MACHINE LEARNING-BASED PREDICTION OF UNDER-FIVE MORTALITY USING HEALTH,SOCIO-DEMOGRAPHIC,AND CLIMATE DATA

M.Sc. THESIS

FEYISA ABEBE

APRIL 2025
HARAMAYA UNIVERSITY, HARAMAYA

Machine Learning-Based Prediction of Under-Five Mortality Using Health, Socio-Demographic, and Climate Data.

A Thesis Submitted to the Department of Computer Science,
Post Graduate Program Directorate
HARAMAYA UNIVERSITY

In Partial Fulfillment of the Requirements for the Degree of
MASTER OF SCIENCE IN COMPUTER SCIENCE

Feyisa Abebe

April 2025
Haramaya University, Haramaya

POST GRADUATE PROGRAM DIRECTORATE

HARAMAYA UNIVERSITY

We hereby certify that we have read and evaluated the thesis entitled with **"Machine Learning-Based Prediction of Under-Five Mortality Using Health, Socio-Demographic, and Climate Data."** prepared under our guidance by **Feyisa Abebe**. We recommend that it be submitted as fulfilling the thesis requirement.

Abebe Belay Adege (PhD)		
Major Advisor	Signatures	Date
Mr. Tadesse Kebede (MSc)		
Co- Advisor	Signatures	Date

As a member of the Board of Examiners of the MSc Thesis Open Defense Examination, we certify that we have read and evaluated the Thesis prepared by **Feyisa Abebe** and examined the candidate, I recommend that the thesis be accepted as fulfilling the Thesis requirements for the degree of Master of Science in Computer Science.

Chair Person	Signatures	Date
Internal Examiner	Signatures	Date
Dr. Million Meshesha	million	April 15, 2025
External Examiner	Signatures	Date

Final approval and acceptance of the Thesis is contingent upon the submission of its final copy to the Council of Post Graduate Directorate (CPGD) through the candidate`s department or school graduate committee (DGC or SGC).

DEDICATION

To my beloved family and country!

STATEMENT OF THE AUTHOR

By my signature below, I declare and affirm that this Thesis is my own work. I have followed all ethical and technical principles of scholarship in the preparation, data collection, data analysis and compilation of this Thesis. Any scholarly matter that is included in the Thesis has been given recognition through citation.

This Thesis is submitted in partial fulfilment of the requirements for a Master degree at the Haramaya University. The Thesis is deposited in the Haramaya University Library and is made available to borrowers under the rules of the Library. I solemnly declare that this Thesis has not been submitted to any other institution anywhere for the award of any academic degree, diploma of certificate.

Brief quotation from this thesis may be made without special permission provided that accurate

and complete acknowledgment of the source is made. Request for permission for extended quotations from or reproduction of this Thesis in whole or in part may be granted by the Head of the School or Department when in his or her judgment the proposed use of the material is in the interest of scholarship. In all other instances, however, permission must be obtained from the author of the Thesis.

Name:_____

Signature:_____

Date:_____

School/Department:_____

BIOGRAPHICAL SKETCH

Feyisa Abebe is an academic staff member of Haramaya University, and he is currently working as a senior database administrator at Hararghe Health Research (HHR), a partnership program between Haramaya University and the London School of Hygiene and Tropical Medicine. He was born February 19, 1994, in Meki town, East Shoa, Oromia, Ethiopia. He followed his elementary, high school, and preparatory education in Meki town at Oda Bokota elementary, high school, and preparatory school, respectively. Feyisa graduated from Haramaya University in July 2017 with a degree in Bachelor of Science in software engineering. He worked as a software developer for more than one year and also worked as an assistant lecturer in the software engineering department at Haramaya University. He commenced his master's degree in computer science in a regular program at the Haramaya University Post Graduate Program. Feyisa is expected to present his academic research work for a master's in Computer Science in February 2025.

ACKNOWLEDGMENTS

First of all, I would like to express my deepest gratitude to the Almighty GOD for his guidance and blessing. Next, I would like to express the heartfelt thanks to my advisors, Dr. Abebe Belay and Mr. Tadesse Kebede. The investigation of the research direction, the working-out of the research process and the achievements of the paper have all benefited from their constructive comments, invaluable guidance, strong and unwavering support. In scientific research or in life, their invaluable guidance, advice and suggestions can always make me enlightened. I honor both of you sincerely.

I am also profoundly grateful to the Ethiopian National Meteorology Agency and Hararghe Health Demographic Surveillance system for their cooperation and willingness to share the data I requested for this study.

Last but not least, thanks to my beloved family particularly my wife Beshadu, whose good care and endless support is the driving force for my successful completion of the master study.

LIST OF ACRONYMS AND ABBREVIATIONS

AUC	Area Under Curve
AUC-ROC	Area Under the Receiver Operating Characteristic Curve
CNN	Convolutional Neural Network
DL	Deep Learning
DNN	Deep Neural Network
DT	Decision Tree
EPHI	Ethiopia Public Health Institute
EDHS	Ethiopian Demographic Health survey
FPR	False Positive Rate
HDSS	Health Demographic Surveillance System
KNN	K-Nearest Neighbor
LR	Logistic Regression
MAR	Missing At Random
MCAR	Missing Completely At Random
MDG	Millennium Development Goal
ML	Machine Learning
MNAR	Missing Not At Random
NB	Naïve Bayes
NFHS	National Family Health Survey
RF	Random Forest
SDG	Sustainable Development Goal
SMOTE	Synthetic Minority Oversampling technique
SVM	Support Vector Machine
TPR	True Positive Rate
U5M	Under-Five Mortality
WHO	World Health Organization

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

Health is a state of full well-being and a cornerstone of international development, with considerable investments made over the last three decades to reduce morbidity and mortality. Under-five mortality, which is generally defined as the death of children under the age of five, is still a critical public health challenge in developing countries. The study here proposes a machine learning-based approaches for predicting under-five mortality using health, socio-demographic, and climate data from Eastern Hararghe, Ethiopia.

The data used in this study were collected from the Hararghe Health Demographic Surveillance System and Ethiopian National Meteorology Agency. The following eight supervised machine learning algorithms were considered: Naïve Bayes (NB), Support Vector Machine (SVM), K-Nearest Neighbors (KNN), Decision Tree (DT), Random Forest (RF), eXtreme Gradient Boosting (XGBoost), Attentive Tabular Network (TabNet), and Convolutional Neural Network (CNN). The proposed framework covers data preprocessing, exploratory data analysis, model training, prediction, performance evaluation, and identification of the key determinants of under-five mortality. It was observed that an 80:20 split produced an optimum performance in the models. Preprocessing techniques were then applied to enhance data quality before training the machine learning models. There were two experimental setups: one with a data-balancing technique and the other without.

Results indicated that balanced datasets always outperformed. Amongst all the models developed, the XGