these datasets, despite providing control on the number of false negatives. The next section discusses the related work that has been done to achieve asymmetric error control. The dissertation, then discusses the methodology behind the creation of the AEC Classifier and concludes with the evaluation and results on three diverse applications, and their datasets.

## 3.2 Related Work

This section has been divided into four main parts. The first part discusses traditional binary classifiers and how they use threshold moving to adjust the error costs. The second part goes over cost sensitive learning and ROC curves. Next, this section discusses NP classification and concludes with decision trees in medical domain.

### *3.2.1 Traditional Binary Classifiers and Threshold Moving*

The current state of the art binary classifiers algorithms are optimized to minimize the classification error [66] instead of providing asymmetric error control. The classification error [49] is calculated as shown in equation 2.1 in chapter 2.

Equation 2.1 shows that we can control the classification error using the traditional classification techniques but there is no way to limit one type of error. This is because the false negatives and false positives are grouped together in this equation.

The only way to control one type of error with the traditional binary classifiers is using threshold moving. A classification threshold is the value above which the class is predicted as positive. The default value of the classification threshold is 0.5 for values normalized between 0 to 1. This value can be adjusted as part of threshold moving to a lower value which will reduce the false negatives. However, this approach is only valid for empirical data as there is no probabilistic guarantee that it will be valid for population data.

### *3.2.2 Cost Sensitive Learning and ROC Curve*

Currently, the classification algorithms have two other main ways of asymmetric error control. The first is using cost sensitive learning. In this algorithm, the model takes the cost of prediction errors into account, while getting trained on the training dataset. However, this approach has three shortcomings for our purpose. Firstly, this algorithm does not provide any consensus way to assign the cost of each error. This leads to a lot of variability and inconsistent results. Secondly, this learning still does not provide any mechanism for directly controlling the number of false negatives below a certain threshold. There is no guarantee that the prediction will minimize the false negatives. Lastly, there is the ethical dilemma, in assigning costs to real life data samples. For example, some may consider it unethical to assign more value to the life of a younger person compared to an older person while assigning costs in cost sensitive learning. Similarly, in disease diagnosis, the life of a patient is priceless and we never want to cruelly set a value for it and sacrifice potential lives for the algorithm to maximize its interest [3].

The second way of asymmetric error control is provided using ROC curve [4]. The

ROC space is defined as a two dimensional space where horizontal and vertical axes correspond to false positive and false negative errors [2]. The scoring function of a binary classification model can be estimated using the ROC Curve. This concept has been extended to control one type of error. However, this has only been successful to control empirical error and has no success using population error [3]. This means that for a random sample from the population, there is no guarantee that the false negatives will be controlled using the ROC curve.

### *3.2.3 NP Classification*

The Neyman-Pearson (NP) classification paradigm is a binary classification paradigm that aims to address asymmetric errors in machine learning [34]. Their research explores the practicality of the NP classification paradigm and evaluation of these classification methods. The main contribution of their research is how to evaluate and compare the performance of different NP classification methods. It proposes NP receiver operating characteristic (NP-ROC) bands, a variant of ROC, as a new visualization tool for NP classification. Some possible use cases of NP-ROC bands include evaluating the threshold (alpha) in a data adaptive way and comparing different classifiers.

Another contribution of the NP classification paradigm is that it proposes an umbrella algorithm, which can help implement some classification methods under the NP paradigm. However, this implementation has some limitations. Firstly, it only provides implementation of the three main classification methods (logistic regression [39], support vector machines [40], and random forests [41]) with no details on how to make it extensible for other classification models. Secondly, this algorithm does not provide a way to swap between the control of false negatives and false positives. Each dataset will be suited to only one type of error control with the other control not possible without altering the hypothesis and the dataset. This makes it very difficult to apply this implementation to multiple datasets. Lastly, another considerable limitation of this implementation is that it is only valid for population error control if the dataset and the implementation is truly random, which may not be the case for all classification models that are made NP compatible by the umbrella algorithm.

### *3.2.4 Decision Trees in Medical Domain*

In medical disease diagnosis, it is difficult to obtain large amounts of data, since each row could correspond to the actual patient's medical record. Along with the scarcity of data, the data imbalance is also very common in disease diagnosis. As stated in the Journal of Healthcare Engineering, "Identifying rare but significant healthcare events in massive unstructured datasets has become a common task in healthcare data analytics. However, imbalanced class distribution in many practical datasets greatly hampers the detection of rare events, as most classification methods implicitly assume an equal occurrence of classes and are designed to maximize the overall classification accuracy" [11]. Similarly, in predicting cardiovascular disease (CVD), which is the problem chosen for this research, we have a lot more negative class (No Risk) compared to positive class (High risk of CVD) in our dataset. Another issue with medical dataset is the handling of missing values. It is challenging to rerun experiments with patients involved which makes it very difficult to recollect missing data.

Decision trees [65] are generally considered to perform best with an imbalanced, small dataset with missing values. Due to the nature of the medical dataset we decided to proceed with the decision trees. The results section vindicate our decision as decision tree based classifiers outperform the traditional classification models on our cardiac patient dataset.

### *3.2.5 Tree Splitting Criterion*

Np classification umbrella algorithm only implements one decision tree classifier, which is random forest. However, even though random forest picks random samples from the dataset to develop each decision tree from a bootstrap sample of the training dataset [10], it does not choose the splitting criteria of the tree randomly. The splitting point for the npc implementation of random forest is chosen by Gini impurity [43] which is defined as:

$$GiniImpurity = \sum_{i=1}^{k} p_{\mathrm{i}}(1 - p_{\mathrm{i}})$$

where,

k is the number of classes

$p_{\mathrm{i}}$ is the proportion of cases belonging to case i.

This equation shows that Gini Impurity is not optimized for unbalanced datasets as it is dependant on k and $p_{\mathrm{i}}$. As mentioned earlier, an unbalanced dataset is common characteristic of a medical dataset. Another tree based splitting criterion, which is commonly used is called Entropy or information gain. Its equation is calculated as:

$$Entropy == -\sum_{i=1}^{k} p_{\mathrm{i}} \log(p_{\mathrm{i}})$$

where,

k is the number of classes

$p_{\mathrm{i}}$ is the proportion of cases belonging to case i.

The equation shows that Entropy is also similar to Gini Impurity and does not account for the data imbalance. The classifier created as part of this research, AEC Tree Classifier, is optimized to address the imbalance without the need of extensive sampling, since it uses Hellinger distance as the splitting criterion. Hellinger distance is calculated as:

$$h(P, Q) = \frac{1}{\sqrt{2}} \cdot \|\sqrt{P} - \sqrt{Q}\|_2$$

where h(P,Q) is the Hellinger distance between 2 probability distributions P and Q. The complete hellinger distance equation and derivation is listed in Appendix B.

Tree based classifiers typically use Gini Index or Entropy as their splitting criterion. However, both favor splits that can result in an uneven class distribution over splits which will lead to an even distribution of the classes. This works well for balanced data but is a major problem in an imbalanced dataset. This is because a split, which will produce 90-10 class balance in the children nodes will get high Gini and Entropy scores, but it may not improve the class separation in any way. As part of this research we use Hellinger distance, instead of Gini Index or Entropy Information Gain, as the tree splitting criterion for the AEC Tree Classifier due to the imbalance in the dataset. Appendix D shows how Hellinger distance performs compared to Gini Index and Entropy on balanced and imbalanced datasets for different tree classifiers. Appendix D also shows how hellinger distance is able to outperform Gini and Entropy on an imbalanced dataset.

The results from Appendix D show that Hellinger distance accuracy does not drop much as data gets imbalanced compared to Gini and Entropy. We notice that even though Gini and Entropy may perform better than Hellinger distance on balanced data, they may not be the best choice for imbalanced data. Since our original data is not balanced, Hellinger distance is a better option as it does not require the use of extensive sampling techniques. These sampling techniques are being used to balance the data using SMOTE, which requires oversampling the minority class and under sampling the majority class. Under sampling can cause removal of important data, whereas oversampling can lead to overfitting so balancing the datasets with sampling may not be the best option. The results section later justifies our decision in choosing Hellinger distance over Gini and Entropy information gain on imbalanced datasets.

## 3.3 Methodology

The AEC Tree Classifier is based on ensemble [63] decision tree classifiers [56]. It consists of a large number of individual decision trees that operate together as an ensemble [26]. Each tree in the Classifier produces a class prediction and the model's predicted class is selected based on the class that has received majority of votes.

Similar to popular ensemble decision trees, such as Random Forest [28] and extra trees, our classifier is trained over various sub-samples of the data and each tree is grown to its largest [57]. However, the most critical point of decision tree induction algorithms is the choice of the splitting criteria [60] of a node. We use Hellinger distance as the splitting criteria because it addresses the imbalance in the dataset by quantifying the difference between two probability distributions. This eliminates the need of extensive sampling techniques and hyper tuning that are required using traditional splitting criterion's such as Gini Index and Entropy. Hellinger distance as a function is calculated for all attributes and it provides the highest value of split measure for our feature set.

### *3.3.1 Traditional vs AEC Tree Classifier*

For traditional classifiers, only the scoring function needs to be constructed from the training data, the threshold is chosen as 0.5 by default. However, for the AEC Tree Classifier, both the scoring function and the threshold value is constructed from the training data. Figure 3.1 and 3.2 summarize this for the traditional and AEC Tree Classifier respectively. Figure 3.1 clearly shows that the training data is being used as input only to the scoring function whereas in Figure 3.2, the training data is being used to calculate the threshold. This classifier is then adapted to control the false negatives below a certain threshold using the NP classification paradigm.

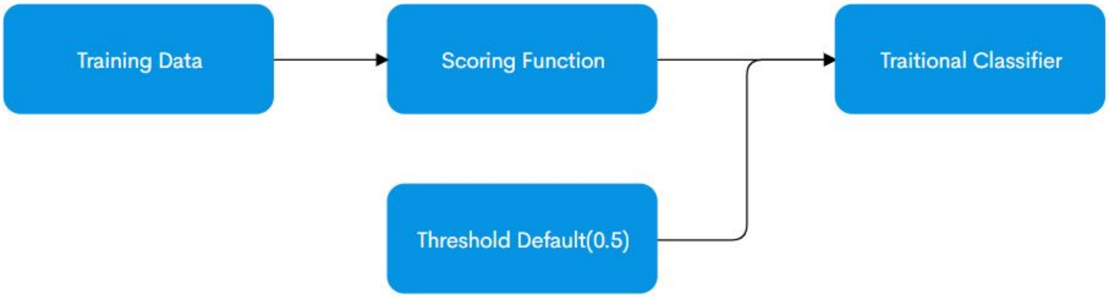


Figure 3.1: Traditional Classifier

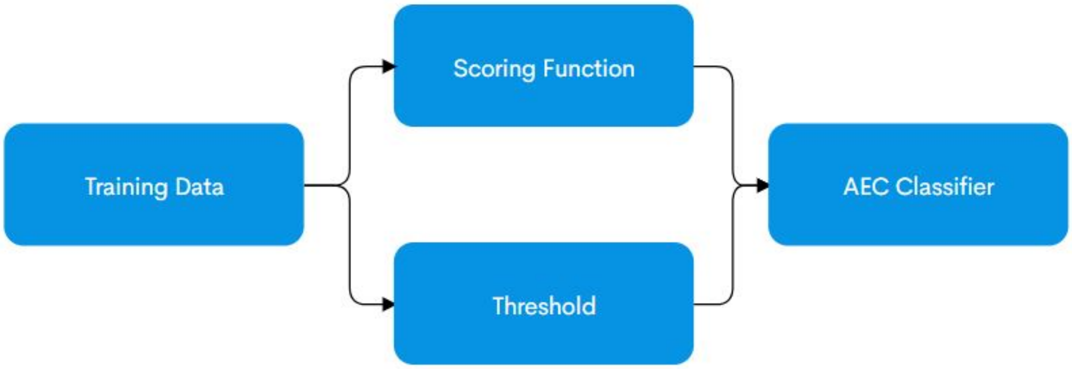


Figure 3.2: AEC Tree Classifier

### *3.3.2 Neyman-Pearson Lemma*

The NP classification is based on the NP Lemma. The complete lemma and it's proof is listed in Appendix A. The lemma states that the likelihood test is the most powerful hypothesis testing among all other tests [54]. Given that X is the data or the observation, $H_0$ is null hypothesis and $H_1$ is alternative hypothesis, then P(X, $H_0$) is probability distribution of X given $H_0$ and P(X, $H_1$) is probability distribution of X given $H_1$.

This leads us to the likelihood ratio test L(X) which states:

$$L(X) = \frac{P(X_1, H_1)}{P(X_1, H_0)} > \gamma$$

where $\gamma$ (gamma) is threshold gamma. To maximize $P_D$ for given $P_{FA} = \alpha$ where $P_D$ is probability of detection and $P_{FA}$ is probability of false alarm. The given alpha $\alpha$ is the upper bound threshold value. $P_{FA}$ is then calculated as:

$$P_{FA} = P_r L(X) > \gamma, H_0 = \alpha$$

which is the probability that the likelihood ratio is greater than gamma under the null hypothesis $H_0$.

### *3.3.3 AEC Classification*

The first step is sample splitting which involves splitting the training data into three parts as explained in figure 3.3 below. The first part consists of mixed classes of 0 and 1 samples. This data is fed as input to the AEC Tree Classifier to create the scoring function. This scoring function is then used on the left out class 0 samples to create a list of classification scores.

From this list of classification scores, a threshold is chosen using Order Statistics, such that the violation rate (that is, the probability that the type 2 error exceeds the user-specified upper bound alpha) is minimized. We chose the smallest value of the threshold (alpha) on the classification scores such that the violation rate remains minimized. To find the threshold, brute force and bagging were tried initially but order statistics was proven to be the best in finding this threshold value from the

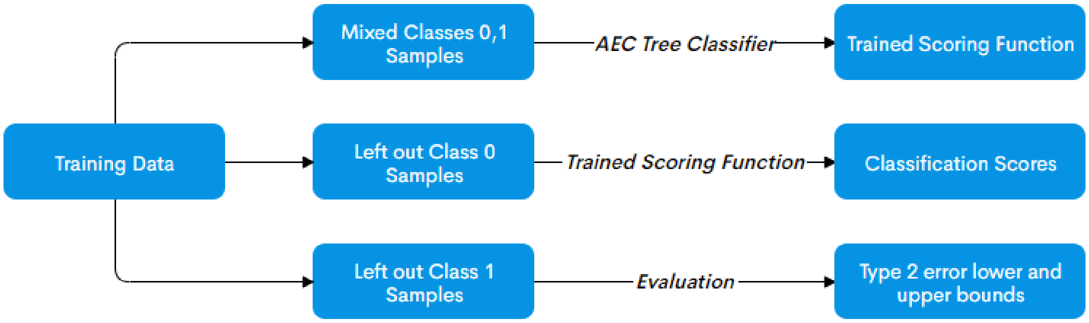


Figure 3.3: Sample Splitting on Training Data

classification scores [55]. The probability that the type 2 error exceeds alpha is controlled by a user specified tolerance parameter $\delta$.

$$\mathbb{P}\left[R_0\left(\varphi^c\right) > \alpha\right] \leq \delta$$

This Equation shows that the probability that the violation rate exceed alpha is bound by the user specified tolerance parameter. The threshold chosen from the list of classification scores is chosen from the left-out class 0 samples which was not used to train the AEC Tree Classifier as we saw during sample splitting in figure 3.3. The left out class 1 samples are used to limit the false positive error bound. The three steps above are incorporated in our decision tree classifier to create the AEC Tree Classifier that is able to control the false negatives below a certain threshold with high probability.

### *3.3.4 Constraints*

There are three main constraints to the Neyman-Pearson based AEC Tree Classifier that we created in the previous steps.

### Violation Rate

The upper bound on the type 2 error achieved by our classifier is violated as described by the violation rate, which is calculated as:

$$V(k) = \sum_{j=k}^{n} (j^n)(1-\alpha)^j \alpha^{n-j}$$

where V(k) is violation rate, which is the probability that the false negatives exceed the threshold value and n is the sample size. $\alpha$ is the threshold which is the upper bound on the percentage of false negatives and K is the list of threshold scores. The violation rate only depends on the sample size and the rank threshold K.

### Optimal Order

To minimize the number of false positives, such that the false negatives are under alpha, the optimal order is denoted as shown in the equation below. k* is the minimal threshold whose violation rate is under $\delta$ such that:

$$\mathrm{k}^* = \min\{\mathrm{k}- > (1, \ldots \mathrm{n}) : \mathrm{v(k)} <= \delta\}$$

### Sample Size

There is a minimum sample size needed to guarantee the upper bound on false negatives which is:

$$n \geq \log \delta / \log(1-\alpha)$$

where,

n is the minimum sample size

$\alpha$ is the threshold

$\delta$ is the tolerance parameter

Therefore, as long as the sample size is greater than n, the control can be achieved on type 2 error control. If the sample size is less than n, then the violation rate can not be controlled. As long as these constraints are respected, this AEC Tree classifier can be used in many areas of medical diagnosis [64], where the cost of one type of error greatly outweighs the other and there is a need for asymmetric error control.

## 3.4 Evaluation

This section is divided into two main parts: The first part goes over the details of each dataset used to evaluate the classifier. The second part describes the experimental setup and the range of hyper parameters used to run the classifier.

### *3.4.1 Datasets*

Three different datasets were used as part of the evaluation. Since the best use case of asymmetric error control is applicable in medical disease diagnosis, two of the three datasets were related to it. More specifically, the medical datasets focused on cardiac disease, since it is the leading cause of death worldwide [59].

For the first dataset used in evaluating our classifier, we obtained the publicly available dataset of the patients involved in the Framingham Heart Study [59]. This includes patient records of over 4200 patients and 15 attributes. The features of this dataset are shown in table 3.1. The data obtained is from a cardiovascular study on the residents in the town of Framingham in Massachusetts. The classification goal from he dataset is to predict whether the patient has 10-year risk of Cardiovascular Disease (CVD). Using this dataset, we trained and optimized our machine learning model to predict the risk of a ten-year CVD, with not only comparable accuracy and

AUC-ROC score, but also reduced underestimation.

| **Attributes** | **Value** |
|---|---|
| Sex | Male or Female |
| Age | Age of the patient |
| Education | 1 = Some High School;<br>2 = High School or GED;<br>3 = Some College or Vocational School;<br>4 = College |
| Current Smoker | Whether or not the patient is a current smoker |
| Cigs Per Day | The number of cigarettes that the person smoked on average in one day |
| BPMeds | Whether or not the patient was on blood pressure medication |
| PrevalentStroke | Whether or not the patient had previously had a stroke |
| PrevalentHyp | Whether or not the patient was hypertensive |
| Diabetes | Whether or not the patient had diabetes |
| TotChol | Total cholesterol level |
| SysBP | Systolic blood pressure |
| DiaBP | Diastolic blood pressure |
| BMI | Body Mass Index |
| HeartRate | Heart Rate |
| Glucose | Glucose Level |

Table 3.1: Dataset Features used as input to Predict the 10-year risk of CVD

The second dataset [61] aims to predict the presence of cardiac disease in the patient. The difference from the first dataset is that it predicts the immediate presence or absence of cardiac disease, rather than predicting the 10-year risk of cardiac disease. Moreover, this data consists of 70,000 patient records and 11 features. These features are listed in table 3.2. This dataset allows us to test out the AEC classifier on a larger cardiac disease dataset to verify how the algorithm results are affected as the data is increased. The results show that the control over the number of false negatives is still maintained on the larger cardiac disease dataset.

| **Attributes** | **Value** |
|---|---|
| Age | Number of days (int) |
| Height | Height in cm (int) |
| Weight | Weight in kg (int) |
| Gender | Categorical code |
| Systolic blood pressure | int value |
| Diastolic blood pressure | int value |
| Cholesterol | 1: normal, 2: above normal, 3: well above normal |
| Glucose | 1: normal, 2: above normal, 3: well above normal |
| Smoking | Yes or No (Binary) |
| Alcohol intake | Yes or No (Binary) |
| Physical activity | Yes or No (Binary) |

Table 3.2: Dataset Features used as input to Predict Cardiac Disease

The final dataset is used to provide some context to the evaluations outside medical diagnosis, to ensure that the results remain intact across diverse large scale datasets. This dataset contains 10 years of daily weather observations from numerous Australian weather stations. This is a large scale dataset with over 142,000 records.

Table 3.3 shows the main attributes used from this weather dataset to train the AEC Classifier. The classification goal of this dataset is to predict, if it will rain tomorrow. A successful implementation of asymmetric error control on this dataset demonstrates the ability of our AEC Classifier to control the number of false negatives across different domains and data samples.

| **Attributes** | **Value** |
|---|---|
| Date | The date of observation |
| Location | The common name of the location of the weather station |
| MinTemp | The minimum temperature in degrees celsius |
| MaxTemp | The maximum temperature in degrees celsius |
| Rainfall | The amount of rainfall recorded for the day in mm |
| Evaporation | The evaporation (mm) in the 24 hours to 9am |
| Sunshine | The number of hours of bright sunshine in the day. |
| WindGustDir | The direction of the strongest wind gust in the 24 hours to midnight |
| WindGustSpeed | The speed (km/h) of the strongest wind gust in the last 24 hours |
| Pressure9am | Atmospheric pressure (hpa) reduced to mean sea level at 9am |
| Cloud9am | Fraction of sky obscured by cloud at 9am. (oktas) |
| RainToday | Boolean: 1 if precipitation(mm) in the 24 hours exceeds 1mm, else 0 |

Table 3.3: Dataset Features used as input to Predict Rain Tomorrow

The three datasets are summarized in table 3.4. Each row corresponds to the dataset mentioned in the previous three tables and also lists the number of records found in each dataset. The table shows that the weather dataset is the biggest followed by the cardiac disease datasets.

| | **Dataset** | **Number of Records** |
|---|---|---|
| 1 | Predict 10-Year CVD Risk | 4241 |
| 2 | Predict Cardiac Disease | 70,000 |
| 3 | Predict Rain tomorrow | 142,194 |

Table 3.4: Datasets used to evaluate AEC Tree Classifier

### *3.4.2 Experimental Setup*

We used the latest version of Python to code our classifier with PyCharm as the integrated development environment (IDE). After construction of our tree-based classifier, we made it NP lemma compatible by enforcing the restraints as explained in methodology section.

The NP classification provides a way for asymmetric error control on type 1 error, which is the number of false positives. However, in the prediction of CVD, we need to control the type 2 error which is the number of false negatives. Using the same theoretical foundation based on the NP lemma, we flipped our predicted variable column in the training set. All the 1 classes were changed to 0 and all the 0 classes were changed to 1. This meant that our null and alternative hypothesis was also swapped which enabled us to use the same algorithm to control the type 2 error instead of type 1. This also meant that we had to adjust the calculation of the precision and recall from the confusion matrix.

There were some additional challenges in the first cardiac disease dataset related to imbalance and missing values. The dataset has a lot more negative classes as compared to positive classes. A simple naive model that returns all samples as negative could have high accuracy. To address this, Synthetic Minority Over-sampling Technique (SMOTE) [58] was used in addition to using hellinger distance as the tree splitting criterion to balance the dataset. With a more balanced dataset, the accuracy's of different models are more meaningful.

There were some values missing from the dataset in certain rows. Typically, rows with missing values are removed from the dataset. However, in the medical domain, we often do not have the luxury of a massive dataset, as each row represents an actual patient's medical record which makes it very difficult to re-run experiments to fill the missing value. Therefore, in order to maintain the row count of the original sample, the average of each column was calculated. This average was replaced in each column to fill in the missing value in order to complete the row.

70 percent of the dataset was used in training and the remaining 30 percent was used in testing. The number of trees in each forest was chosen to be 300. The minimum number of samples to split the node was chosen as 2 and the minimum number of samples required to be at leaf node was 1. The rest of the parameters were kept the same as the default values of decision tree algorithms.

The experiment was run multiple times and the average from the results were plotted to account for any bias. For a comprehensive evaluation, we compared the accuracy, AUC-ROC score, precision and the recall rate of our classifier versus the traditional classification models.

## 3.5 Results

The results for all the datasets from table 3.4 are provided in this section. For each dataset, we first show how the AEC Tree Classifier performs in comparison with the other common machine learning classifiers. Secondly, we show how our classifier is providing asymmetric error control and how we can control the number of false negatives using the threshold value.

### *3.5.1 Dataset 1: Predict 10-Year CVD Risk Dataset*

We compared the AEC Tree Classifier performance on the Framingham dataset with other machine learning classifiers as shown in figure 3.4. We compare the accuracy, AUC-ROC score, precision, recall and the number of false negatives for each classifier. All the results shown have been adjusted to percentages for simplicity and table 3.5 shows the exact values used to plot figure 3.4.

**Comparison with Traditional Binary Classifiers**

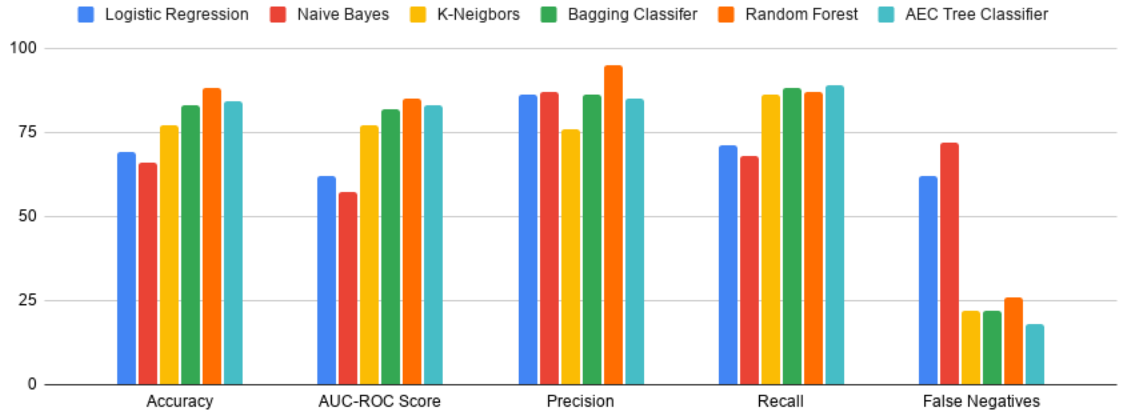


Figure 3.4: Results from Dataset 1 - AEC Tree Classifier versus ML Classifiers to predict 10 year risk of CVD

Table 3.5: Dataset 1 - AEC Tree Classifier vs ML Classifiers

| Classifiers | Accuracy | AUC-ROC | Precision | Recall | False Negatives |
|---|---|---|---|---|---|
| Logistic Regression | 69 | 62 | 86 | 71 | 62 |
| Naive Bayes | 66 | 57 | 87 | 68 | 72 |
| K-Neigbors | 77 | 77 | 76 | 86 | 22 |
| Bagging Classifer | 83 | 82 | 86 | 88 | 22 |
| Random Forest | 88 | 85 | 95 | 87 | 26 |
| AEC Tree Classifier | 84 | 83 | 85 | 89 | 18 |

### Accuracy

For this dataset, the tree based classifiers perform the best in terms of accuracy with Random Forest Classifier leading the chart with accuracy at 88 percent. The AEC Tree Classifier and bagging classifier, which is based on decision trees as the base estimator, follow with accuracy in excess of 80 percent. The traditional non-tree based classification algorithms fall behind as shown in the figure. These results can be explained by the ability of decision trees to perform better on an imbalanced dataset by balancing error in class populations. The results vindicate our decision to pick a tree based classifier as decision tree based classifiers outperform the traditional classification models on our 10-years cvd risk dataset.

The accuracy of correct predictions alone may not be a perfect metric to measure the effectiveness of classifiers. In an imbalanced dataset, the accuracy can be a little misleading, since any algorithm that returns the majority class could have the highest accuracy. This makes it essential to compare our classifiers with additional metrics as shown below.

### AUC-ROC Score

AUC-ROC is explained by the Data Science Journal as “It is a performance measurement for classification problems at various thresholds settings. ROC is a probability curve and AUC represents degree or measure of separability. It tells how much a model is capable of distinguishing between classes. Higher the AUC, better the model is at predicting 0s as 0s and 1s as 1s. By analogy, higher the AUC, better the model is at distinguishing between patients with disease and no disease.[50]”.

Figure 3.4 shows that AEC Tree Classifier, along with random forest, outperforms other machine learning with an AUC-ROC score over 80 percent. The bagging classifier follows with third best AUC scores also in excess of 80 percent. This score is an especially important validation for our AEC Tree Classifier because of imbalance in our first dataset.

### Recall and False Negatives

Recall is one of the best measures to calculate how many true positives our models can capture. This makes recall one of the best metrics, where there is a high cost associated with a false negative. Recall rate is inversely proportional to the number of false negatives. This means higher the recall rate, the lower the number of false negatives. This is one of the most significant metrics for our first two datasets as they relate to cardiac disease prediction. In this scenario, a false negative means that the classifier failed to diagnose a patient which had a high risk of cardiac disease. This could potentially cause a loss of a human life, since the patient will not get any treatment.

Our AEC Tree Classifier has been designed to keep the false negatives to a minimum and recall rate to the maximum, using the theoretical approach from the NP

Lemma which was explained previously. The recall results from figure 3.4 clearly show that AEC Tree Classifier outperforms all other classifiers in terms of recall rate across all the datasets.

Random Forest Classifier, which was exceeding the previous metrics, falls behind to AEC Tree Classifier in terms of the recall rate, which does not make it a good fit for control over the number of false negatives. The AEC Tree Classifier is able to achieve a recall rate close to 90 percent. In other words, it is able to identify 90 percent of the patients with high risk of 10-year CVD. Moreover, the false negatives of AEC Tree classifier are lowest compared to all other classifier. Random forest has almost 8 percent more false negatives compared to our classifier, which does not make it the best fit to predict cardiac disease.

### Precision

To fully evaluate the AEC Tree Classifier, it is important to measure both precision and recall. As we improve recall, precision usually suffers. However, our classifier falls only behind random forest in terms of precision. The precision of 85 percent from our classifier can be improved by reducing the bound on the false negatives, which will decrease the recall rate. However, based on our task to predict cardiac disease, recall is more important than precision, so we choose the threshold accordingly to maximize the recall rate.

### Asymmetric Error Control by AEC

The ability to control the false negative below a certain threshold is the pivotal part of our AEC Tree Classifier, since it provides asymmetric error control on our predictions. figure 3.5 shows how the performance metric varies with the upper bound $\alpha$ on the false negatives for this dataset. The x-axis show the value of upper bound on false negatives, known as the threshold $\alpha$, varying from 0 to 1. The solid lines in the graph represent how each metric varies as $\alpha$ is increased. The dashed line in the figure shows the optimal value of the threshold.

The goal is to select a value of $\alpha$ that minimizes the false negatives and maximizes the accuracy and AUC-ROC score of the model. This graph allows us to choose $\alpha$ in a data adaptive way. The exact values used to plot figure 3.5 are listed in table 3.6. The detailed results are able to provide the exact values of the accuracy, AUC-ROC score, Precision, Recall and the number of false negatives produced at each threshold value. figure 3.5 shows that we are able to get an accuracy of 50 percent with false negatives close to 0 using our AEC Tree Classifier. Even though an accuracy of 50 percent is very low, it gives us the supreme advantage of keeping false negatives close to 0 which is not possible using traditional methods.

However, the optimal is achieved with upper bound on false negatives close to 0.22, since it still gives us an accuracy and AUC-ROC score over 80 percent. At this value, the AUC-ROC Scores and Accuracy still exceed 80 percent with recall around 90 percent. The false negatives are also just around 18 percent which may be a good compromise for the higher numbers in other metrics. The dashed line in figure 3.5 shows the optimal value of the threshold in this use case. This value could change if other metrics become more important, and this graph allows us to choose the value visually in a data adaptive way. None of these metrics go wildly out of control as the upper bound $\alpha$ is loosened, which indicates we can adjust the value according to our use cases.

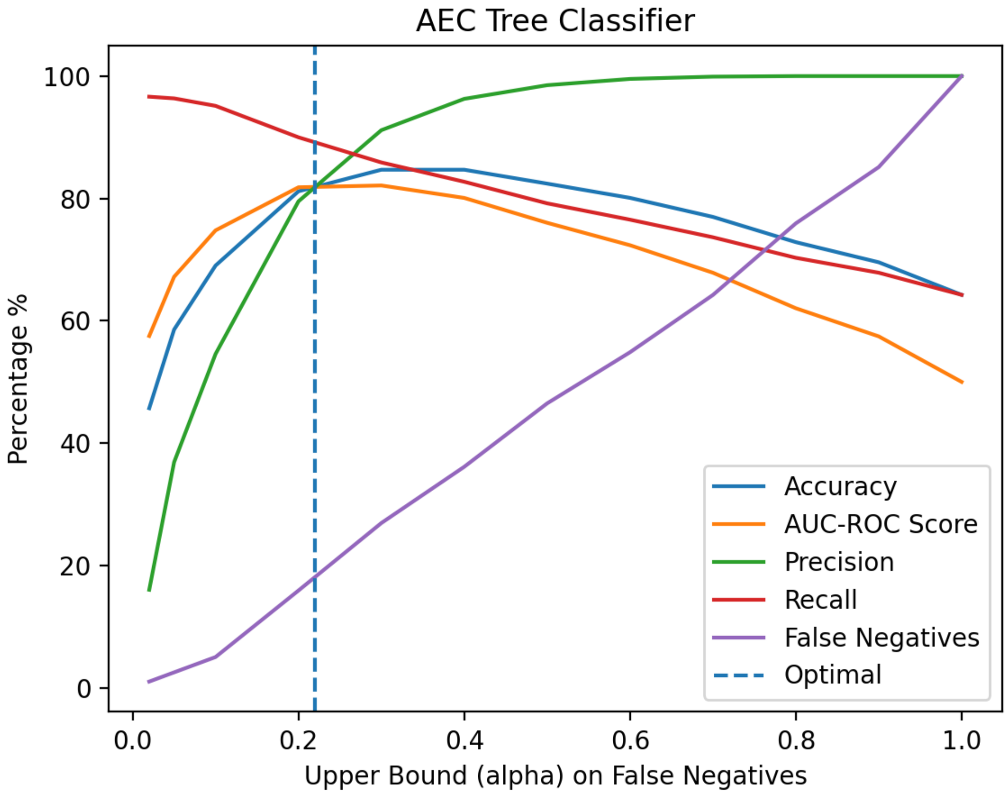


Figure 3.5: Dataset 1 - AEC Tree Classifier Control Over False Negatives

Table 3.6: Dataset 1 - AEC Tree Classifier Control Over False Negatives with Varying values of $\alpha$

| Alpha | Accuracy | AUC-ROC | Precision | Recall | False Negatives |
|---|---|---|---|---|---|
| 0.02 | 42.9 | 55.5 | 11.4 | 98.4 | 0.33 |
| 0.1 | 70.0 | 75.1 | 57.5 | 93.3 | 7.4 |
| 0.2 | 82.5 | 82.3 | 83.1 | 88.9 | 18.6 |
| 0.3 | 84.9 | 82.8 | 90.2 | 86.8 | 24.6 |
| 0.4 | 85.9 | 81.8 | 96.1 | 84.2 | 32.4 |
| 0.5 | 83.5 | 77.6 | 98.3 | 80.4 | 43.1 |
| 0.6 | 79.6 | 71.7 | 99.3 | 76.1 | 55.9 |
| 0.7 | 76.5 | 67.2 | 99.8 | 73.3 | 65.3 |
| 0.8 | 73.5 | 62.9 | 1.0 | 70.8 | 74.1 |
| 0.9 | 68.5 | 55.9 | 1.0 | 67.1 | 88.1 |
| 1.0 | 64.2 | 0.5 | 1.0 | 64.2 | 99.9 |

It should be noted that although we can increase our accuracy further and bring it close to 85 percent at around 0.4 value of $\alpha$ , this may not be a good option, since we want to keep false negatives low. In other use cases, the threshold value should be chosen carefully depending on the cost of a false negative. The results here also show that not all classifiers should be designed to be optimized for accuracy as other metrics, such as false negatives may be equally important in some cases. Figure 3.5 shows how asymmetric error control is achieved using our AEC Tree Classifier.

Table 3.7: Confusion Matrix of AEC Tree Classifier at threshold of 0.2

| | **Actual Positive** | **Actual Negative** |
|---|---|---|
| Predicted Positive | 502 | 235 |
| Predicted Negative | 96 | 839 |

Table 3.8: Confusion Matrix of AEC Tree Classifier at threshold of 0.4

| | **Actual Positive** | **Actual Negative** |
|---|---|---|
| Predicted Positive | 389 | 46 |
| Predicted Negative | 209 | 1028 |

Table 3.9: Confusion Matrix of AEC Tree Classifier at threshold of 0.6

| | **Actual Positive** | **Actual Negative** |
|---|---|---|
| Predicted Positive | 284 | 7 |
| Predicted Negative | 314 | 1067 |

Table 3.10: Confusion Matrix of AEC Tree Classifier at threshold of 0.8

| | **Actual Positive** | **Actual Negative** |
|---|---|---|
| Predicted Positive | 133 | 3 |
| Predicted Negative | 465 | 1071 |

The confusion matrix is another measure to see the actual number of true positives and true negatives compared to the number of false positives and false negatives. Tables 3.7 through 3.10 show how the confusion matrix numbers change with different threshold values (0.2, 0.4, 0.6 and 0.8) of the upper bound on false negatives. Our test data has 1672 records with 1074 as negative class and 598 positive records which have CVD. We can sum up the values in the confusion matrix to make sure we get the same numbers. The tables show that as the threshold is increased, the number of false negatives increase, whereas the number of false positives decrease. The results show that a high value of threshold may not be suitable due to the high number of false negatives. This data provides us more insight into choosing the threshold value that is best suited for the problem.

### *3.5.2 Dataset 2: Predict Cardiac Disease*

For the larger cardiac disease dataset, AEC Tree Classifier is within two percent of the best performing classifier in terms of both accuracy and AUC-ROC Score. This data has very balanced overall distribution of the two classes, which means that accuracy and AUC-ROC scores are very similar for all the classifiers. The complete results from this dataset can be seen from figure 3.6 with table 3.11 showing the exact values used to plot the graph.

The results show that our classifier has the best recall rate and the lowest number of false negative by a significant margin. The number of false negatives are less than 9 percent of the the false negatives by random forest. This means that it will be able to predict cardiac disease for additional 9 percent of the population, which can be potentially life saving. This benefit makes up for the loss in precision compared to other classifiers, given that the accuracy is still reasonably high. Moreover, in the cardiac disease datasets, the primary focus remains on the controlling the number of false negatives.

The results from figure 3.6 clearly show that our AEC Tree Classifier outperforms all other machine learning classifiers in the recall rate and false negatives. Despite the increased control on the number of false negatives, the AEC Tree Classifier is still comparable to the other classifiers in terms of accuracy and AUC-ROC scores. This implies that our AEC Tree Classifier is the best classifier overall, to predict cardiac disease.

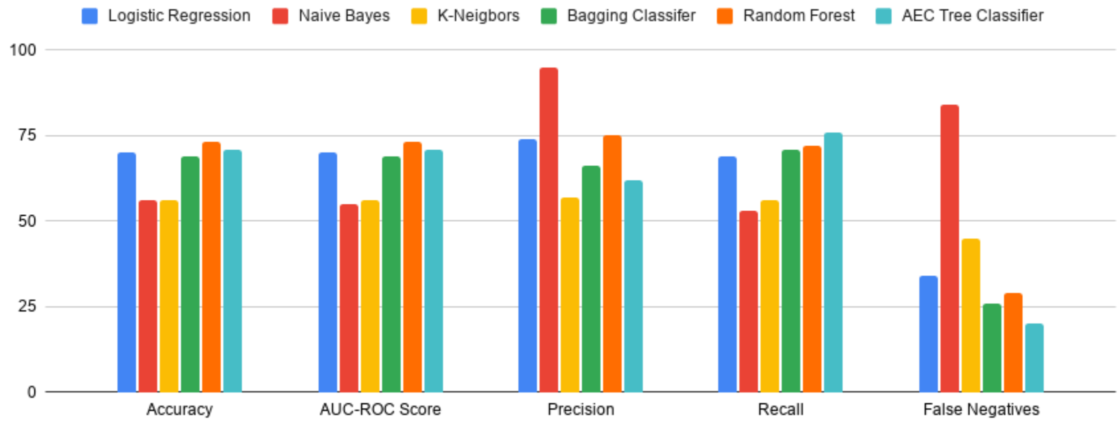


Figure 3.6: Results from Dataset 2 - AEC Tree Classifier versus ML Classifiers to predict cardiac disease

Table 3.11: Dataset 2 - AEC Tree Classifier vs ML Classifiers

| Classifiers | Accuracy | AUC-ROC | Precision | Recall | False Negatives |
|---|---|---|---|---|---|
| Logistic Regression | 70 | 70 | 74 | 69 | 34 |
| Naive Bayes | 56 | 55 | 95 | 53 | 84 |
| K-Neigbors | 56 | 56 | 57 | 56 | 45 |
| Bagging Classifer | 69 | 69 | 66 | 71 | 26 |
| Random Forest | 73 | 73 | 75 | 72 | 29 |
| AEC Tree Classifier | 71 | 71 | 62 | 76 | 20 |

Figure 3.7 shows how asymmetric error control is achieved on varying values of the threshold on the larger cardiac disease dataset. Table 3.12 shows the values used to

plot the graph in figure 3.7. The figure shows how the classifier is able to control the false negatives with the value of $\alpha$. The optimal value is represented by the dashed line at 0.27 on the x-axis. At this value, the recall rate is still over 75 percent while still having the accuracy and AUC-ROC scores in excess of 70 percent. This graph makes it easily to visualize the performance at different values of $\alpha$, making it easier to pick the threshold value that guarantees control over the number of false negatives.

The graph in figure 3.7 also shows the recall rate decreasing and the precision increasing as the upper bound on the number of false negatives is relaxed. F1 score, which is a measure of the balance of precision and recall, is not the best metric for asymmetric error control, as the goal is not to strike a balance between precision and recall, but to increase the recall rate while minimizing the loss to precision as much as possible. This is why accuracy and AUC ROC score are used along with precision and recall to evaluate the classifier.

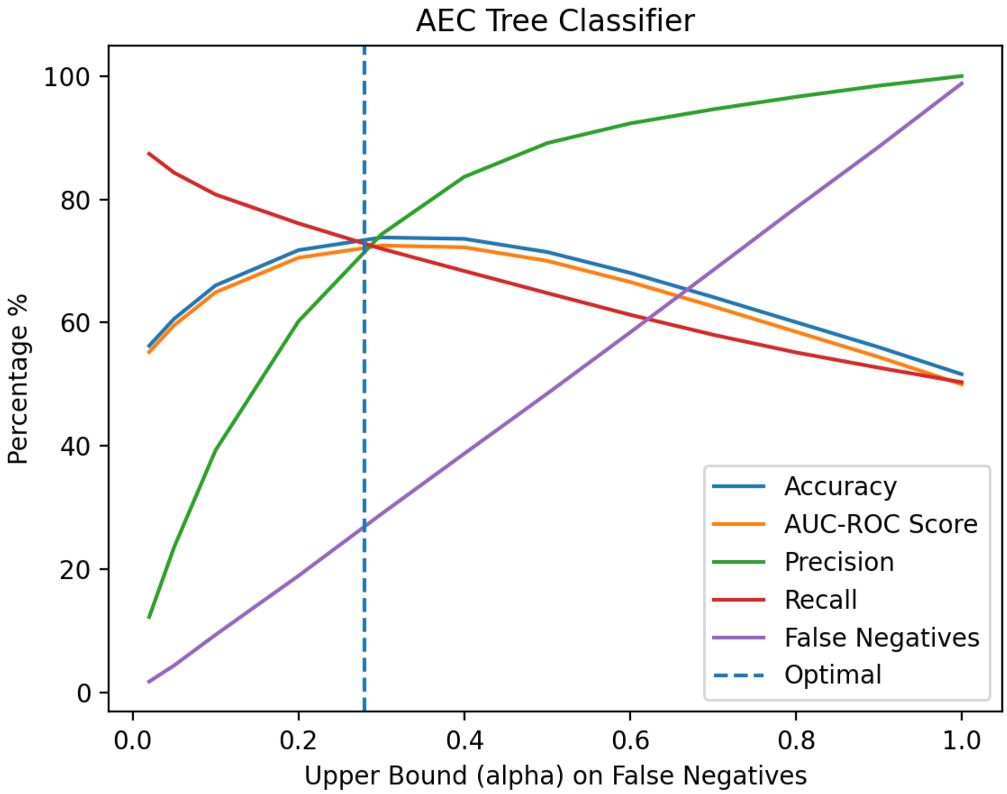


Figure 3.7: Dataset 2 - AEC Tree Classifier Control Over False Negatives

Table 3.12: Dataset 2 - AEC Tree Classifier Control Over False Negatives with Varying values of $\alpha$

| Alpha | Accuracy | AUC-ROC | Precision | Recall | False Negatives |
|---|---|---|---|---|---|
| 0.02 | 54.9 | 55.2 | 12.3 | 87.4 | 1.8 |
| 0.1 | 64.7 | 64.9 | 39.3 | 80.8 | 9.4 |
| 0.2 | 70.4 | 70.5 | 60.2 | 76.1 | 18.9 |
| 0.3 | 72.5 | 72.5 | 74.3 | 71.9 | 28.9 |
| 0.4 | 72.3 | 72.2 | 83.7 | 68.4 | 38.7 |
| 0.5 | 70.1 | 70.0 | 89.1 | 64.8 | 48.5 |
| 0.6 | 66.8 | 66.6 | 92.3 | 61.3 | 58.4 |
| 0.7 | 62.8 | 62.6 | 94.6 | 58.0 | 68.5 |
| 0.8 | 58.8 | 58.5 | 96.6 | 55.2 | 78.6 |
| 0.9 | 54.7 | 54.4 | 98.4 | 52.7 | 88.5 |
| 1.0 | 50.3 | 50.0 | 1.0 | 50.3 | 98.8 |

### *3.5.3 Dataset 3: Predict Rain Tomorrow*

For the Australian rain weather large scale dataset, the AEC Tree classifier is within three percent of the best performing algorithm in terms of accuracy. Moreover, in terms of AUC-ROC scores, the AEC classifier exceeds the chart with a score of 71 percent. The results, shown in figure 3.8 and table 3.13, also show that our classifier again produces, not only the best recall rate, but also keeps false negatives to the lowest by a big margin of 14 percent. This shows that we are not losing much on accuracy with an increased control over the number of false negatives. The precision score is the lowest due to the strong constraints on the recall rate. The results from these figures show that even for larger diverse datasets, the AEC Tree Classifier is able to provide asymmetric error control and produce comparable results.

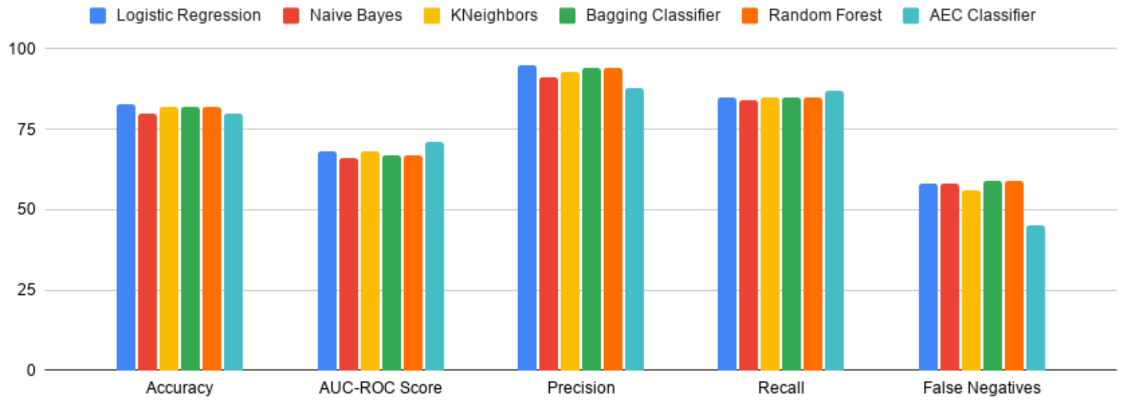


Figure 3.8: Results from Dataset 3 - AEC Tree Classifier versus ML Classifiers to predict rain tomorrow

Table 3.13: Dataset 3 - AEC Tree Classifier vs ML Classifiers

| Classifiers | Accuracy | AUC-ROC | Precision | Recall | False Negatives |
|---|---|---|---|---|---|
| Logistic Regression | 83 | 68 | 95 | 85 | 58 |
| Naive Bayes | 80 | 66 | 91 | 84 | 58 |
| KNeighbors | 82 | 68 | 93 | 85 | 56 |
| Bagging Classifier | 82 | 67 | 94 | 85 | 59 |
| Random Forest | 82 | 67 | 94 | 85 | 59 |
| AEC Classifier | 80 | 71 | 88 | 87 | 45 |

Figure 3.9 demonstrates the ability of asymmetric error control by our classifier for a large-scale non-medical disease dataset. Table 3.14 shows the values used to plot figure 3.9. The control over this dataset proves the ability of the classifier to work

across diverse datasets. However, it must be noted that asymmetric error control may not be as useful in predicting rain, and higher accuracy might be preferred. However, using our AEC Tree Classifier, we have the option to control the number of false negatives if needed.

Figure 3.9 shows the dashed line that represents the optimal value. This value is chosen as 0.47 $\alpha$, where the accuracy is close to 80 percent and AUC-ROC score over 70 percent. These values are still comparable to other classifiers without asymmetric error control on this dataset. The graph also shows the recall rate decreasing as the value of the upper bound is increased. The graphs show how the recall rate and false negative are controlled using the AEC Tree classifier, allowing asymmetric error control for diverse binary classification problems.

## 3.6 Discussion and Contributions

This section discusses how this chapter addresses some of the research objectives proposed in section 1.2. The research presented in this chapter helps address two of the four research objectives justified by the results and methodology from this chapter. This discussion also shows how some of the challenges posed by the research objectives have been addressed by this research.

The first research objective (RO1) from section 1.2 that aims, “To achieve asymmetric error control in binary classification with high probability that the population false negatives will not exceed a predefined threshold.” is met based on the results section from this chapter. The results section clearly shows that we are able to provide asymmetric control for binary classification on our AEC Tree Classifier, which is built based on the NP Lemma. This has been evaluated for three different datasets and the figures 3.5, 3.7 and 3.9 show the control we have over the number of false negatives for each classifier. The related works section from this chapter answers the

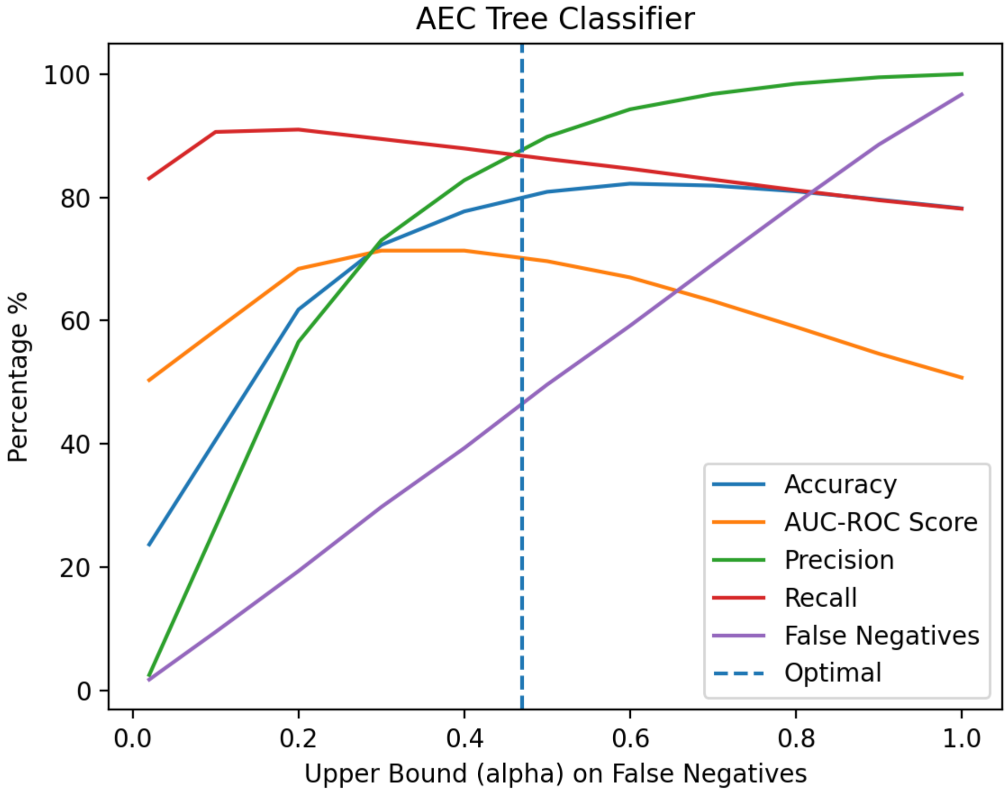


Figure 3.9: Dataset 3 - AEC Tree Classifier Control Over False Negatives

differences of this approach from cost based learning.

Another contribution from this dissertation is that it has established a data adaptive way to pick the upper bound threshold $\alpha$ value that gives us optimal result balancing the acceptable number of false negatives with the performance metrics. This threshold value can easily be adjusted according to our use case of asymmetric error control. This is shown from the dashed line in figures 3.5, 3.7 and 3.9. Despite providing control over the number of false negatives, our classifier is still able to produce AUC-ROC scores that are comparable with other classifiers. The loss in precision is acceptable in medical disease diagnosis.

The second research objective (RO2) in section 1.2 aims to choose a classification

Table 3.14: Dataset 3 - AEC Tree Classifier Control Over False Negatives with Varying values of $\alpha$

| Alpha | Accuracy | AUC-ROC | Precision | Recall | False Negatives |
|---|---|---|---|---|---|
| 0.02 | 23.4 | 50.2 | 2.5 | 81.8 | 1.9 |
| 0.1 | 40.3 | 58.3 | 26.3 | 90.6 | 9.4 |
| 0.2 | 60.8 | 67.9 | 55.2 | 91.1 | 18.7 |
| 0.3 | 71.7 | 71.3 | 72.0 | 89.7 | 28.6 |
| 0.4 | 78.3 | 71.8 | 83.4 | 88.2 | 38.8 |
| 0.5 | 81.3 | 70.2 | 89.9 | 86.6 | 48.2 |
| 0.6 | 82.5 | 67.6 | 94.2 | 85.0 | 57.5 |
| 0.7 | 82.3 | 63.7 | 96.9 | 83.2 | 67.5 |
| 0.8 | 81.5 | 59.7 | 98.5 | 81.6 | 76.9 |
| 0.9 | 79.9 | 54.8 | 99.6 | 79.8 | 87.6 |
| 1.0 | 78.4 | 50.8 | 99.9 | 78.3 | 95.8 |

threshold, independently from the training of the classifier,that controls the population type 2 error. One of the contributions of this dissertation is that it proposes a way to select a threshold such that the population type 2 error is controlled with high probability. From the training data of 0 and 1 samples, where 1 represents the critical class whose error we need to control, we split into mixed classes of 0 and 1 samples and some left out class 1 samples. The scoring function is trained on the mixed classes of 0 and 1 samples and then it is applied to the left out class 1 samples. From the classification scores obtained from the left out class 1 samples, we pick a threshold value using order statistics such that the type 2 error is under alpha with high probability. This approach ensures that the threshold search is not biased on the training data because the threshold selection is based on left-out class 1 data, which was not used for training. This allows us to interpret the training of the scoring function and the threshold selection as two independent procedures. The methodology section describes this approach in detail with equations based on mathematical lemma to back this claim.

## 3.7 Summary

To summarize, the two main problems in binary classification for medical diagnosis are imbalanced datasets and the uneven distribution of the cost of the two errors. The AEC Tree Classifier is able to solve the former using Hellinger Distance as the tree splitting criterion and the latter by using a Neyman Pearson Lemma based mathematical approach to provide asymmetric error control. This classifier is tested on diverse datasets to predict the 10-year risk of CVD, the immediate absence or presence of cardiac disease, and to predict if it will rain tomorrow.

The results show that the AEC Tree Classifier is able to control the number of false negatives and provide asymmetric error control across different datasets. Although other tree-based classifiers, such as random forest and bagging tree classifiers, are able to compete with our AEC Tree Classifier in terms of accuracy and AUC-ROC score, yet our classifier emerges as the clear winner in predicting cardiac disease by having the lowest number of false negatives. Furthermore, it also has the ability to easily reduce the false negatives further at a cost of reduced accuracy, which is often acceptable in medical disease diagnosis.

To conclude, the AEC Tree Classifier provides full control over the number of false negatives in binary classification problems, and is able to predict cardiac disease with full asymmetric error control. The methods used to create this classifier can be easily expanded to work with any binary classification problem, where there is a need for asymmetric error control. This approach can have a massive impact in the medical domain, especially in disease diagnosis, where we typically need to control the number of false negatives. Moreover, the results also show that asymmetric error control can be achieved outside the field of medical disease diagnosis in large diverse datasets.

Chapter 4

# MULTI-CLASS CLASSIFICATION WITH ASYMMETRIC ERROR CONTROL

## 4.1 Introduction

Classification [67], in machine learning, requires the use of different algorithms to assign class labels to examples from the dataset. It is a part of supervised learning, which aims to automatically predict unique outcomes for new observations after training on the relevant labeled data. Some examples of classification include medical disease diagnosis, recognizing spam email and image classification. The two main types of classification include binary classification and multiclass classification. The former predicts one of two classes whereas the latter involves prediction of more than two classes. This dissertation aims to improve the classification outcomes for multi-class classification in medical disease diagnosis.

### *4.1.1 Challenges in Classification for Medical Disease Diagnosis*

There are two main challenges in multiclass classification for medical disease diagnosis. Firstly, the cost of a false negative greatly outweighs the cost of a false positive in medical disease diagnosis. For example, predicting a patient infected with Covid-19 as not positive (false negative) can cause it to spread uncontrollably, whereas misclassifying a patient not infected with the novel Covid-19 strains as positive (false positive) has less serious repercussions, even though not ideal. Similarly, classifying a patient with cardiac arrhythmia as healthy could mean no treatment for the patient and could cost a patient their life. Conversely, misdiagnosing a healthy patient as having cardiac arrhythmia has less severe consequences.

The imbalance in the cost of a false negative compared to a cost of false positive in medical disease diagnosis highlights the need for asymmetric error control in classification. Traditional classification models may not be ideal in this scenario as they do not provide a mechanism to control the number of false negatives below a certain threshold with a mathematical guarantee. Even if these models result in improved accuracy and reduced classification error, they may not be optimal for medical disease diagnosis, since the cost of one error greatly outweighs the other.

Most classification models optimize for accuracy without providing a mathematical guaranteed control over the number of false negatives. Our published work [69] has highlighted ways of providing asymmetric error control for binary classification using the Neyman Pearson (NP) Lemma. The classification model, proposed in this dissertation extends that approach, providing asymmetric error control for multiclass classification, and thus enabling control over the false negatives with a mathematical guarantee that the false negatives will not exceed a certain user specified threshold.

The second main challenge for classification in medical disease diagnosis is the data imbalance. Multiclass classification problems with imbalanced dataset present different challenges compared to a binary classification problem. The skewed distribution of classes in multiclass classification, makes conventional machine learning models less reliable in predicting, especially in minority class examples. The imbalance in medical datasets arises because a positive disease diagnosis is typically a rare event compared to a negative diagnosis, leading to the negative class outweighing the positive class.

This problem is usually addressed using Synthetic Minority Over-sampling Technique (SMOTE) [92] which can balance the dataset. However, this technique is not ideal as extensive use of sampling can distort the results in a couple of ways. Firstly, oversampling the minority class can lead to overfitting as well as increasing the learning time of an algorithm. Secondly, under sampling the majority class can lead to removal of some important data points. To avoid the overuse of SMOTE, this dissertation proposes the use of Hellinger distance [70] as the splitting criteria, instead of Gini index and entropy. This splitting criterion in our decision tree addresses the imbalance in the dataset by quantifying the difference between two probability distributions and eliminates the need of extensive sampling techniques.

### *4.1.2 Multi-Class Classification in Cardiac Arrhythmia*

Cardiac arrhythmia [68] is a condition in which a person's heartbeat is irregular and may beat too quickly, too slowly or just with an irregular rhythm. Even though some minor heart arrhythmia's could be harmless, other irregular heartbeats can result from a weak or damaged heart and can lead to morbidity and mortality. In other cases, there can be serious complications leading to fatal symptoms. Arrhythmia is very difficult to diagnose because it might not cause any noticeable symptoms. This problem is compounded by the fact that not only do some people with life threatening arrhythmia may have no symptoms, but also some people with symptoms may not have severe arrhythmia.

Since some types of cardiac arrhythmia are life threatening, their prediction, detection and classification for both diagnosis and treatment, are important issues in clinical cardiology. Timely treatment is still essential for preventing further complications, which may include stroke and heart failure. This dissertation proposes the use of machine learning to determine the type of arrhythmia from the electrocardiogram (ECG) recordings. The aim is to predict the absence or presence of cardiac arrhythmia and to classify it in one of the 16 groups. Table 4.1 shows the different types of class codes available in our dataset. Class 01 refers to 'normal" whereas classes 02 to 15 refer to different types of arrhythmia, and class 16 belongs to unclassified ones.

Table 4.1: Cardiac Arrhythmia Classes

| **Code** | **Class** |
|---|---|
| 01 | Normal |
| 02 | Ischemic changes(Coronary Artery Disease) |
| 03 | Old Anterior Myocardial Infarction |
| 04 | Old Inferior Myocardial Infarction |
| 05 | Sinus tachycardy |
| 06 | Sinus bradycardy |
| 07 | Ventricular Premature Contraction (PVC) |
| 08 | Supraventricular Premature Contraction |
| 09 | Left bundle branch block |
| 10 | Right bundle branch block |
| 11 | 1st. degree AtrioVentricular block |
| 12 | 2nd. degree AtrioVentricular block |
| 13 | 3rd. degree AtrioVentricular block |
| 14 | Left ventricule hypertrophy |
| 15 | Atrial Fibrillation or Flutter |
| 16 | Others |

After consulting with four cardiologists as part of this research, the five most critical or life threatening types of cardiac arrhythmia were identified. They were then ranked according to their order of severity as shown in figure 4.1. From this list of hierarchical order ($1 \geq \ldots \geq K$ ) of severity for misclassification, as listed in figure 4.1, the multiclass classification algorithm was applied such that we have control over the number of false negatives for the most critical class.

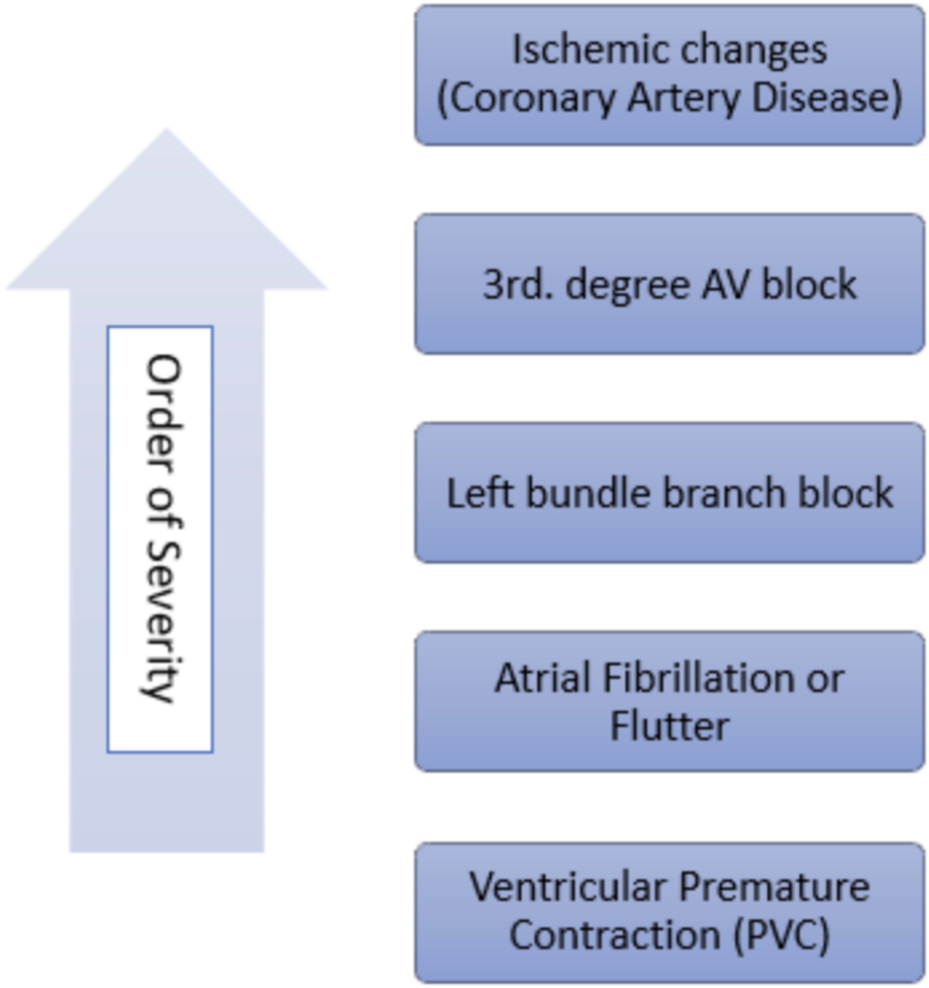


Figure 4.1: Accuracy of ML Classifiers vs CVD Tree Classifier

The next section discusses the related work in this domain. Next, the dissertation proposes the methodology and the theoretical foundation behind the multiclass Neyman Pearson Tree Classifier that is created as part of this research. The results section starts by explaining the experimental setup and hyper parameters used to compare the classifier. Finally, the results from the evaluation are presented and compared with the existing models to conclude that the tree-based classifier is able to provide asymmetric error control in multiclass classification.

## 4.2 Related Works

This section discuss discusses the related work and the existing methods being used to predict different types of cardiac arrhythmia. In this section, the dissertation also covers the current ways of asymmetric error control for classification followed by a summary of other techniques, which enable binary classification techniques to be applied to multi-class problems in classification. Finally, this section discuses machine learning in cardiac arrhythmia domain.

### *4.2.1 Asymmetric Error Control in Classification*

The two main ways of asymmetric error control in classification algorithms are cost sensitive learning and ROC Curve. For cost sensitive learning, the model gets trained on the dataset, while calculating the cost of prediction errors. This approach can have three shortcomings for error control. Firstly, there is no consensus way to assign costs of each error, leading to inconsistent and variability in results. Secondly, the cost sensitive learning does not provide a mathematical guarantee that it will be able to control the number of false negatives below a certain threshold. Lastly, there is an ethical dilemma, in assigning costs to human lives. For example, it can be considered morally unethical to assign more value to one life over another. Similarly, in disease diagnosis, the patient's life is of paramount importance and we never want to cruelly set a value for it, knowing the algorithm could potentially sacrifice lives to maximize it's interest [79].

ROC curve [78] is another way for asymmetric error control in classification. The ROC space is defined as a two dimensional space, where each axis represents the false positive and false negative rates respectively [84]. The scoring function of a binary classification model can be estimated using the ROC curve, and this can be expanded to control one type of error. However, extending this to multiclass has a huge challenge in terms of computational complexity, that increases exponentially to the number of classes, resulting in many problems being intractable [72].

The current state of art classification algorithms are optimized to minimize the classification error instead of providing asymmetric control over one type of error. This can be shown from the classification error equation, which has been mentioned in chapter 2 as well but it is also included here:

$$\text{Classification Error} = FP + FN * 100/N \tag{4.1}$$

where,

FP is the number of false positives (type 1 error)

FN is the number of false negatives (type II error)

N is the total number of samples

This equation clearly shows that the false negatives and false positives are grouped together in the classification error equations. The goal of classification algorithms is to minimize the classification error, which implies that it will be optimized to reduce both kinds of errors. The only way to control one type of error in this scenario is using threshold moving. However, that is complicated for multiclass classification as the classes can start overlapping, as the threshold is moved. Even for binary classification, this approach is only valid for empirical data and there is no probabilistic guarantee that for a random sample from the population, the false negatives will be controlled using this approach.

### *4.2.2 Multi-Class Classification to Binary Classification*

Error-Correcting Output Codes [74] is a technique that allows a multiclass classification problem to be reconstructed as multiple binary classification problems, allowing the use of native binary classification techniques on multiclass datasets. The error-correcting output codes technique allows each class to be encoded as an arbitrary number of binary problems in classification. It allows the extra models to act as error correction predictions when over determined representation is used [75]. This allows error control that can result in better predictive performance, however this approach does not scale as the number of classes increases, as there is no mathematical upper bound on the number of false negatives for any specific class.

One-Vs-Rest (OVR) and One-Vs-One (OVO) for Multi-Class Classification [73] are other similar solutions that decompose a multiclass classification problem into binary classification problems. They allow native binary classification models to be used in multiclass datasets by converting them into multiple binary problems. However, even with these models, there exists no mathematical guarantee to control the false negatives of a particular class.

### *4.2.3 Cardiac Arrhythmia and Machine Learning*

In recent years, there has been an increased interest in the use of machine learning and artificial methods in the medical domain, in the hope to discover new predictive and diagnostic tools [71]. Recent research has also focused on the detection and classification of arrhythmia using machine learning. There have been algorithms implemented to predict the onset of cardiac arrhythmia for patients with Implantable Cardioverter-Defibrillators (ICD), which is used to treat patients with cardiac arrhythmia's. However, the study noticed that the prediction of arrhythmia still remains challenging.

The results obtained from the Computational Science – ICCS 2020 study show that RR intervals carry the necessary information about the onset of arrhythmia [71]. Even though the study had some limitations in terms of data size and lack of clinical input, it showed potential to use machine learning in arrhythmia outcomes. However, the ability to predict cardiac arrhythmia with asymmetric error control still lacks in modern research. This dissertation aims to solve this problem following the methodology described in the next section.

## 4.3 Methodology

The Neyman-Pearson (NP) [87] classification paradigm aims to enable asymmetric error control in machine learning for binary classification outcomes. Np classification is based on hypothesis testing and that requires null hypothesis and alternative hypothesis, which is naturally suited to binary classification. Clearly, NP oracle inequalities are not immediately applicable for multiclass NP classification. This research proposes a new variant of the algorithm called Multi Class Asymmetric Error Control Classifier (MCAEC) that is able to provide asymmetric error control for multiclass classification.

### *4.3.1 NP Classification*

The NP classification is based on the NP Lemma, which states that the likelihood test is the most powerful hypothesis testing among all other tests [88]. Given that X is the data or the observation, $H_0$ is null hypothesis and $H_1$ is alternative hypothesis, then P(X, $H_0$) is probability distribution of X given $H_0$ and P(X, $H_1$) is a probability distribution of X given $H_1$.

This leads us to the likelihood ratio test L(X) which states:

$$L(X) = \frac{P(X_1, H_1)}{P(X_1, H_0)} > \gamma$$

where $\gamma$ (gamma) is threshold gamma. To maximize $\mathrm{P_D}$ for given $\mathrm{P_{FA}} = \alpha$ where $\mathrm{P_D}$ is probability of detection and $\mathrm{P_{FA}}$ is the probability of false alarm. The given alpha $\alpha$ is the upper bound threshold value. $\mathrm{P_{FA}}$ is then calculated as:

$$P_{FA} = P_r L(X) > \gamma, H_0 = \alpha$$

which is the probability that the likelihood ratio is greater than gamma under the null hypothesis $\mathrm{H_0}$.

This dissertation extends the NP umbrella algorithm to the multiclass classification problem. A simple use case for multiclass classification outside cardiac arrhythmia is in cancer diagnosis. For example, if we have three classes: class A (cancer of the most dangerous kind), class B (benign cancer) and class C (no cancer) and we would first like to control the error of misclassifying class A and then control the cost of misclassifying class B. In this scenario, we will adapt our algorithm based on the OVR approach so that NP properties still hold as shown in the following figures. Figure 4.2 shows how the dataset will appear with three cancer classes A,B and C. Figure 4.3 shows the resulting dataset after B and C are clumped together using the OVR approach.

### *4.3.2 MCAEC Algorithm*

The same approach can be extended to more classes including the 16 classes of cardiac arrhythmia dataset by applying the above technique recursively and incorporating it with the NP oracle inequalities. The essential idea is to maintain a hierarchical order ($1 \geq \ldots \geq$ K ) of severity for misclassification. First the NP methods are applied to 1 versus ($2 \geq \ldots \geq$ K ). If a new observation is assigned to class 1,

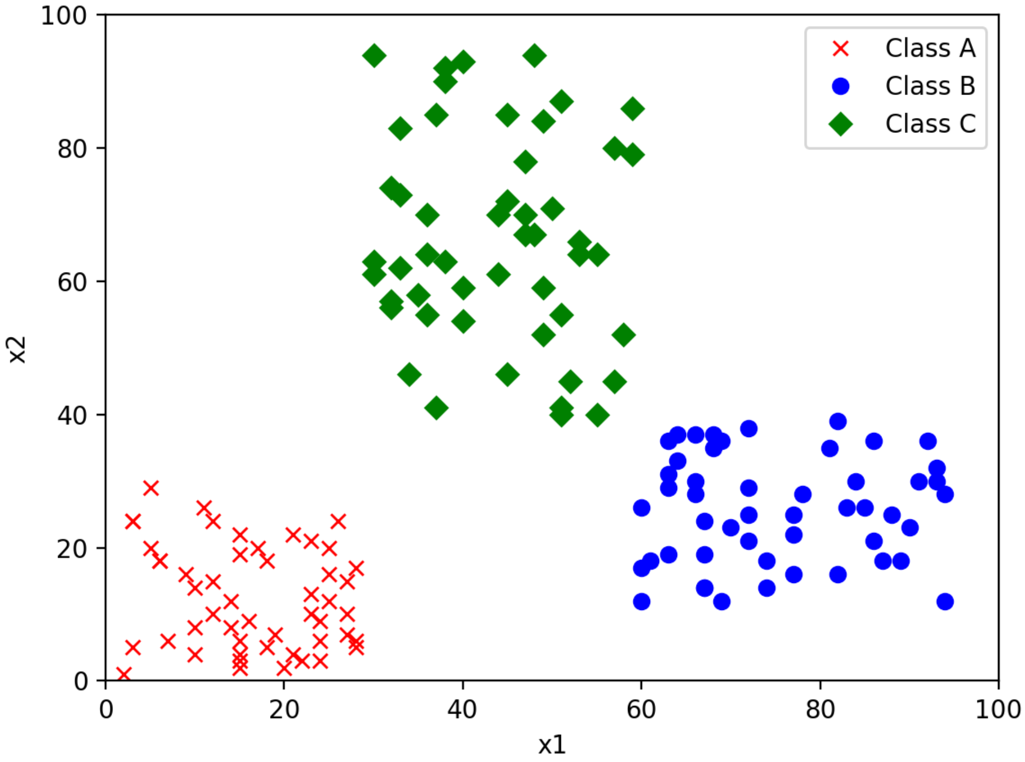


Figure 4.2: Multiclass Data Original

the algorithm is stopped and assigns the class label to class 1. Otherwise, apply NP methods to 2 versus $(3 \geq \ldots \geq K)$ and so on, until the new observation is assigned a label. The pseudo code of this algorithm is explained in Algorithm 1.

For traditional classifiers, the threshold is chosen as 0.5 by default for binary classifiers and only the scoring function needs to be calculated from the training data. Similarly for multiclass classification, the threshold is not calculated as part of training the algorithm. However, for the MCAEC classifier, both the scoring function and the threshold is calculated from the training data. This concept is summarized in figure 4.4 and figure 4.5. Figure 4.4 shows that training data is fed as input only for the scoring function and the default values of the threshold is being used. Meanwhile,

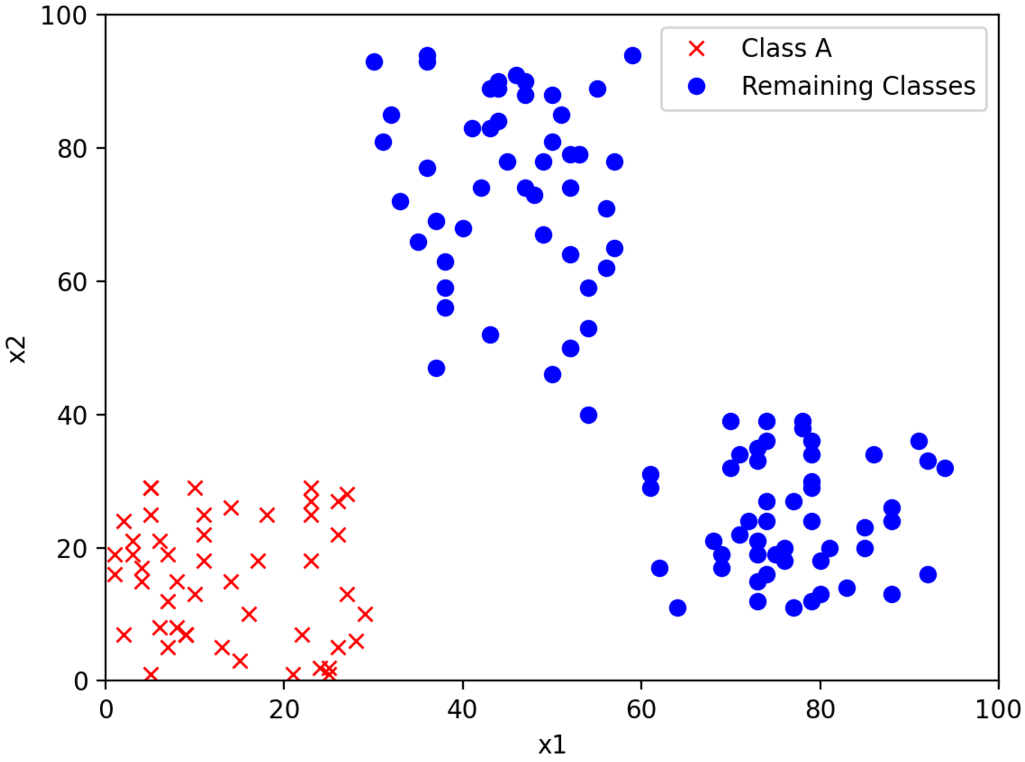


Figure 4.3: Multiclass Data after OVR

figure 4.5 shows that the threshold is being trained as well. The training data is used to calculate the threshold, such that the false negatives are bound by a user specified value.

### *4.3.3 Hellinger Distance as Splitting Criterion*

The other significant challenge in multiclass classification in medical disease diagnosis, apart from asymmetric error control, is the imbalanced datasets. As mentioned earlier, the imbalance occurs because a positive disease diagnosis is typically a rare event compared to a negative diagnosis, leading to the negative class outweighing

**Algorithm 1** MCAEC ALGORITHM

```
Input: data $x_i$, array of order of severity (1 $\geq$ ... $\geq$ k )
Initialize $classIndex = 0$.
Initialize $label = None$.
repeat
    Initialize $criticalClass = array[classIndex]$.
    $classIndex + +$
    while $label$ != $criticalClass$ do
        $label = NP(array, classIndex, criticalClass)$
        if $label$ == $criticalClass$ then
            $break$
        else
            $classIndex + +$
        end if
    end while
until return $label$
```

the positive class. Decision trees are generally considered to be the best with an imbalanced small scale dataset due to their ensemble capabilities [90]. The MCAEC classifier is based on ensemble decision tree classifiers [26]. It includes a large number of individual decision trees that operate together as an ensemble. Each individual tree predicts a class label and the predicted class is based on majority voting, similar to random forest [91] and extra tree classifiers [26].

A key differentiation point from MCAEC tree classifier versus the popular ones is the choice of the splitting criterion [94]. The splitting criteria of the decision tree is one of the most critical points in decision tree induction algorithms. This dissertation proposes the use of Hellinger distance as the splitting criterion over more populous

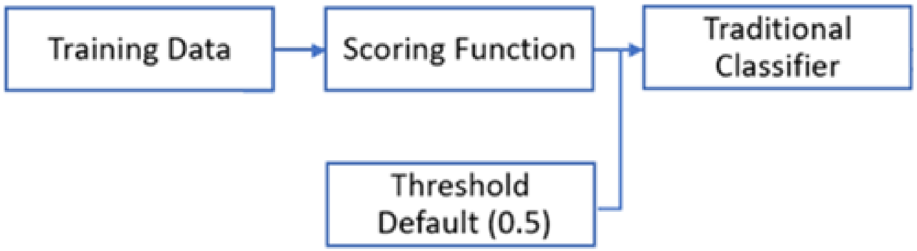


Figure 4.4: Traditional Classifier

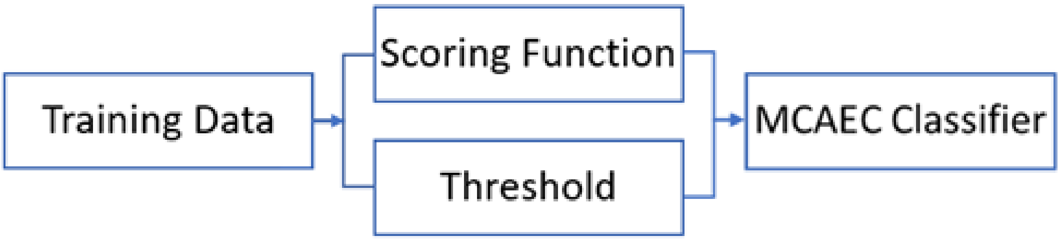


Figure 4.5: MCAEC Classifier

ones such as Gini Index and Entropy. The reason is that hellinger distance addresses the imbalance between the different classes in a dataset without the need of extensive sampling techniques. This eliminates the need of extensive sampling techniques, which have their own downsides. For example, under sampling can remove some important data, whereas oversampling could lead to overfitting.

The Gini impurity [77] which is used in traditional tree based classification algorithms is calculated as:

$$GiniImpurity = \sum_{i=1}^{k} p_{\mathrm{i}}(1-p_{\mathrm{i}}) = -\sum_{i=1}^{k} p_{\mathrm{i}} \log(p_{\mathrm{i}})$$

where,

k is the number of classes

$\mathrm{p_i}$ is the proportion of cases belonging to case i.

The equation shows that Gini Impurity is not optimized for unbalanced datasets as it is dependent on k and p. Hellinger distance, which is used as the splitting criterion, is calculated as:

$$h(P,Q) = \frac{1}{\sqrt{2}} \cdot \|\sqrt{P} - \sqrt{Q}\|_2 \tag{4.2}$$

where h(P,Q) is the Hellinger distance between 2 probability distributions P and Q. By its very nature, hellinger distance is able to address the imbalance between two classes by considering them as two different probability distributions.

Figure 4.6 summarizes the steps taken to split the training data into a form where the MCAEC classifier can provide asymmetric control. The first step in the process is to identify the most critical classes from our multiclass dataset and record the hierarchical order of severity of misclassification on these classes. Next, the most critical class, the first one from the ordered list is chosen as the class whose false negatives will be controlled. Then using OVR, the training data is divided into 0 and 1 classes as shown in figure 4.6. From this data sample, it is further split into three parts. The first part consists of mixed samples of 0 and 1 classes, the second part has the left out class 0 samples and the third part consists of remaining class 1 samples, as visible in figure 4.6.

The mixed classes of 0 and 1 samples are fed as input to the MCAEC classifier

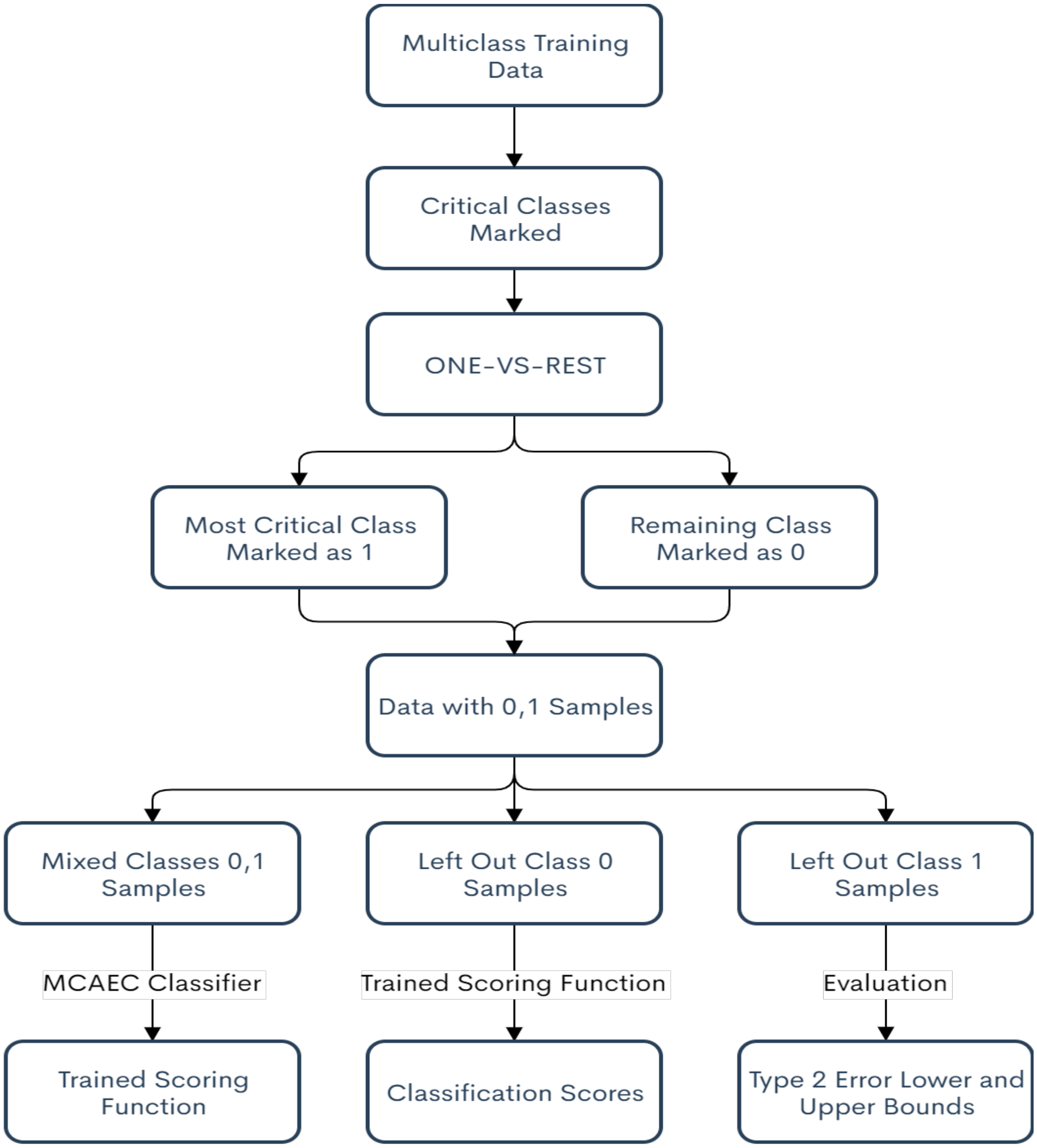


Figure 4.6: Splitting on Multi Class Training Data

to create the scoring function. This scoring function is used on the remaining class 0 samples to come up with a list of classification scores. Order statistics is used to find a threshold from this list, such that the violation rate, which is the probability of type 2 error exceeding the user specified upper bound alpha, is minimized. Order statistics was proven to be the best in finding the threshold from the list of classifications compared to brute force and bagging [88]. The probability that the false negative error exceeds alpha is controlled by a user specified tolerance parameter $\delta$.

$$\mathbb{P}\left[R_0\left(\varphi^c\right) > \alpha\right] \leq \delta \tag{4.3}$$

This equation shows that the probability that the violation rate exceeds alpha is bound by a user specified parameter. The threshold chosen from the classification scores is taken from the left out class 0 samples, which were not used to train the classifier. The remaining class 1 samples are used to minimize the false positive error rate as much as possible, with the false negative error fixed. The above steps can then be repeated for the next class in the order of severity of misclassification to incorporate the error control for that class. However, there is a constraint that error control can only be applied to one class in a single run. These steps are incorporated with the decision tree hellinger distance classifier to create the MCAEC classifier that is able to achieve asymmetric error control for multiclass classification.

## 4.4 Results

### *4.4.1 Results on Cardiac Arrhythmia Dataset*

The dataset to test the MCAEC classifier was taken from the proceedings of the cardiology conference in Sweden published by H. Altay Guvenir [76]. This dataset consists of 279 attributes of over 450 instances. Even though this is not a massive dataset, it is one of the largest ones in the cardiac arrhythmia space, as it is difficult to obtain useful data from cardiac arrhythmia patients. The 16 different classes from this dataset are mentioned in table 4.1. Class code 02, which is Ischemic changes (Coronary Artery Disease), is identified as the most critical class from this list followed by 3rd. degree AV block as shown in figure 4.1 earlier. This section will show asymmetric error control on class code 02. The same technique can be applied to provide error control on any specific class from a multiclass dataset.

The data was split into a 70-30 training-test split with the random state set to 10 to produce repeatable results. The missing values were replaced with the average of the columns. The number of trees in the classifier, which is known as n estimators, was set to 300. The minimum samples required to split was set to 2, and the maximum depth was left as the default. A hellinger distance probability distribution was implemented as the splitting criterion for the decision tree instead of using one of the standard options to account for the data imbalance. All the remaining hyper-parameters were left as default. The experiment was run multiple times and the average of the results was taken for each value.

The MCAEC classifier was compared to other popular algorithms in terms of accuracy, auc roc scores, f1 Score, precision and recall as shown in table 4.2. The results are also presented in figure 4.7. The values of all these accuracy metrics was normalized so they all fall in the 0 to 1 range.

The results show that the MCAEC classifier performs the best in terms of recall rate. The recall rate measures the ratio of patients, with the most critical class of cardiac arrhythmia, that were correctly identified. In other words, the recall rate measures the ability to reduce the number of false negatives. This implies that the MCAEC classifier was able to correctly identify 98 percent of the patients with the most critical class of cardiac arrhythmia. No other classifier is able to produce such a high recall rate as visible from table 4.2 and figure 4.7.

In terms of accuracy and F1 score, the MCAEC classifier is outperformed by the other classifiers. However, that may be acceptable in medical disease diagnosis, where the onus is on controlling the false negatives while still providing relatively high accuracy. The MCAEC classifier is still able to produce accuracy of over 80 percent on this dataset. The AUC ROC score is another important metric in an unbalanced

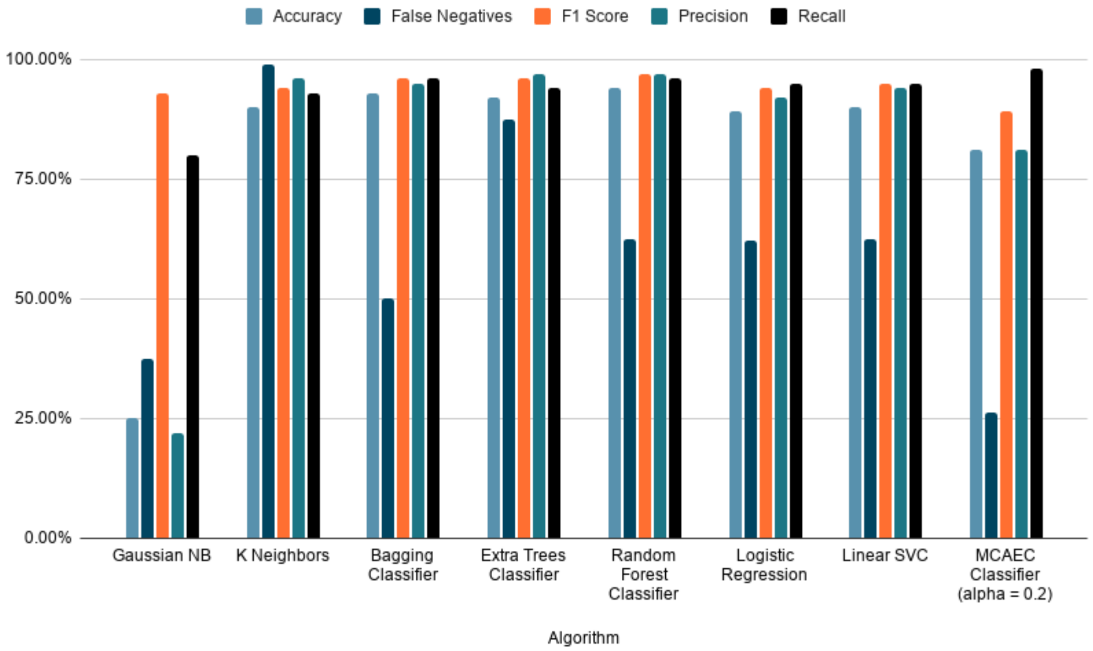


Figure 4.7: MCAEC Classifier vs Traditional Classifiers

dataset, as accuracy can be misleading, if the algorithm just predicts the majority class. The MCAEC classifier performs the best in this metric with a score of 0.78. All the results are plotted with the threshold value of 0.3, which means at most the false negatives will not exceed 30 percent for the specified class.

Figure 4.8 shows how the MCAEC classifier accuracy metrics are impacted with varying the threshold. The x-axis shows the value of $\alpha$, which is the user specified upper bound on the number of false negatives. The graph shows that as the value of the upper bound is increased, which means that the bound is loosened, the accuracy and f1-score improves. However, as the upper bound exceeds 0.5, the accuracy and f1-score start flattening out. The number of false negatives is inversely proportional to the recall rate and as the threshold increases, the false negatives increase causing the recall rate to decrease.

Table 4.2: MCAEC Classifier vs Traditional Classifiers

| **Classifier** | **Accuracy** | **False Negatives** | **F1 Score** | **Precision** | **Recall** |
|---|---|---|---|---|---|
| Gaussian NB | 25.00% | 37.50% | 93.00% | 22.00% | 80.00% |
| K Neighbors | 90.00% | 99.00% | 94.00% | 96.00% | 93.00% |
| Bagging Classifier | 93.00% | 50.00% | 96.00% | 95.00% | 96.00% |
| Extra Trees Classifier | 92.00% | 87.50% | 96.00% | 97.00% | 94.00% |
| Random Forest | 94.00% | 62.50% | 97.00% | 97.00% | 96.00% |
| Logistic Regression | 89.00% | 62.00% | 94.00% | 92.00% | 95.00% |
| Linear SVC | 90.00% | 62.50% | 95.00% | 94.00% | 95.00% |
| MCAEC Classifier (alpha=0.2) | 81.00% | 26.10% | 89.00% | 81.00% | 98.00% |

This graph allows us to choose the value of the threshold in a data adaptive way, such that the recall rate is high enough so that not only our false negatives are controlled, but also the accuracy metrics are still reasonably high. The dotted line marked as optimal in figure 4.8, shows the desired value of the threshold for the cardiac arrhythmia use case. At this value, the MCAEC classifier is able to get a recall rate of 98 percent with accuracy still exceeding 80 percent. Since, the value of the threshold is a user specified value to the MCAEC algorithm, we are able to control the number of false negatives of a specific class with a mathematical guarantee that it will not exceed this threshold value.

### *4.4.2 Results on Glass Dataset*

To further evaluate the MCAEC Classifier, it was tested on a different glass identification dataset [95]. This dataset contains 10 attributes and the classification goal

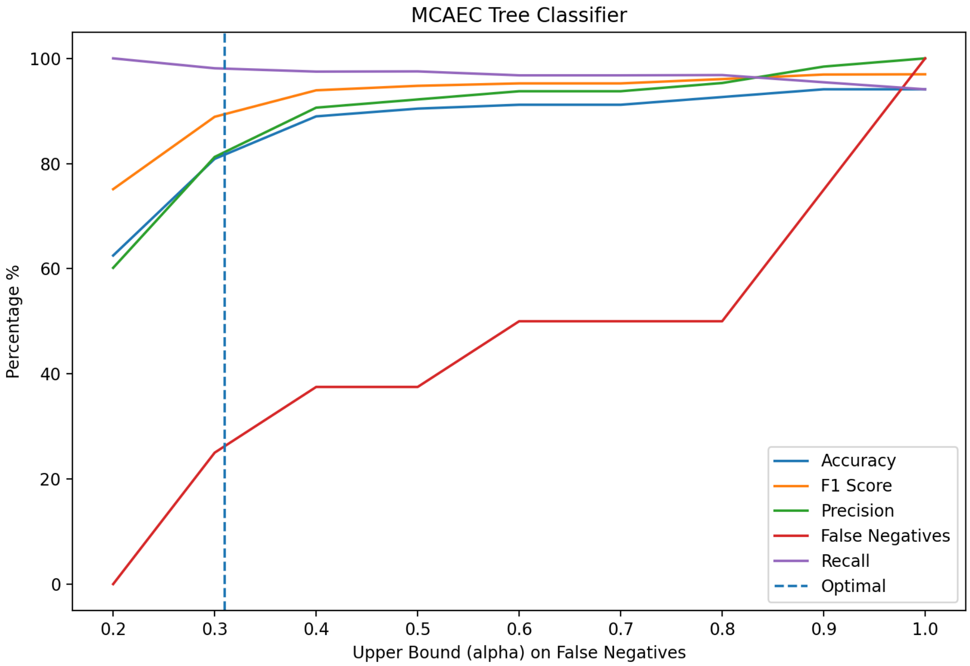


Figure 4.8: Asymmetric Error Control by MCAEC Classifier

is to predict the glass type. These attributes used to classify the glass type are shown in table 4.3. The multiple glass types are grouped in three classes (0,1 and 2) and the asymmetric error control is applied on class 0, as it was identified as the most critical glass type. Even though in the glass identification dataset, there is not a large need to control the number of false negatives, it still serves as a suitable multi-class classification problem to test the MCAEC Algorithm.

The graph from figure 4.9 shows how MCAEC Classifier is able to control the number of false negatives with varying values of alpha for the glass identification dataset. The graph shows that the number of false negatives increases and recall rate decreases as the threshold is increased. The graph also shows precision increasing with increased threshold, as the number of false positives reduce with a loose bound on the false negatives.

Table 4.3: Glass Identification Attributes

| **Attributes** | **Class** |
|---|---|
| 01 | RI: refractive index |
| 02 | NA: Sodium |
| 03 | Al: Aluminum |
| 04 | Si: Silicon |
| 05 | K: Potassium |
| 06 | Ca: Calcium |
| 07 | Ba: Barium |
| 08 | Fe: Iron |
| 09 | Mg: Magnesium |
| 10 | Type of glass |

The graph also shows how the accuracy and f1 score are affected with different values of threshold. The optimal value of the threshold is easy to select based on this figure as shown by the dotted line from this graph at 0.21 on the x-axis. At this value of the threshold, the false negatives are still under 3 percent with recall rate close to 98 percent. The accuracy and precision is still exceeding 60 percent, despite the strong bound on the number of false negatives. With F1 score close to 80 percent, this threshold value gives a strong control over the number of false negatives and still reasonably high numbers for the other metrics.

Figure 4.10 shows how the MCAEC Classifier performs on the glass dataset compared to other classifiers. The numbers used to plot figure 4.10 are shown in table 4.4. The results show that even though random forest and bagging classifier are able to achieve higher accuracy, the MCAEC Classifier is still within 5 to 6 percent range

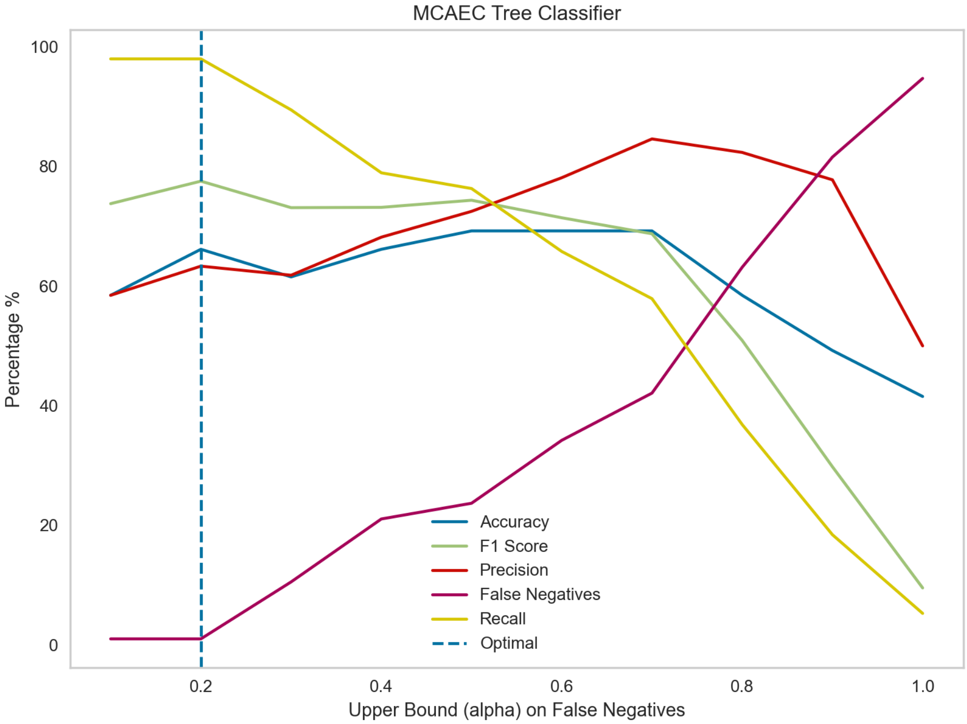


Figure 4.9: Asymmetric Error Control by MCAEC Classifier for glass dataset

in terms of accuracy. Moreover, the MCAEC Classifier greatly out performs all other classifiers in terms of the highest recall rate and lowest number of false negatives. It produces recall rate of 98 percent, which is more than 10 percent higher than the recall rate of random forest and bagging classifier.

If accuracy is the most important metric and the cost of false negatives and false positives are the same, then other classifiers may be able to out perform the MCAEC Classifier. However, these results show that MCAEC Classifier is the best option if there is a need to control the number of false negatives.

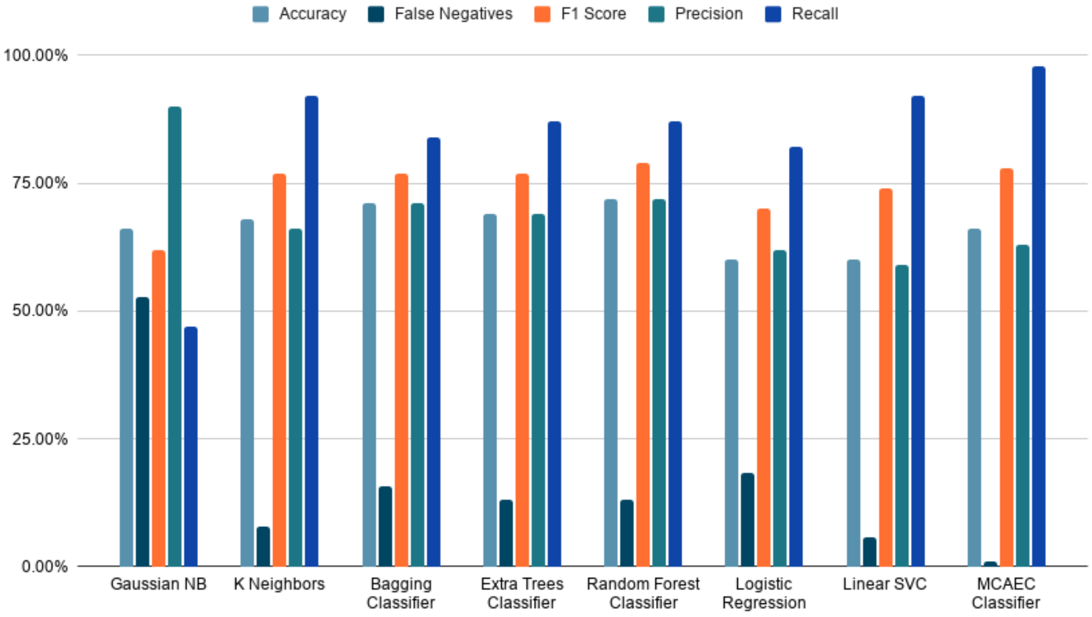


Figure 4.10: MCAEC Classifier vs Traditional Classifiers for glass dataset

Table 4.4: MCAEC Classifier vs Traditional Classifiers for glass datasets

| **Classifier** | **Accuracy** | **False Negatives** | **F1 Score** | **Precision** | **Recall** |
|---|---|---|---|---|---|
| Gaussian NB | 66.00% | 52.60% | 62.00% | 90.00% | 47.00% |
| K Neighbors | 68.00% | 7.89% | 77.00% | 66.00% | 92.00% |
| Bagging Classifier | 71.00% | 15.79% | 77.00% | 71.00% | 84.00% |
| Extra Trees Classifier | 69.00% | 13.15% | 77.00% | 69.00% | 87.00% |
| Random Forest | 72.00% | 13.16% | 79.00% | 72.00% | 87.00% |
| Logistic Regression | 60.00% | 18.42% | 70.00% | 62.00% | 82.00% |
| Linear SVC | 60.00% | 5.63% | 74.00% | 59.00% | 92.00% |
| MCAEC Classifier (alpha = 0.21) | 66.00% | 1.00% | 78.00% | 63.00% | 98.00% |

## 4.5 Summary and Contributions

This dissertation proposes a Multi Class Asymmetric Error Control Classifier (MCAEC) that is able to control the number of false negatives for the most critical classes in multiclass classification. This classifier is designed with focus on the two main challenges of multiclass classification in medical disease diagnosis, which are the high cost of a false negative and data imbalance. The MCAEC classifier is able to address both the challenges, the former using Np Lemma and the latter using Hellinger distance as the decision tree splitting criterion. The results show that the MCAEC classifier is not only able to control the recall rate under a user specified threshold, but also able to produce acceptable accuracy metrics.

These results also help us achieve one of the research objectives (RO3) from section 1.2 which aimed to extend asymmetric error control for binary classification to multiclass classification using the Neyman-Pearson (NP) Lemma in hypothesis testing. The objective also aimed to control the false negatives of the most critical class with high probability that the population false negatives will not exceed a predefined threshold and that it has been met based on the results.

As future research, multiclass classification with asymmetric error control should be explored with different techniques outside the NP lemma domain. This might allow a more natural extension to binary classification without the need to apply these techniques in a recursive manner.

Chapter 5

# HANDLING IMBALANCED DATA

Class Imbalance is a common problem in the medical domain as described earlier in this dissertation. This problem often arises because a positive disease diagnosis is typically considered a rare event, compared to a negative disease diagnosis. This leads to an increased number of the majority class compared to the minority class. This chapter explores this problem from the asymmetric control perspective. Specifically, it explores data imbalance techniques that can not only address the imbalance of the data, but also provide control over one type of error.

Moreover, this chapter explores the performance of these approaches on the Framingham Dataset that was introduced in chapter 2. This dataset serves as a good test to explore the cardiac disease prediction problem on imbalanced data that also requires asymmetric error control. The error control is needed because the cost of predicting a false negative for cardiac disease is much higher than the cost of predicting a false positive. Section 5.1 describes how the imbalance is handled by other approaches and how these approaches may also be able to achieve asymmetric error control. Section 5.2 discusses the performance and results of these approaches.

## 5.1 Imbalanced Dataset Approaches

In this section we will discuss multiple approaches to handle the data imbalance. The most common approach to handle the data imbalance problem is using sampling. A common technique for achieving this is using Synthetic Minority Oversampling Technique (SMOTE). This technique over-samples the minority class as part of its data augmentation process. Sampling has the advantage of being used with any classifier, however it does involve tampering with the actual data. This means that oversampling could lead to overfitting, whereas under-sampling could lead to loss of important data.

The use of ensemble classification methods is another natural way to handle the data imbalance. Recent works [112; 114] have shown that ensemble techniques are able to handle the data imbalance better than sampling in most cases. The results from the comparison performed by Florida Atlantic University [111] also indicate that ensemble learning methods are more efficient in classification of imbalanced data for bioinformatics data [113] as compared to sampling techniques. The AEC Tree classifier that is proposed in this dissertation is also using ensemble tree-based classification based on the Hellinger distance splitting criterion to address the data imbalance. The use of this technique allows a reduced need of sampling techniques for this approach. Moreover, the threshold chosen using this approach is able to achieve asymmetrical error control using AEC Tree classifier as shown in chapter 3 of this dissertation.

Cost sensitive learning is the most appropriate comparison to the AEC Classifier that was described in chapter 3. The reason for this is because cost sensitive learning not only provides asymmetric error control by adjusting the weights of different costs, but also provides a way to handle the imbalance as well. In many ways, the two approaches -handling data imbalance and achieving asymmetric error control- are analogous for cost sensitive learning. This is explained in figure 5.1, which shows how under-sampling is similar to down-weighting and oversampling is similar to oversampling. The figure shows how the two classes, 0 and 1 have unequal costs, and are imbalanced. The figure also shows how the two different approaches, cost sensitive learning and sampling, are applied to address the data imbalance problem. The added advantage of cost sensitive learning compared to sampling is that it is able to provide asymmetric error control by adjusting the different cost of errors. However, this approach has the disadvantage that it is not able to control the number of false negatives under a user specified percentage.

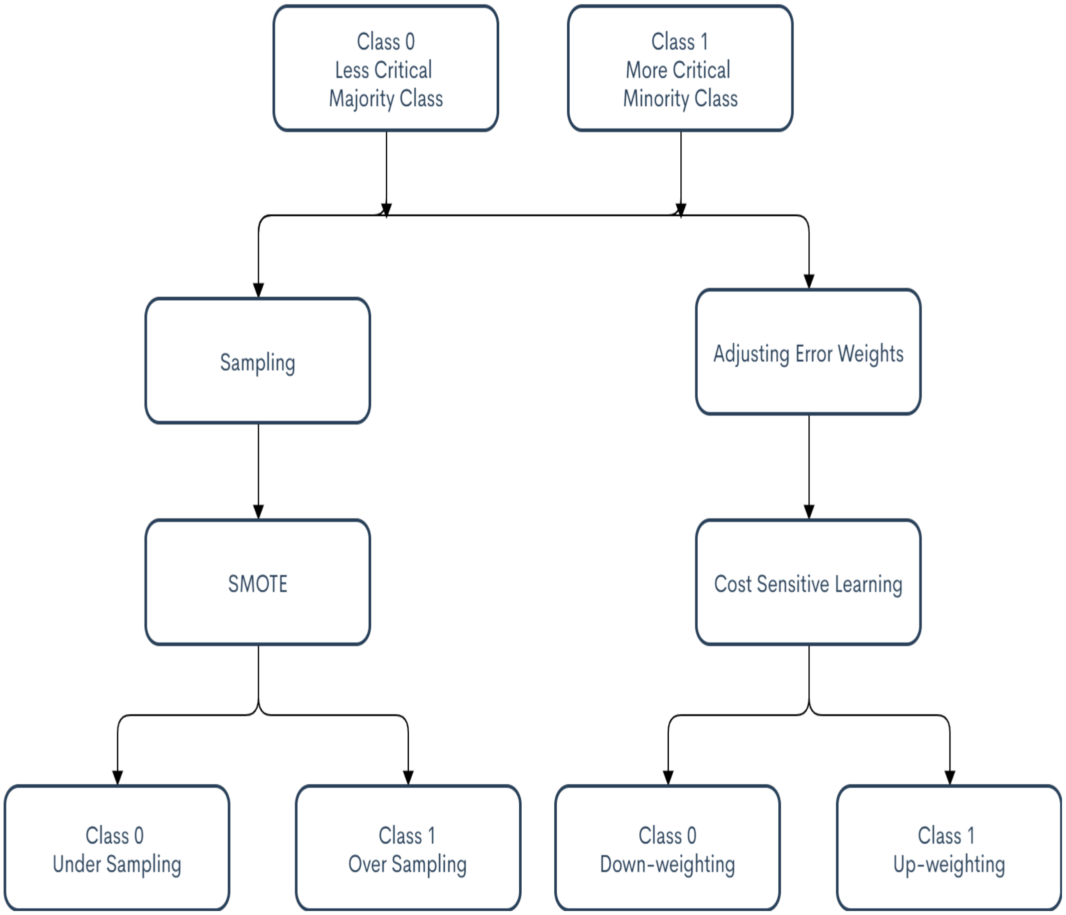


Figure 5.1: Cost-Sensitive Learning and Sampling

Moreover, there is no confirmation that the error control achieved by cost sensitive learning on the test data will also hold for the population data. The AEC Tree classifier provides this assurance with high probability that even if population data is widely different, the error control will still hold but the number of false positives may increase a lot. However, on population data that is taken from the same population, the AEC Tree Classifier is backed by the NP Lemma proof that shows how the bound on false negatives will always hold. The next section compares the performance of the different approaches discussed in this section on the Framingham Heart Study dataset.

## 5.2 Performance Results

This section shows the evaluation and results obtained from the comparison of the above mentioned approaches on imbalanced data. The top performing classifiers on this dataset were identified in the previous chapters, and their evaluation on the imbalanced data can be divided into four main areas. Firstly, the classifiers were compared on pure data with no measures taken to address the data imbalance. Secondly, the classifiers' results were obtained with the use of sampling techniques. Thirdly, cost sensitive learning was applied by adjusting the class weights of the two classes to handle the data imbalance and the results obtained. Lastly, a hybrid of cost sensitive learning along with sampling was applied to gauge its performance.

### *5.2.1 Hyper Parameters for Data Imbalance*

SMOTE was used for the sampling with a configuration of 0.55 to increase the minority class sample. Different values for SMOTE were tried. However, the value of 0.55 was identified where the imbalance was decreased but still prevents overfitting as all the imbalance was not eliminated. The results shown for sampling are performed at 0.55.

For cost sensitive learning, the class weight of 0.7 was given to the minority class and 0.3 class weight was given to the majority class. This value was identified as a balance where the accuracy was reasonably high and the number of false negatives were reduced.

For the AEC Tree Classifier, the value of alpha, which is a user-specified input to the function that controls the number of false negatives, was chosen as 0.22. The reasoning for selecting the optimal value of alpha is explained in chapter 3. Basically, at this value the number of false negatives are still controlled and the accuracy remains relatively high. The results shown in this section are plotted at this value of alpha.

### *5.2.2 Results on Framingham Dataset*

Tables 5.1 to 5.5 along with Figures 5.2 to 5.6 show the comparison of different classifiers with different ways of data imbalance. The metrics used to compare in these figures and tables include accuracy, AUC ROC scores, precision, recall and the number of false negatives. The AEC Tree classifier uses the NP Lemma based approach to control the false negatives. Therefore, the cost sensitive learning does not apply to it (indicated by N/A in the table).

The accuracy comparison on pure data is not the best metric, since there is imbalance on the dataset. This means that a naive algorithm that just predicts the majority class for all samples can get an accuracy up to 0.85. After sampling, the accuracy results from table 5.1 and figure 5.2 show that the ensemble tree classifiers outperform the others. We notice that using cost sensitive learning, along with sampling increases the accuracy of the ensemble tree classifiers slightly. However, this causes a reduction in accuracy for SVC and logistic regression.

The AUC ROC is a more appropriate measure for an imbalance dataset. The results from table 5.2 and figure 5.3 show how the AEC Tree classifier is able to achieve the best score on pure data. The reason for this is that it is using Hellinger distance as the tree splitting criterion, which can perform better than the splitting criterion's chosen for random forest and extra trees classifiers. After sampling, we notice the improvement in AUC ROC scores for all the classifiers, with the ensemble tree classifiers performing the best. Using cost sensitive learning on its own is not

Table 5.1: Accuracy

| Classifier | On Pure Data | Using. Sampling | Using Cost Sensitive Learning | Using Cost Sensitive with Sampling |
|---|---|---|---|---|
| Random Forest | 0.850 | 0.877 | 0.845 | 0.860 |
| Extra Trees | 0.845 | 0.880 | 0.847 | 0.880 |
| SVC | 0.845 | 0.782 | 0.845 | 0.69 |
| Logistic Regression | 0.851 | 0.693 | 0.823 | 0.649 |
| AEC Tree Classifier | 0.76 | 0.82 | N/A | N/A |

able to improve this score without the help of sampling.

Table 5.2: AUC ROC Score

| Classifier | On Pure Data | Using. Sampling | Using Cost Sensitive Learning | Using Cost Sensitive with Sampling |
|---|---|---|---|---|
| Random Forest | 0.532 | 0.844 | 0.519 | 0.822 |
| Extra Trees | 0.531 | 0.850 | 0.526 | 0.848 |
| SVC | 0.5 | 0.695 | 0.5 | 0.5 |
| Logistic Regression | 0.530 | 0.625 | 0.595 | 0.684 |
| AEC Tree Classifier | 0.61 | 0.83 | N/A | N/A |

The precision results from table 5.3 and figure 5.4 show that the other techniques are able to outperform the AEC Tree Classifier, which is expected since the AEC Tree Classifier is built to maximize the recall. However, even with this constraint, this classifier is still able to obtain a decent precision score of 0.84 without the need

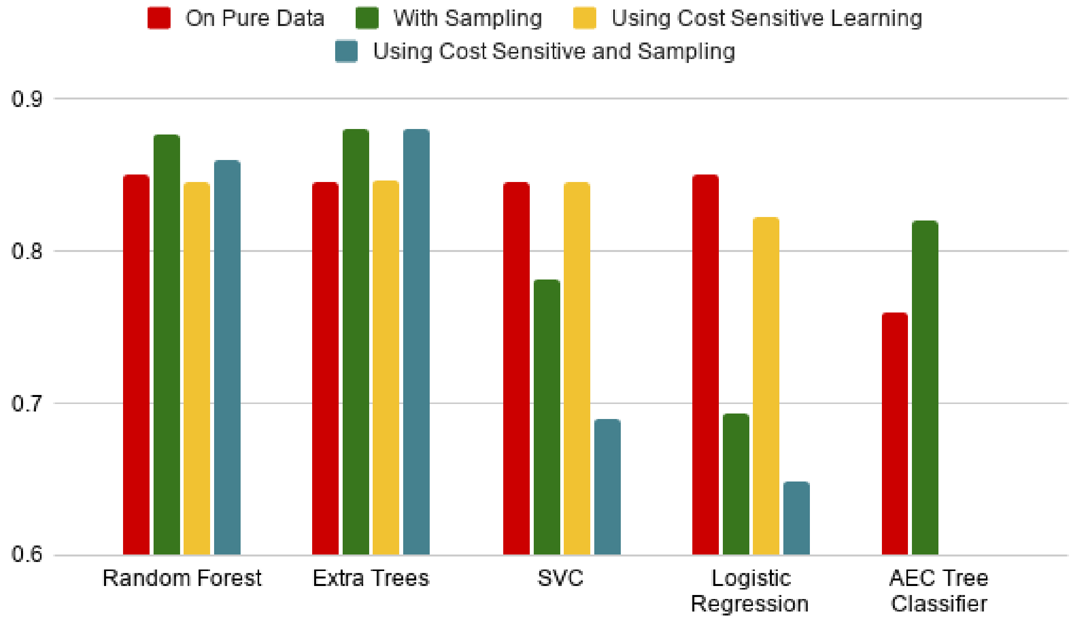


Figure 5.2: Accuracy

of sampling techniques. From figure 5.4, it seems that the precision is impacted significantly with the control on the false negatives. However, the Accuracy (Table 5.1) and AUC score (Table 5.2) are not reduced significantly. Moreover, this dataset used to plot figure 5.4 and figure 5.5 is very imbalanced so precision and recall may not be the best metrics for comparison without the use of sampling. If we look at the AUC ROC scores (figure 5.3), we can see that cost sensitive learning with sampling improves it significantly.

Table 5.4 and figure 5.5 show the recall results. The AEC Tree classifier is able to outperform all data imbalance handling techniques with a recall score of 0.89 without sampling, and 0.915 with the use of sampling. It is able to obtain a superior recall rate by around 5 percent from the next best classifier. Even with the use of cost sensitive learning and assigning higher class weight to the minority class, it does not lead to a recall rate that can beat the AEC Tree Classifier. The recall rate is one of the most

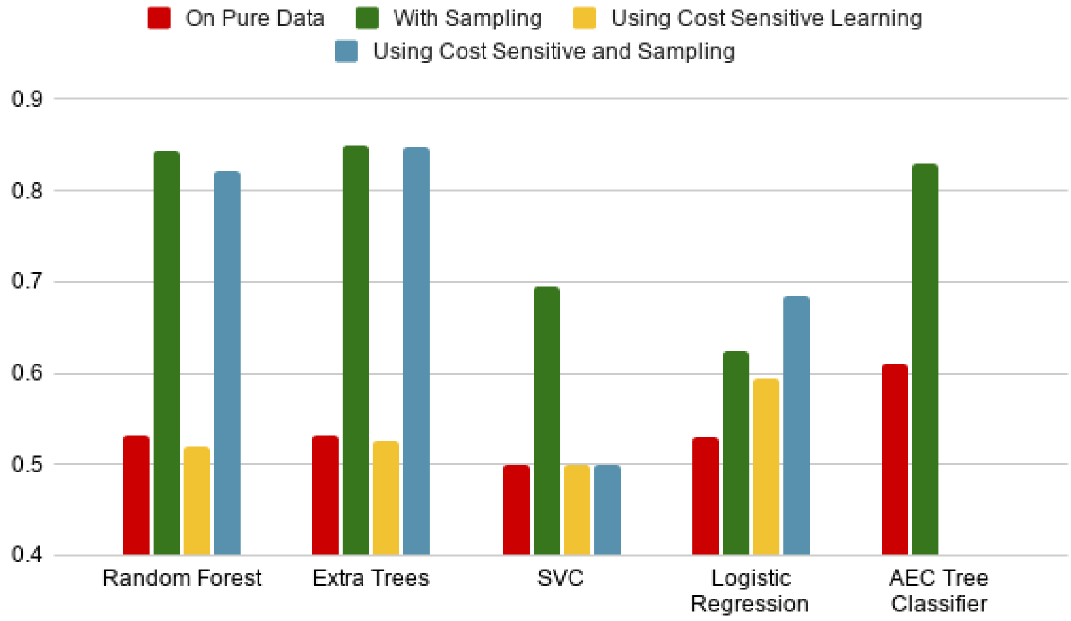


Figure 5.3: AUC ROC Score

important metrics, since it measures the amount of positive patients that the classifier is able to predict. A naive algorithm that predicts all samples as positive will have a recall rate of 1, but its accuracy will be around 0.15. The AEC Tree classifier is able to achieve a higher recall rate without compromising much on the accuracy.

The actual number of false negatives for each classifier are presented in table 5.5 and figure 5.6. The results show that AEC Tree Classifier outperforms the other methods by having the least amount of false negatives. Each false negative on this dataset implies that a patient with high risk of cardiac disease is predicted as low risk, which implies that the patient may not seek treatment and could potentially cost a life. This shows how critical it is to reduce the number of false negatives in this problem. On data without sampling, the AEC Tree classifier has almost 50 less false negatives than the other classifiers. Moreover, the AEC Classifier produces only around half the number of false negatives, even when compared to sampling and cost

Table 5.3: Precision

| Classifier | On Pure Data | Using. Sampling | Using Cost Sensitive Learning | Using Cost Sensitive with Sampling |
|---|---|---|---|---|
| Random Forest | 0.993 | 0.960 | 0.992 | 0.956 |
| Extra Trees | 0.986 | 0.958 | 0.991 | 0.963 |
| SVC | 1.0 | 1.0 | 1.0 | 1.0 |
| Logistic Regression | 0.995 | 0.862 | 0.926 | 0.562 |
| AEC Tree Classifier | 0.84 | 0.789 | N/A | N/A |

Table 5.4: Recall

| Classifier | On Pure Data | Using. Sampling | Using Cost Sensitive Learning | Using Cost Sensitive with Sampling |
|---|---|---|---|---|
| Random Forest | 0.853 | 0.863 | 0.850 | 0.846 |
| Extra Trees | 0.853 | 0.869 | 0.852 | 0.866 |
| SVC | 0.845 | 0.746 | 0.845 | 0.67 |
| Logistic Regression | 0.853 | 0.717 | 0.872 | 0.838 |
| AEC Tree Classifier | 0.89 | 0.915 | N/A | N/A |

sensitive techniques on the imbalanced data.

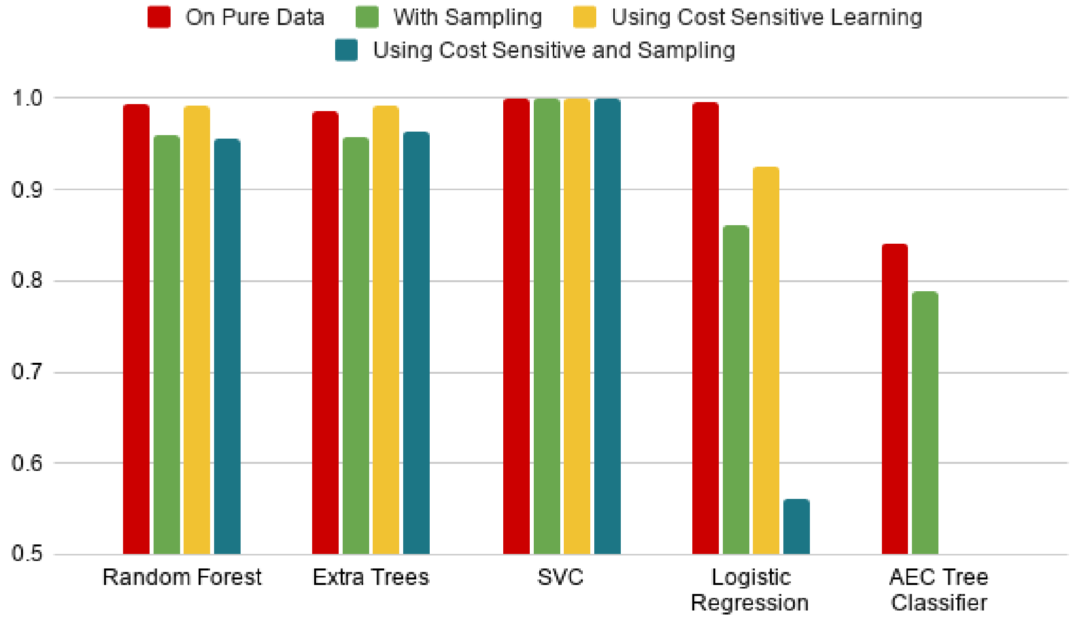


Figure 5.4: Precision

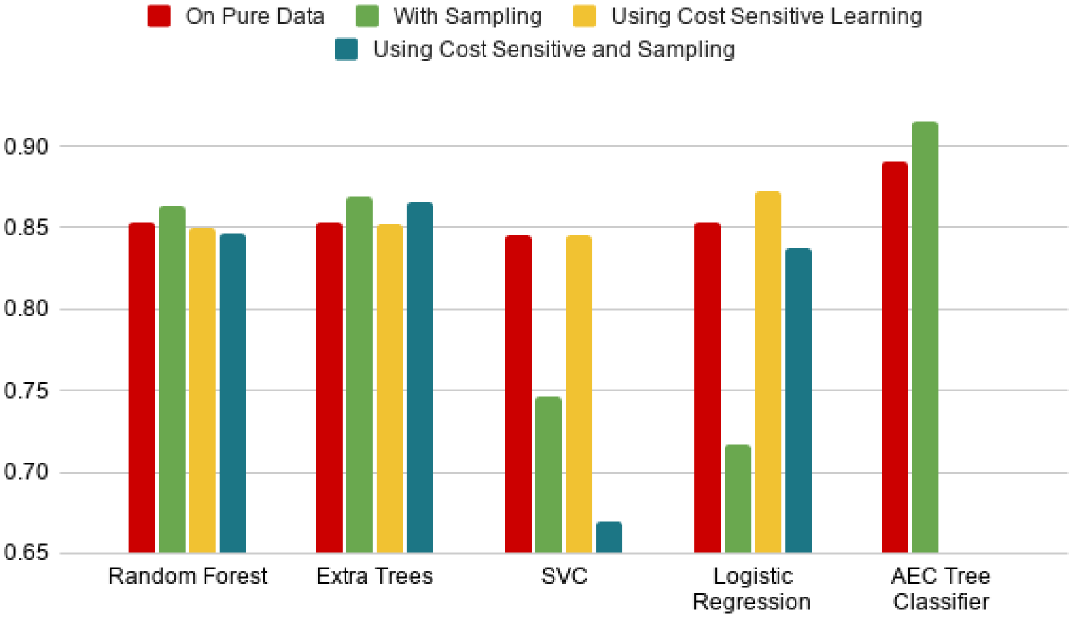


Figure 5.5: Recall

Table 5.5: Number of False Negatives

| Classifier | On Pure Data | Using. Sampling | Using Cost Sensitive Learning | Using Cost Sensitive with Sampling |
|---|---|---|---|---|
| Random Forest | 183 | 163 | 188 | 186 |
| Extra Trees | 182 | 154 | 185 | 160 |
| SVC | 197 | 364 | 197 | 515 |
| Logistic Regression | 184 | 365 | 145 | 116 |
| AEC Tree Classifier | 128 | 86 | N/A | N/A |

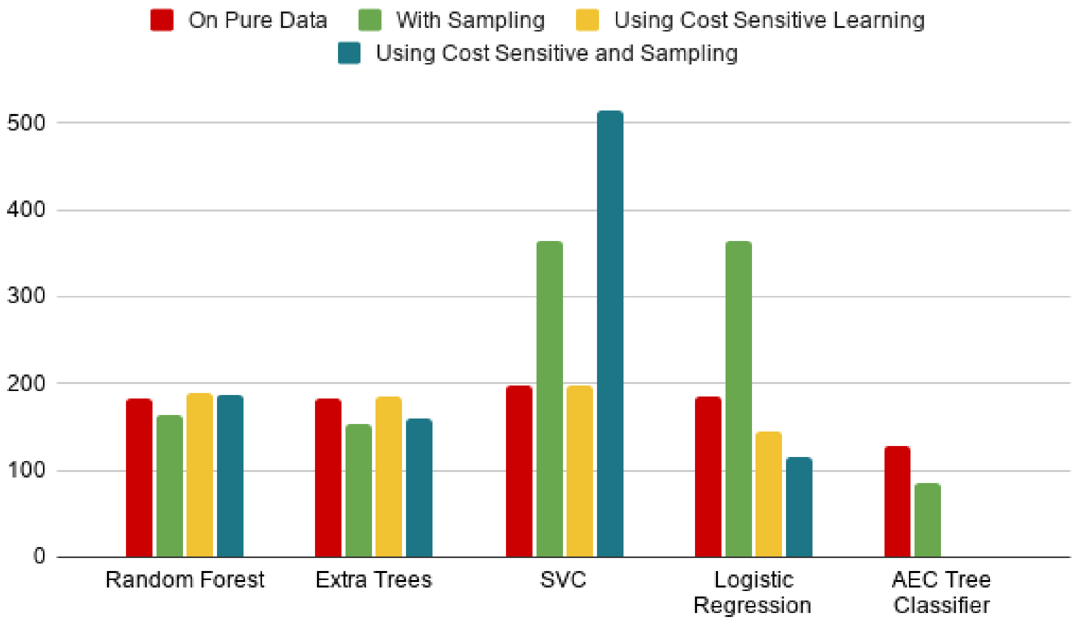


Figure 5.6: Number of False Negatives

The consolidated results from these figures and tables are summarized in table 5.6 and figure 5.7. These results show the comparison of AEC Classifier with these classifiers using cost sensitive learning.

Table 5.6: Dataset 1 - AEC Tree Classifier vs Cost-Sensitive Learning Classifiers

| Classifiers | Accuracy | AUC-ROC | Precision | Recall | False Negatives |
|---|---|---|---|---|---|
| Random Forest | 84.5 | 82.2 | 95.6 | 84.6 | 31 |
| Extra Trees | 84.7 | 84.8 | 96.3 | 86.6 | 26.7 |
| SVC | 84.5 | 50.1 | 99.9 | 67 | 86 |
| Logistic Regression | 82.3 | 68.4 | 56.2 | 83.8 | 19.4 |
| AEC Classifier | 82 | 83 | 78.9 | 91.5 | 14.4 |

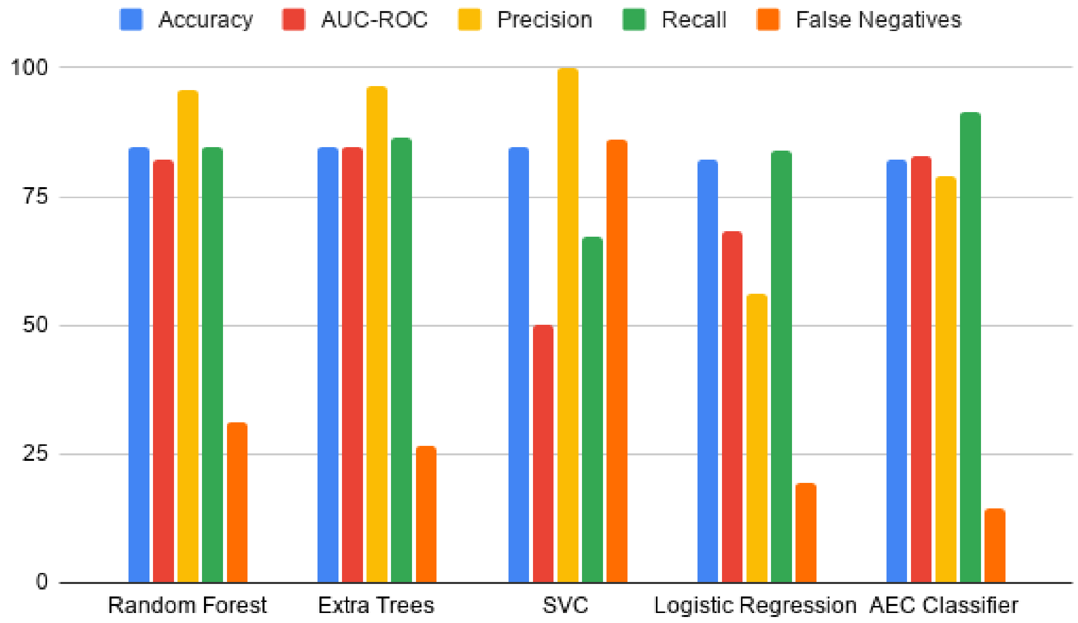


Figure 5.7: Results from Dataset 1 - AEC Tree Classifier versus Cost Sensitive Learning Classifiers to Predict the risk of 10-Year Cardiac Disease

### *5.2.3 Results on Larger Cardiac Disease Dataset*

The experiments were also performed on the larger cardiac disease dataset that was introduced in chapter 2. Table 3.11 in chapter 3 showed the comparison between AEC Classifier and other classifiers with the use of sampling. Table 5.7 shows the comparison of AEC Classifier with these classifiers using cost sensitive learning. All the values from this table are listed in percentages. The results from this table are also plotted in figure 5.8.

Table 5.7: Dataset 2 - AEC Tree Classifier vs Cost-Sensitive Learning Classifiers

| Classifiers | Accuracy | AUC-ROC | Precision | Recall | False Negatives |
|---|---|---|---|---|---|
| Random Forest | 72.8 | 72.7 | 75.9 | 71.6 | 30.2 |
| Extra Trees | 71.6 | 71.7 | 72.6 | 71.5 | 29.3 |
| SVC | 49.7 | 51 | 53 | 52 | 42 |
| Logistic Regression | 66.9 | 67 | 51.6 | 74.9 | 17.3 |
| AEC at $\alpha$=0.05 | 65 | 65.2 | 40.1 | 80.8 | 9.3 |
| AEC Tree Classifier | 71 | 71 | 62 | 76 | 19 |

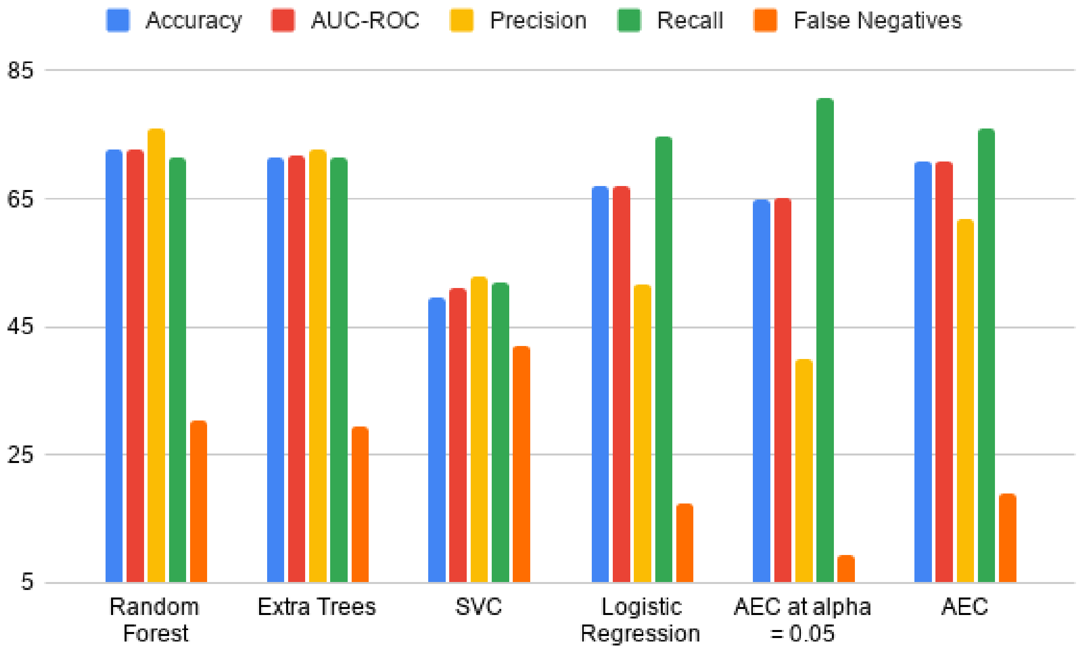


Figure 5.8: Results from Dataset 2 - AEC Tree Classifier versus Cost Sensitive Learning Classifiers to Predict Cardiac Disease

These results show that with the use of cost-sensitive learning, logistic regression's recall rate improves significantly with a drop in its accuracy and AUC-ROC score. The tree classifiers that are already using ensemble do not see much difference with the use of different class weights. The table shows two different values of AEC in the last two rows. The first one shows AEC with a strict bound on the number of false negatives, which is able to control the false negatives to under 10 percent. This percentage is very less compared to all the other approaches. However, the second value of the AEC Tree classifier, which was chosen as optimal, is able to produce a higher accuracy of over 71 percent with false negatives increased to 19 percent. This shows how AEC Tree classifier allows us to adjust the value of alpha to suit the classification problem.

### *5.2.4 Results on Australian Weather Dataset*

This section shows the results from the experiments on the Australian rain dataset. This dataset was introduced in chapter 3, and table 3.13 showed the results of AEC Classifier versus the other classifiers on this dataset without the use of cost-sensitive learning. The results from table 5.8 show the comparison of AEC Classifier with these classifiers using cost sensitive learning.

Table 5.8: Dataset 3 - AEC Tree Classifier vs Cost-Sensitive Learning Classifiers

| Classifiers | Accuracy | AUC-ROC | Precision | Recall | False Negatives |
|---|---|---|---|---|---|
| Random Forest | 80.6 | 70.9 | 87.9 | 87.3 | 42.6 |
| Extra Trees | 80.3 | 71.4 | 87.1 | 87.6 | 44.4 |
| SVC | 83.4 | 69.5 | 94.4 | 85.7 | 54.3 |
| Logistic Regression | 82.7 | 71.8 | 91.4 | 86.9 | 46.9 |
| AEC Classifier | 78.3 | 71.7 | 83.4 | 89.3 | 38.6 |

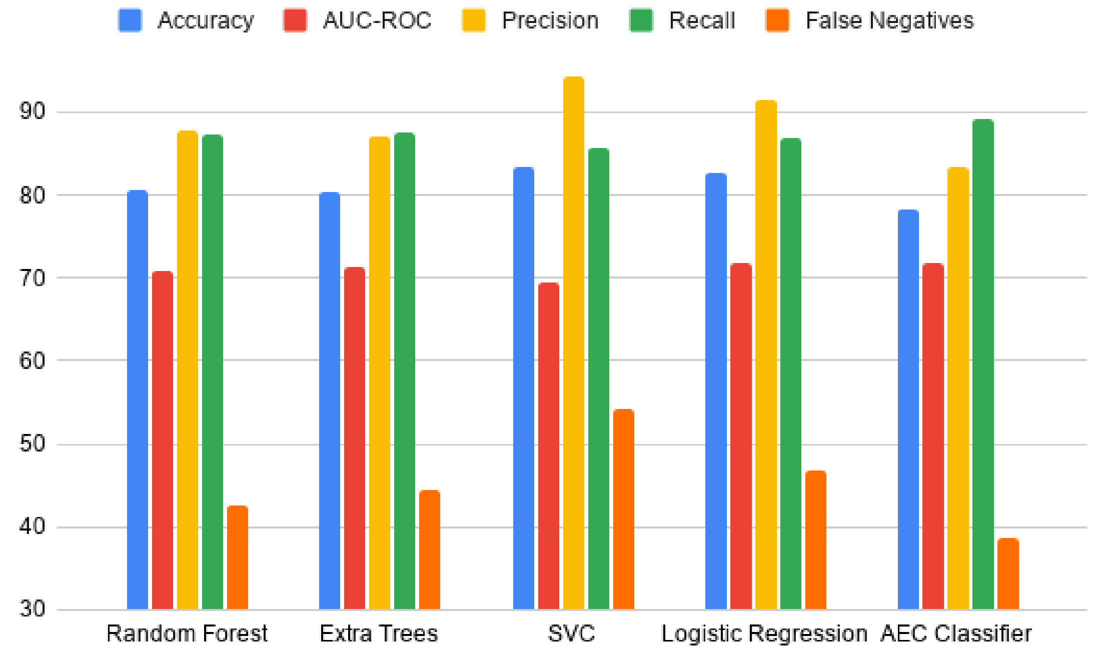


Figure 5.9: Results from Dataset 3 - AEC Tree Classifier versus ML Classifiers to Predict Rain

These results show that the AEC Classifier is able to produce the lowest percentage of false negatives compared to cost-sensitive learning classifiers. Even though the accuracy of our classifier is around 2 percent lower, it has AUC ROC scores comparable to the top classifiers. We notice that the recall rate is increasing with the use of class weights at the cost of reduced accuracy, compared to the results from chapter 3. The results from this table are also plotted in figure 5.9.

## 5.3 Conclusion

The AEC Tree Classifier seems like the best fit overall to handle data imbalance with the need of asymmetric error control. The use of limited sampling with this approach seems to produce the best results. The AEC Tree classifier uses ensemble techniques and the use of hellinger distance as the tree splitting criterion to handle the imbalance. Moreover, the use of NP Lemma to help with the threshold picking to achieve asymmetric error control outperforms the use of cost sensitive learning. Due to these results, this dissertation recommends the use of AEC Tree Classifier to predict the risk of cardiac disease.

Chapter 6

# CONCLUSION AND FUTURE WORK

This dissertation can be broadly divided into four main areas. Firstly, it is able to predict cardiac disease better than the existing methods used in the hospitals as shown by the testing on the Framingham dataset. The classifier that achieves this is called CVD Tree Classifier, and is explained in detail in chapter 2. Secondly, this approach is generalized for different binary classification problems outside the cardiac disease domain that require asymmetric error control. The resulting classifier is named AEC Classifier and it is able to provide control over the number of false negatives with high probability. Chapter 3 discusses this approach in detail. Thirdly, this research is able to extend the binary classification to multi-class problems to create the MCAEC Classifier, which is discussed in chapter 4. This is summarized in figure 6.1. Lastly, Chapter 5 discusses how the AEC Classifier compares to other approaches that handle data imbalance.

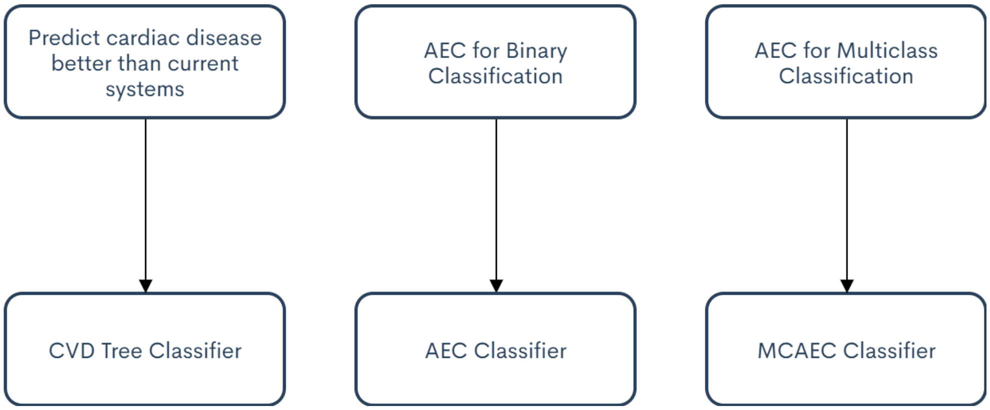

Predict cardiac disease better than current systems
AEC for Binary Classification
AEC for Multiclass Classification
CVD Tree Classifier
AEC Classifier
MCAEC Classifier

Figure 6.1: Asymmetric Error Control Classification

The results from this dissertation show that due to the data imbalance and the need to control one type of error, the tree-based AEC classifiers outperform the traditional classification models in predicting the risk of a 10-year CVD. Although other tree-based classifiers, such as random forest and bagging tree classifiers, are able to compete with the AEC Tree Classifier in terms of accuracy, AUC-ROC score and F1 score, yet the classifier proposed in this research emerges as the clear winner in predicting the risk of 10-year CVD by having the lowest number of false negatives. Furthermore, it also has the ability to easily reduce the false negatives further at a cost of reduced accuracy and F1 score, which is often acceptable in disease diagnosis.

Additionally, this dissertation has established a data adaptive way to pick the upper bound threshold $\alpha$ value that gives us optimal result balancing the acceptable number of false negatives with the performance metrics. This threshold value can easily be adjusted according to the use case of asymmetric error control. Despite providing control over the number of false negatives, this classifier is still able to produce accuracy scores that are comparable with other classifiers.

To summarize, this research proposes a classifier that provides full control over the number of false negatives in binary and multi-class classification problems. This classifier is able to predict the risk of 10-year CVD, not only more accurately, but also with full asymmetric error control. The methods used to create this classifier can be easily expanded to work with any classification problem where there is a need for asymmetric error control. This approach can have a massive impact in the medical domain, especially in disease diagnosis, where we typically need to control the number of false negatives.

As future research, multiclass classification with asymmetric error control should be explored without the use of OVR techniques. This will allow error control over multiple classes in a single run rather than maintaining the order of severity of misclassification of classes and applying the NP control recursively. Other techniques beyond NP lemma should also be explored to remove the two-class constraint. Additionally, the techniques used to predict cardiac disease should be tested in actual real world settings backed by comprehensive evaluation.

# REFERENCES

[1] “CPR Facts and Stats.” Cpr.heart.org, cpr.heart.org/en/resources/cpr-facts-and-stats.

[2] “Cardiovascular Diseases (CVDs).” World Health Organization, World Health Organization, www.who.int/news-room/fact-sheets/detail/cardiovascular-diseases-(cvds).

[3] “Cardiovascular Diseases.” World Health Organization, World Health Organization, www.who.int/health-topics/cardiovascular-diseases/tab=tab1.

[4] Bokhari, Wasif, et al. “Development and Use of a Tablet-Based Resuscitation Sheet for Improving Outcomes during Intensive Patient Care.” Proceedings of the 6th International Conference on Digital Health Conference - DH ’16, 2016, doi:10.1145/2896338.2896362.

[5] Muntner, Paul. “The Need for Accurate CVD Risk Prediction Equations.” American College of Cardiology, 22 June 2015, www.acc.org/latest-in-cardiology/articles/2015/06/22/07/20/the-need-for-accurate-cvd-risk-prediction-equations.

[6] Robinson, et al. “Performance of the Framingham Risk Models and Pooled Cohort Equations for Predicting 10-Year Risk of Cardiovascular Disease: a Systematic Review and Meta-Analysis.” BMC Medicine, BioMed Central, 1 Jan. 1970, bmcmedicine.biomedcentral.com/articles/10.1186/s12916-019-1340-7.

[7] Brindle, Peter M, et al. “The Accuracy of the Framingham Risk-Score in Different Socioeconomic Groups: a Prospective Study.” The British Journal of General Practice : the Journal of the Royal College of General Practitioners, Oxford University Press, Nov. 2005, www.ncbi.nlm.nih.gov/pmc/articles/PMC1570792/.

[8] Rana, J. S., Tabada, G. H., Solomon, M. D., Lo, J. C., Jaffe, M. G., Sung, S. H., Ballantyne, C. M., Go, A. S. (2016). Accuracy of the Atherosclerotic Cardiovascular Risk Equation in a Large Contemporary, Multiethnic Population. Journal of the American College of Cardiology, 67(18), 2118–2130. https://doi.org/10.1016/j.jacc.2016.02.055

[9] Brindle, Peter M, et al. “The Accuracy of the Framingham Risk-Score in Different Socioeconomic Groups: a Prospective Study.” The British Journal of General Practice : the Journal of the Royal College of General Practitioners, Oxford University Press, Nov. 2005, www.ncbi.nlm.nih.gov/pmc/articles/PMC1570792/.

[10] Brindle, Peter M, et al. “The Accuracy of the Framingham Risk-Score in Different Socioeconomic Groups: a Prospective Study.” The British Journal of General Practice : the Journal of the Royal College of General Practitioners, Oxford University Press, Nov. 2005, www.ncbi.nlm.nih.gov/pmc/articles/PMC1570792/.

[11] Irjet net 2020 [online] Available at: ¡https://www.irjet.net/archives/V6/i1/IRJET-V6I1233.pdf¿ [Accessed 1 June 2020].

[12] “Body Basics.” Rady Children’s Hospital-San Diego, www.rchsd.org/health-articles/heart-and-circulatory-system/.

[13] Tong, Xin, et al. “Neyman-Pearson Classification Algorithms and NP Receiver Operating Characteristics.” Science Advances, American Association for the Advancement of Science, 1 Feb. 2018, advances.sciencemag.org/content/4/2/eaao1659.

[14] “Random Forest vs Extra Trees.” The Kernel Trip, 2 Sept. 2018, www.thekerneltrip.com/statistics/random-forest-vs-extra-tree/.

[15] Web.stanford.edu. 2020. [online] Available at: ¡https://web.stanford.edu/˜lmackey/stats300a/doc/stats300a-fall15-lecture13.pdf¿ [Accessed 1 June 2020].

[16] Cohen, A., 2020. 10 The Role Of Order Statistics In Estimating Threshold Parameters.

[17] “BioLINCC: Framingham Heart Study-Cohort (FHS-Cohort).” National Heart Lung and Blood Institute, U.S. Department of Health and Human Services, biolincc.nhlbi.nih.gov/studies/framcohort/.

[18] Bishop, et al. “SMOTE for High-Dimensional Class-Imbalanced Data.” BMC Bioinformatics, BioMed Central, 1 Jan. 1970, bmcbioinformatics.biomedcentral.com/articles/10.1186/1471-2105-14-106.

[19] Narkhede, Sarang. “Understanding AUC - ROC Curve.” Medium, Towards Data Science, 26 May 2019, towardsdatascience.com/understanding-auc-roc-curve-68b2303cc9c5.

[20] Bitton, Asaf, and Thomas A Gaziano. “The Framingham Heart Study’s Impact on Global Risk Assessment.” Progress in Cardiovascular Diseases, U.S. National Library of Medicine, 2010, www.ncbi.nlm.nih.gov/pmc/articles/PMC2904478/.

[21] Peace, J. M., Yuen, T. C., Borak, M. H., Edelson, D. P. (2014). Tablet-based cardiac arrest documentation: A pilot study. Resuscitation, 85(2), 266–269. Moura, D., el-Nasr, M. S., Shaw, C. D. 2011.Google ScholarCross Ref

[22] HOSPITAL CODE DOCUMENTATION. (2010). Retrieved August 10, 2015, from http://www.resuscitationcentral.com/documentation/hospital-code-data/

[23] Deo RC. Machine learning in medicine. Circulation. 2015; 132:1920–1930

[24] Jabeen A, Ahmad N, Raza K. Machine learning-based state-of-the-art methods for the classification of rna-seq data. In: Dey N, Ashour AS, Borra S, eds. Classification in BioApps: Automation of Decision Making. Cham: Springer International Publishing; 2018:133–172.

[25] Ali, K. (1995). On the link between error correlation and error reduction in decision tree ensembles. Technical report, Department of Information and Computer Science, University of California, Irvine.

[26] Breiman, L., Friedman, J., Olsen, R.,  Stone, C. (1984). Classification and regression trees. Wadsworth International.

[27] Buntine, W.,  Niblett, T. (1992), A further comparison of splitting rules for decision-tree induction. Machine Learning, 8, 75–85.

[28] Breiman, L. (2001). Random forests. Machine Learning, Ch 45, Pages 5–32.

[29] D. A. Cieslak, T. R. Hoens, N. V. Chawla, et al., "Hellinger Distance Decision Trees Are Robust and Skew-insensitive", Data Mining and Knowledge Discovery, vol. 24, no. 1, pp. 136–158, 2012.

[30] Ieeexplore.ieee.org. 2020. Rate-Optimal Meta Learning Of Classification Error - IEEE Conference Publication.

[31] Ieeexplore.ieee.org. Framingham Risk Score Calculator - Family Practice Management.

[32] W. Bokhari and A. Bansal, "Asymmetric Error Control for Binary Classification in Medical Disease Diagnosis," 2020 IEEE Third International Conference on Artificial Intelligence and Knowledge Engineering (AIKE), Laguna Hills, CA, USA, 2020, pp. 25-32, doi: 10.1109/AIKE48582.2020.00013.

[33] Tong, X., Feng, Y., amp; Li, J. (2018, February 01). Neyman-Pearson classification algorithms and NP receiver operating characteristics.

[34] Dong, R. (2020, February 21). Summary for Neyman-Pearson Classification Algorithms.

[35] Hajian-Tilaki, K. (2013). Receiver Operating Characteristic (ROC) Curve Analysis for Medical Diagnostic Test Evaluation. Retrieved July 02, 2020, from https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3755824

[36] Chakure, A. (2020, June 07). Decision Tree Classification.

[37] Sug, H. (2012, November 07). Applying Randomness Effectively Based on Random Forests for Classification Task of Datasets of Insufficient Information.

[38] Brownlee, J. (2020, June 25). How to Develop an Extra Trees Ensemble with Python.

[39] Priya Ranganathan, R., 2020. Common Pitfalls In Statistical Analysis: Logistic Regression. [online] PubMed Central (PMC).

[40] Ieeexplore.ieee.org. 2020. Fault Detection In Wireless Sensor Networks Through SVM Classifier - IEEE Journals  Magazine.

[41] 2020. Congestive Heart Failure Detection Using Random Forest Classifier.

[42] Zhao, Y., Wong, Z., amp; Tsui, K. (2018, May 22). A Framework of Rebalancing Imbalanced Healthcare Data for Rare Events' Classification: A Case of Look-Alike Sound-Alike Mix-Up Incident Detection.

[43] How is Splitting Decided for Decision Trees? (2018, November 07).

[44] Dl.acm.org. 2020. Comprehensible Classification Models: A Position Paper: ACM SIGKDD Explorations Newsletter: Vol 15, No 1.

[45] Dl.acm.org. 2020. Big Data Classification: Problems And Challenges In Network Intrusion Prediction With Machine Learning: ACM SIGMETRICS Performance Evaluation Review: Vol 41, No 4.

[46] Zhai, X., Oliver, A., Kolesnikov, A. and Beyer, L., 2020. S4L: Self-Supervised Semi-Supervised Learning.

[47] Menon, A. and Williamson, R., 2020. The Cost Of Fairness In Binary Classification. [online] PMLR.

[48] Bechavod, Y. and Ligett, K., 2020. Penalizing Unfairness In Binary Classification.

[49] Ieeexplore.ieee.org. 2020. Rate-Optimal Meta Learning Of Classification Error - IEEE Conference Publication.

[50] Das, H., Naik, B. and Behera, H., 2020. Medical disease analysis using Neuro-Fuzzy with Feature Extraction Model for classification. Informatics in Medicine Unlocked, 18, p.100288.

[51] Ieeexplore.ieee.org. 2020. An Investigation Of Transfer Learning And Traditional Machine Learning Algorithms - IEEE Conference Publication.

[52] Abdar, M., Ksiażek, W., Acharya, U., Tan, R., Makarenkov, V. and Pławiak, P., 2019. A new machine learning technique for an accurate diagnosis of coronary artery disease. Computer Methods and Programs in Biomedicine, 179, p.104992.

[53] Chuanfeng, S. and Shaolin, J., 2020. The Neyman-Pearson Lemma For Convex Expectations.

[54] Web.stanford.edu. 2020. Stats 300A: Theory Of Statistics.

[55] Order Statistics and Inference, 1991. Cohen–Whitten Estimators: Using First Order Statistics. pp.273-297.

[56] Cervantes, B., Monroy, R., Medina-Pérez, M. A., Gonzalez-Mendoza, M., amp; Ramirez-Marquez, J. (2018). Some features speak loud, but together they all speak louder: A study on the correlation between classification error and feature usage in decision-tree classification ensembles.

[57] Wray Buntine Tim Niblett .January 1992.A Further Comparison of Splitting Rules for Decision-Tree Induction

[58] Bishop, C., H. He, E., S. Daskalaki, I., S. Ramaswamy, K., MA. Shipp, K., N. Iizuka, M., . . . LD. Miller, J. (1970, January 01). SMOTE for high-dimensional class-imbalanced data.

[59] "BioLINCC: Framingham Heart Study-Cohort (FHS-Cohort)." National Heart Lung and Blood Institute, U.S. Department of Health and Human Services, biolincc.nhlbi.nih.gov/studies/framcohort/.

[60] Q. Mai, Y. Yang, H. Zou, Multiclass sparse discriminant analysis. https://arxiv.org/abs/1504.05845 (2015)

[61] Dartmouth class COSC 89.20, Data Science for Health. https://www.kaggle.com/sulianova/cardiovascular-disease-dataset

[62] MIT Technology Review. 2020. What Is Machine Learning?. [online] Available at www.technologyreview.com/2018/11/17/103781/what-is-machine-learning-we-drew-you-another-flowchart/

[63] Deo RC. Machine learning in medicine. Circulation. 2015; 132:1920–1930

[64] Jabeen A, Ahmad N, Raza K. Machine learning-based state-of-the-art methods for the classification of rna-seq data. In: Dey N, Ashour AS, Borra S, eds. Classification in BioApps: Automation of Decision Making. Cham: Springer International Publishing; 2018:133–172.

[65] Ali, K. (1995). On the link between error correlation and error reduction in decision tree ensembles. Technical report, Department of Information and Computer Science, University of California, Irvine.

[66] Ieeexplore.ieee.org. 2020. Rate-Optimal Meta Learning Of Classification Error - IEEE Conference Publication.

[67] Freitas, A. A. Comprehensible classification models: A position paper. ACM SIGKDD Explorations Newsletter, 15(1), 2020.

[68] Newman, T. What to know about arrhythmia. Medical News Today, 2020.

[69] Bokhari, W. Predicting cardiovascular disease (cvd) using machine learning. AIMed Cardiology, 2020.

[70] David A. Cieslak, T. Ryan Hoens, N. V. C. W. P. K. Hellinger distance decision trees are robust and skewinsensitive. 2011.

[71] Golinska, A. K. Towards prediction of heart arrhythmia ´ onset using machine learning. Nature Public Health Emergency Collection, 2020.

[72] Thomas C.W. Landgrebe, R. P. D. Approximating the multiclass roc by pairwise analysis. Pattern Recognition Letters, 2007.

[73] Brownlee, J. One-vs-rest and one-vs-one for multi-class classification. 2020a.

[74] Brownlee, J. Error-correcting output codes (ecoc) for machine learning. 2020b.

[75] Dietterich, T. G. and Bakiri, G. Solving multiclass learning problems via error-correcting output codes. 1995.

[76] Guvenir, H. A., Acar, B., Demiroz, G., and Cekin, A. A supervised machine learning algorithm for arrhythmia analysis. In Computers in Cardiology 1997, pp. 433–436, 1997.

[77] Hoare, J. How is splitting decided for decision trees? 2018.

[78] Hajian-Tilaki, K. Receiver operating characteristic (roc) curve analysis for medical diagnostic test evaluation. Caspian journal of internal medicine, 4:627–635, 09 2013.

[79] Dong, R. Summary for neyman-pearson classification algorithms. 2020.

[80] Suthaharan, S. Big data classification: Problems and challenges in network intrusion prediction with machine learning. ACM SIGMETRICS Performance Evaluation Review, 41(4), 2014.

[81] Xiaohua Zhai, Avital Oliver, A. K. and Beyer, L. S4l: Self-supervised semi-supervised learning, 2019

[82] Aditya Krishna Menon, R. C. W. The cost of fairness in binary classification. Proceedings of Machine Learning Research, 81:107–118, 2018.

[83] Bechavod, Y. and Ligett, K. Penalizing Unfairness in Binary Classification. PhD thesis, Hebrew University of Jerusalem, 2018.

[84] Tong, X., Feng, Y., and Li, J. J. Neyman-pearson classification algorithms and np receiver operating characteristics. Science Advances, 4(2), 2018.

[85] Weiss, K. R. and Khoshgoftaar, T. M. An investigation of transfer learning and traditional machine learning algorithms. In 2016 IEEE 28th International Conference on Tools with Artificial Intelligence (ICTAI), pp. 283–290, 2016.

[86] Moloud Abdar, Wojciech Ksiazek, U. R. A. R.-S. T. V. M. ˙ and Pławiak, P. A new machine learning technique for an accurate diagnosis of coronary artery disease. Computer Methods and Programs in Biomedicine, 179, 2019.

[87] Sun, C. and Ji, S. The neyman-pearson lemma for convex expectations. 12 2019.

[88] Mackey, L. Theory of statistics. Stats 300A, 2020.

[89] Balakrishnan, N. and Cohen, A. C. Chapter 8 - co-hen–whitten estimators: Using first order statistics. In Order Statistics and Inference, pp. 273–297. Academic Press, San Diego, 1991.

[90] Barbara Cervantes, Ra ´ ul Monroy, M. A. M.-P.-M. G.-M. ´ and Ramirez-Marquez, J. A study on the correlation between classification error and feature usage in decisiontree classification ensembles. Engineering Applications of Artificial Intelligence, 67:270–282, 2018.

[91] Buntine, W. and Niblett, T. A further comparison of splitting rules for decision-tree induction. Machine Learning, 8: 75–85, 01 1992.

[92] Blagus, R. and Lusa, L. Smote for high-dimensional classimbalanced data. BMC bioinformatics, 14:106, 03 2013.

[93] James, G. and Hastie, T. The error coding method and picts.Journal of Computational and Graphical Statistics, 7(3):377–387, 1998.

[94] Mai, Q., Yang, Y., and Zou, H. Multiclass sparse discriminant analysis, 2015

[95] Vina Spiehler, Ph.D., D. Glass classification, 1987.

[96] Alaa Mabrouk Salem Omar, et al. "Artificial Intelligence-Based Assessment of Left Ventricular Filling Pressures From 2-Dimensional Cardiac Ultrasound Images". JACC: Cardiovascular Imaging 11. 3(2018): 509-510.

[97] "Lab Algorithms." Zebra Medical Vision — Medical Imaging amp; AI. Web.

[98] KenSci. "AI Platform for Digital Health." KenSci. Web.

[99] "What's Your Heart Telling You?" Cardiogram. Web.

[100] Ballinger, Brandon, Hsieh, Johnson, Singh, Avesh, Sohoni, Nimit, Wang, Jack, Tison, Geoffrey, Marcus, Gregory, Sanchez, Jose, Maguire, Carol, Olgin, Jeffrey, AND Pletcher, Mark. "DeepHeart: Semi-Supervised Sequence Learning for Cardiovascular Risk Prediction" AAAI Conference on Artificial Intelligence (2018): n. pag. Web.

[101] "Artificial Intelligence Used to Detect Early-stage Heart Disease." European Pharmaceutical Review. 24 Jan. 2019. Web.

[102] Attia, Z.I., Kapa, S., Lopez-Jimenez, F. et al. Screening for cardiac contractile dysfunction using an artificial intelligence–enabled electrocardiogram. Nat Med 25, 70–74 (2019). https://doi.org/10.1038/s41591-018-0240-2

[103] "Heart Health Calculator Predicts the Age of Hearts and More Dangerous Risks." European Pharmaceutical Review. Web.

[104] Sriram Somanchi, Samrachana Adhikari, Allen Lin, Elena Eneva, and Rayid Ghani. 2015. Early Prediction of Cardiac Arrest (Code Blue) using Electronic Medical Records. In Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '15). Association for Computing Machinery, New York, NY, USA, 2119–2126. DOI:https://doi.org/10.1145/2783258.2788588

[105] Lipton, Zachary Chase. "The Hard Problems AI Can't (Yet) Touch." Blog post. Kdnuggets. July 2016. Web.

[106] Ballinger, Brandon. "Three Challenges for Artificial Intelligence in Medicine." Blog post. Cardiogram. 19 Sept. 2016. Web.

[107] "Neyman-Pearson Lemma." Fisher Stats. Web.

[108] ”Communication Complexity.” Hellinger Distance. TIFR, 23 Sept. 2011. Web.

[109] Dubov, Evgeni. ”Classifying Imbalanced Data Using Hellinger Distance.” Mediu. 26 Mar. 2019. Web.

[110] Dong, Ruhan. “Summary for Neyman-Pearson Classification Algorithms.” Medium, Medium, 7 Nov. 2020, medium.com/@ruhandong/summary-for-neyman-pearson-classification-algorithms-a0c9595632a9.

[111] Dittman, David. “Ensemble vs. Data Sampling: Which Option Is Best Suited to Improve Classification Performance of Imbalanced Bioinformatics Data?” IEEE Xplore, ICTAI, Nov. 2015, ieeexplore.ieee.org/document/7372202.

[112] Feng, et al. “Class Imbalance Ensemble Learning Based on the Margin Theory.” MDPI, Multidisciplinary Digital Publishing Institute, 18 May 2018, www.mdpi.com/2076-3417/8/5/815.

[113] Sun, T.; Jiao, L.; Feng, J.; Liu, F.; Zhang, X. Imbalanced Hyperspectral Image Classification Based on Maximum Margin. IEEE Geosci. Remote Sens. Lett. 2015, 12, 522–526

[114] Mellor, A.; Boukir, S.; Haywood, A.; Jones, S. Exploring issues of training data imbalance and mislabelling on random forest performance for large area land cover classification using the ensemble margin. J. Photogramm. Remote Sens. 2015, 105, 155–168.

# APPENDIX A

# NP LEMMA PROOF

## A.1 Neyman-Pearson Lemma

Consider the simple null hypothesis

$$H_0 : \theta = \theta_0 \tag{A.1}$$

versus the simple alternative

$$H_A : \theta = \theta_1 \tag{A.2}$$

Consider the rejection region R given by size

$$\alpha = P\left(\mathbf{X} \in R \mid \theta_0\right) \tag{A.3}$$

Let R be any other rejection region of size

$$\alpha = P\left(\mathbf{X} \in R \mid \theta_0\right) \tag{A.4}$$

Then the likelihood ratio test is more powerful than this other test, that is

$$P\left(\mathbf{X} \in R^* \mid \theta_1\right) \leq P\left(\mathbf{x} \in R \mid \theta_1\right) \tag{A.5}$$

Source: Taken directly from Hurvich, Clifford. ”The Neyman Pearson Lemma.” People.stern.nyu.edu. Web.

## A.2 Neyman-Pearson Proof

This is given in the case of $X$ having a density. For a set $A$ let $\mathrm{I}_A$ be the indicator function of this set. Thus we can make a 1 to 1 correspondence between a rejection region and its indicator function

$$\begin{aligned} R \leftrightarrow \quad & \mathrm{I}_R \\ R^* \leftrightarrow & \mathrm{I}_{R^*} \end{aligned}$$

Next notice that

$$\begin{aligned} cf(\mathbf{x};\theta_1) - f(\mathbf{x};\theta_0) > 0 \quad & \text{if} \quad \mathrm{I}_R(\mathbf{x}) = 1 \\ cf(\mathbf{x};\theta_1) - f(\mathbf{x};\theta_0) \leq 0 \quad & \text{if} \quad \mathrm{I}_R(\mathbf{x}) = 0 \end{aligned}$$

Thus for every possible value of $\mathbf{x}$ we obtain

$$\mathrm{I}_{R^*}(\mathbf{x})\left(cf(\mathbf{x};\theta_1) - f(\mathbf{x};\theta_0)\right) \leq \mathrm{I}_R(\mathbf{x})\left(cf(\mathbf{x};\theta_1) - f(\mathbf{x};\theta_0)\right)$$

Remark: If we integrate the left hand side with respect to (w.r.t.) $\mathbf{x}$ we get

$$\int \mathrm{I}_{R^*}(\mathbf{x})\left(cf(\mathbf{x};\theta_1) - f(\mathbf{x};\theta_0)\right) d\mathbf{x} = cP\left(R^* \mid \theta_1\right) - P\left(R^* \mid \theta_0\right)$$

Thus we get $c$ times the power of $R^*$ minus the size of $R^*$. The other side will give a similar term. Integrate both sides w.r.t. $\mathbf{x}$. This gives

$$cP\left(R^* \mid \theta_1\right) - P\left(R^* \mid \theta_0\right) \leq cP\left(R \mid \theta_1\right) - P\left(R \mid \theta_0\right)$$

Rearranging gives

$$P\left(R \mid \theta_0\right) - P\left(R^* \mid \theta_0\right) \leq c\left\{P\left(R \mid \theta_1\right) - P\left(R^* \mid \theta_1\right)\right\}$$

Thus if $R^*$ is a test (or rejection region) of size $\leq \alpha = P\left(R \mid \theta_0\right)$, the LHS is $\geq 0$, and hence the same is true for the RHS (recall $c > 0$ ) giving

$$P\left(R \mid \theta_1\right) - P\left(R^* \mid \theta_1\right) \geq 0$$

This later piece says that the power with rejection region $R$ is greater than or equal to the power with rejection region $R^*$. Thus the test of hypothesis, of a simple null versus simple alternative hypothesis, based on the likelihood ratio (1) is more powerful than any other test of the same or smaller size.

Source: Taken directly from Hurvich, Clifford. "The Neyman Pearson Lemma." People.stern.nyu.edu. Web.

# APPENDIX B

# HELLINGER DISTANCE DERIVATION

Let $P = \{p_i\}_{i \in [n]}, Q = \{q_i\}_{i \in [n]}$ be two probability distributions supported on $[n]$. A natural way of defining a distance between them is to consider the $\ell_1$ -distance between the probability vectors $P$ and $Q$.

$$\|P - Q\|_1 = \sum_{i \in [n]} |p_i - q_i|$$

The total variation distance, denoted by $\Delta(P, Q)$ (and sometimes by $\|P - Q\|_{TV}$), is half the above quantity. It is an easy exercise to check that

$$\Delta(P, Q) = \max_{S \subseteq [n]} |P(S) - Q(S)|$$

Because of the above equality, this is also referred to as the statistical distance. Taking the $\ell_1$ norm of the difference made sense because $P$ and $Q$ where unit vectors according to the $\ell_1$ norm. Since $\sqrt{P} = \left(\sqrt{p_1}, \sqrt{p_2}, \ldots \sqrt{p_n}\right)$ is a unit vector according to $\ell_2$ norm, we can also consider the $\ell_2$ norm of the difference of the square root vectors.
Definition 12.1 (Hellinger Distance). For probability distributions $P = \{p_i\}_{i \in [n]}, Q = \{q_i\}_{i \in [n]}$ supported on $[n]$, the Hellinger distance between them is defined as

$$h(P, Q) = \frac{1}{\sqrt{2}} \cdot \|\sqrt{P} - \sqrt{Q}\|_2$$

By definition:

$$h^2(P, Q) = 1 - F(P, Q)$$

Lemma 12.2 (Hellinger vs. total variation).

$$h^2(P, Q) \leq \Delta(P, Q) \leq \sqrt{h^2(P, Q)\left(2 - h^2(P, Q)\right.} \leq \sqrt{2}h(P, Q)$$

Proof. For the first inequality,

$$\begin{aligned} h^2(P, Q) &= \frac{1}{2} \sum_i |\sqrt{p_i} - \sqrt{q_i}||\sqrt{p_i} - \sqrt{q_i}| \leq \frac{1}{2} \sum_i |\sqrt{p_i} - \sqrt{q_i}| \left(\sqrt{p_i} + \sqrt{q_i}\right) \\ &\leq \frac{1}{2} \sum_i |p_i - q_i| = \Delta(P, Q) \end{aligned}$$

For the last two inequalities,

$$\Delta^2(P, Q) = \frac{1}{4} \left( \sum_{i \in [n]} |p_i - q_i| \right)^2 = \frac{1}{4} \left( \sum_{i \in [n]} \left(\sqrt{p_i} - \sqrt{q_i}\right) \left(\sqrt{p_i} + \sqrt{q_i}\right) \right)^2$$

Source: Taken directly from ”Communication Complexity.” Hellinger Distance. TIFR,23 Sept.2011.Web.

$$\begin{aligned}
&\le \tfrac{1}{4}\left(\sum_{i\in[n]}\left(\sqrt{p_i}-\sqrt{q_i}\right)^2\right)\left(\sum_{i\in[n]}\left(\sqrt{p_i}+\sqrt{q_i}\right)^2\right)\\
&\le \tfrac{1}{2}\cdot h^2(P,Q)\cdot\left(2+2\sum_{i\in[n]}\sqrt{p_i}\sqrt{q_i}\right)\\
&\le h^2(P,Q)\cdot\left(2-h^2(P,Q)\right)\le\sqrt{2}h(P,Q)
\end{aligned}$$

Cut and paste property: In the fooling set argument, we saw that if inputs $(x, y)$ and $(x', y')$ have the same transcript in a deterministic communication protocol, then $(x', y)$ and $(x, y')$ must have the same transcript. This rectangle property can be extended to private coins randomized protocols using Hellinger distance in the follows sense: if the transcript distributions for inputs $(x, y)$ and $(x', y')$ are close in Hellinger distance, then so are the transcript distributions for $(x', y)$ and $(x, y')$
Lemma 12.3 (Cut-and-Paste). Let $\mathcal{P}$ be a randomized private coins protocol and $\Pi_{x,y}$ denote the (randomized) transcript on input $x, y$. Then,

$$h^2\left(\Pi_{x,y},\Pi_{x',y'}\right)=h^2\left(\Pi_{x',y},\Pi_{x,y'}\right)$$

Proof. We can think of a randomized private coin protocol working on input $(x, y)$ as a deterministic protocol on the extended inputs $((x, R_A), (y, R_B))$, where the additional inputs $R_A$ and $R_B$ are chosen according to the suitable private coins distribution. From the rectangle property of deterministic protocols, we have that for any fixed transcript $\tau$ the set of extended inputs that gives rise to it form a rectangle, say Rect $_\tau = S_\tau \times T_\tau$. Now, let's consider the probability that transcript $\tau$ arises for inputs $x$ and $y$.

$$\begin{aligned}
\Pr_{R_A,R_B}\left[\Pi\left(x,y,R_A,R_B\right)=\tau\right] &= \Pr_{R_A,R_B}\left[\left(\left(x,R_A\right),\left(y,R_B\right)\right)\in\mathrm{Rect}_\tau\right]\\
&= \Pr_{R_A,R_B}\left[\left(x,R_A\right)\in S_\tau \text{ and } \left(y,R_B\right)\in T_\tau\right]\\
&= \Pr_{R_A}\left[\left(x,R_A\right)\in S_\tau\right]\cdot\Pr_{R_B}\left[\left(y,R_B\right)\in T_\tau\right]
\end{aligned}$$

This splitting of probabilities follows from the independence of Alice and Bob's private coins $R_A$ and $R_B$ and is used to proved the lemma as follows.

$$\begin{aligned}
&1-h^2\left(\Pi_{x,y},\Pi_{x',y'}\right))=F\left(\Pi_{x,y},\Pi_{x',y'}\right)\\
&=\sum_\tau\sqrt{\Pr\left[\Pi_{x,y}=\tau\right]\cdot\Pr\left[\Pi_{x',y'}=\tau\right]}
\end{aligned}$$

The above cut-and-paste lemma can be extended to communication protocols for $t$ parties. Lemma 12.4 (multiparty cut-and-paste). For any $v \in \{x_1, y_1\} \times \{x_2, y_2\} \cdots \times \{x_t, y_t\}$

$$h^2\left(\Pi_{x_1,x_2,\ldots x_t},\Pi_{y_1,y_2,\ldots y_t}\right)=h^2\left(\Pi_v,\Pi_{\bar{v}}\right)$$

Lemma 12.5 (Hellinger vs. Information [Lin91]). Let $Z$ be a random variable taking values in $\{z_1, z_2\}$ equally likely and $\Pi$ a randomized function of $Z$. Then,

$$I[Z:\Pi(Z)]\ge h^2\left(\Pi_{z_1},\Pi_{z_2}\right)$$

Source: Taken Directly from ”Communication Complexity.” Hellinger Distance. TIFR,23 Sept.2011.Web.

# APPENDIX C

# EMPIRICAL ERROR VS POPULATION ERROR

This simulation study provides an overview of how controlling the empirical or test data error does not correspond to controlling the population error. This simulation and figures C1 and C2 are taken directly from Dong's published article [110].

In this cartoon at human population, color is used to represent medical test results indicating the person's health status. "-" sign indicates the person actually does not have the disease; while "+" sign indicates the person has the disease.

A very fundamental question for physicians is: where should they set the threshold on the medical test result so that they can predict weather the patient has the disease or not. False negative rate is significantly concerned, because mis-predicting a patient with serious disease as a healthy person will delay the treatment and cause a life loss.

Setting the threshold is not difficult when observing the whole population. A threshold that controls the false negative rate under 1 percent while minimizing the false positive rate as threshold 4 can be easily picked up.

However in reality, we only observe a random sample from the population. A simple and intuitive way to set the threshold is such that false negative rate on this sample is under 1 percent. However, it turns out that the population false negative rate given this threshold is 7 percent.

If another random sample is seen, the threshold picked by controlling false negative rate (FNR) under 1 percent empirically can cause a population false negative rate at 17 percent. In fact, as we observe more and more random samples, a distribution of selected threshold will be seen. For half of the chance, we will choose a threshold with a false negative rate greater than 1 percent.

Source: Taken directly from Dong, Ruhan. "Summary for Neyman-Pearson Classification Algorithms." Medium, Medium, 7 Nov. 2020, medium.com/@ruhandong

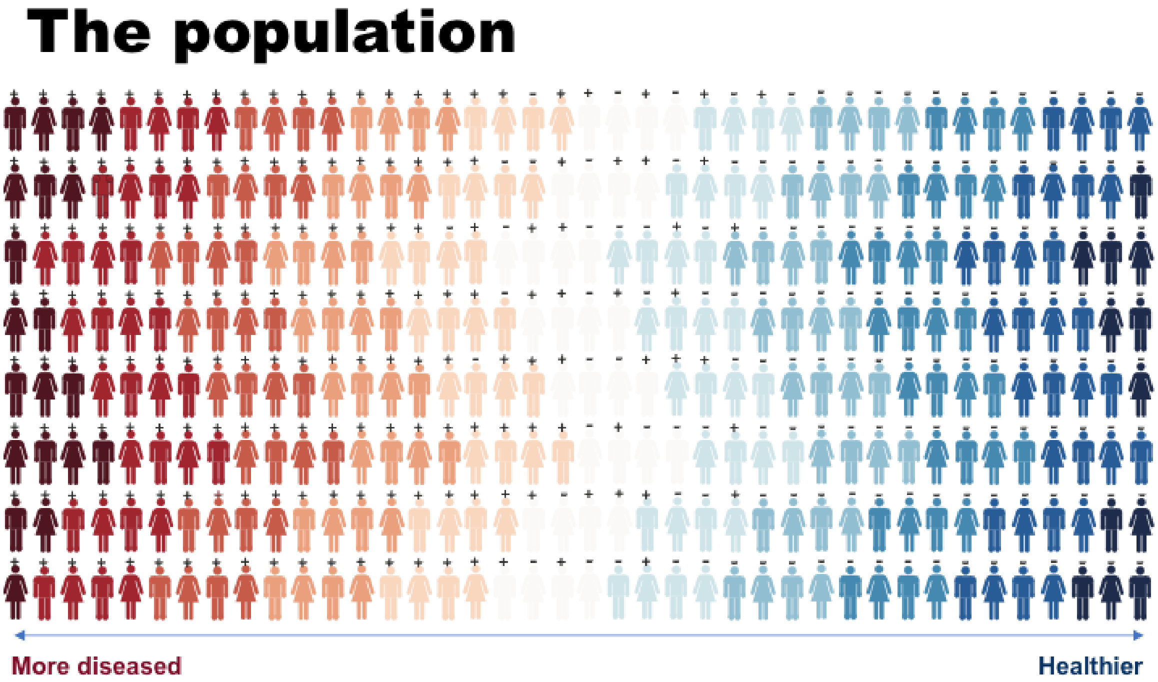


Figure C.1: Whole Population (Source: taken directly from Summary for Neyman-Pearson Classification [110])

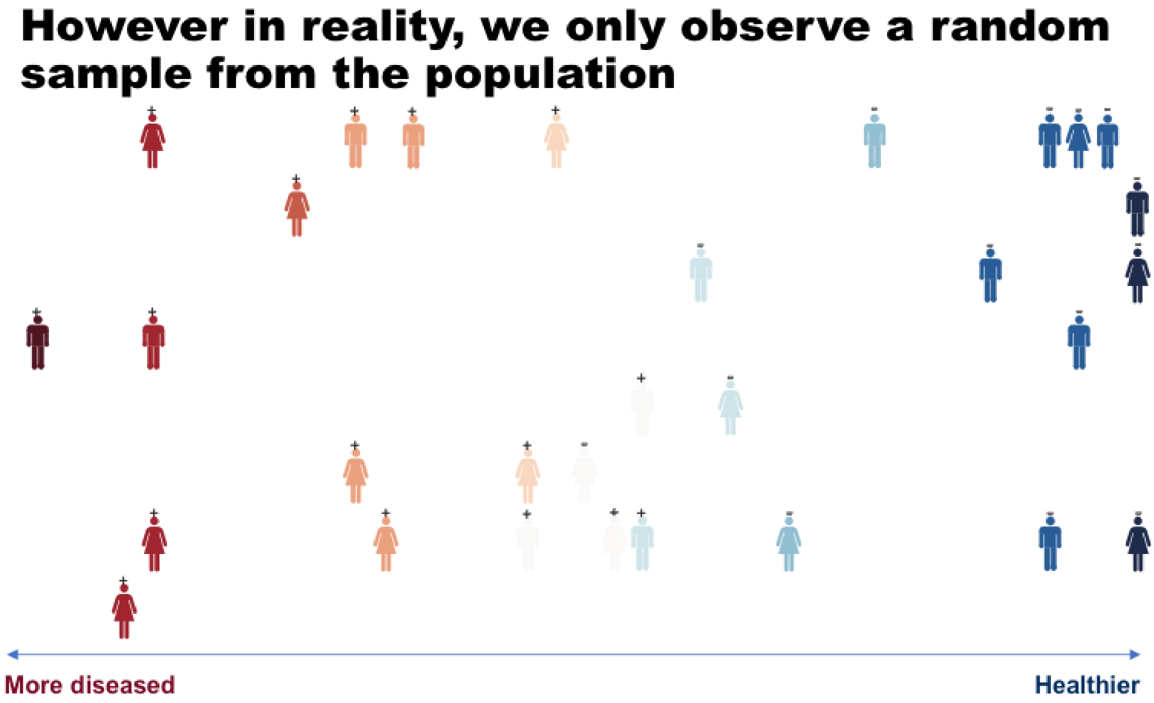

However in reality, we only observe a random sample from the population
More diseased
Healthier

Figure C.2: Random Sample from Population (Source: taken directly from Summary for Neyman-Pearson Classification [110])

# APPENDIX D

# HELLINGER DISTANCE SPLITTING CRITERION STUDY

These are the results obtained from my cardiac disease dataset.

Table D.1: Accuracy of Random Forest on Framingham Dataset

| **Splitting Criterion** | **Balanced Data** | **Imbalanced Data** |
|---|---|---|
| Gini | 0.88 | 0.83 |
| Entropy | 0.87 | 0.83 |
| Hellinger Distance | 0.85 | 0.85 |

Table D.2: Accuracy of Extra Trees on Framingham Dataset

| **Splitting Criterion** | **Balanced Data** | **Imbalanced Data** |
|---|---|---|
| Gini | 0.88 | 0.82 |
| Entropy | 0.88 | 0.82 |
| Hellinger Distance | 0.85 | 0.84 |

Table D.3: Accuracy of AEC Tree on Framingham Dataset

| **Splitting Criterion** | **Balanced Data** | **Imbalanced Data** |
|---|---|---|
| Gini | 0.85 | 0.81 |
| Entropy | 0.85 | 0.80 |
| Hellinger Distance | 0.84 | 0.84 |

This is a splitting criterion study published in the Data Mining Journal [70] and performed by Evgeni Dublov [109].

This study performed the following calculations to calculate the Gini, Entropy and Hellinger distance scores for imbalanced dataset.

Hellinger Distance $= \sqrt{\left\{\sqrt{\frac{N_A^{\text{left}}}{N_A^{\text{parent}}}} - \sqrt{\frac{N_B^{\text{left}}}{N_B^{\text{parent}}}}\right\}^2 + \left\{\sqrt{\frac{N_A^{\text{right}}}{}}\right.}$

$\mathrm{N}_A^{\text{parent}} \; - \sqrt{\frac{N_B^{\text{right}}}{N_B^{\text{parent}}}}^2$ $N_A^{\text{left}}$ = number of Class A samples in left child

Entropy $= \sum_{i=1}^{c} -p_i \log_2 (p_i)$

Gini $= 1 - \sum_{i=1}^{c} (p_i)^2$ Gini drop = Gini $_{\text{parent}} - W_{\text{left}}$ Gini $_{\text{left}} - W_{\text{right}}$ Gini $_{\text{right}}$

Entropy drop = Entropy $_{\text{parent}} - W_{\text{left}}$ Entropy$_left$-W$_{\text{right}}$ Entropy$_right W_{\text{left}}$ =

Population $_{\text{left}}$ / Population $_{\text{parent}}$

The results show that hellinger distance performs the best as splitting criterion as shown in the table C1 and figure C1. The results are compared at the decision tree split on 0.783 value of the feature X.

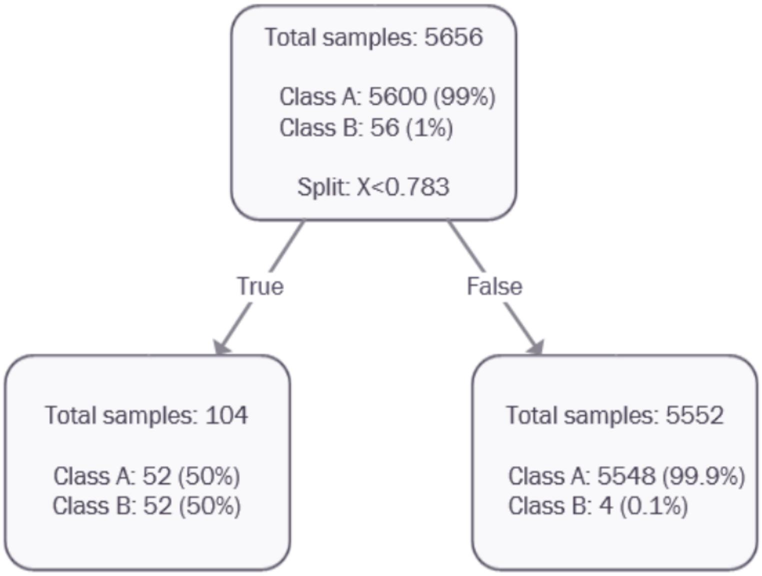


Figure D.1: Splitting Criterion on Feature X (Source: Taken directly from Dubov, Evgeni. ”Classifying Imbalanced Data Using Hellinger Distance.”Mediu. 26 Mar. 2019. Web)

Table D.4: Splitting Criterion Performance (Source: Taken directly from Dubov, Evgeni. ”Classifying Imbalanced Data Using Hellinger Distance.”Mediu. 26 Mar. 2019. Web)

|  | Gini drop | Entropy drop | Hellinger Distance |
|---|---|---|---|
| Example score | 0.009 | 0.053 | 1.132 |
| Perfect score | 0.5 | 1 | 1.414 |
| Percent | 1.8% | 5.3% | 80.1% |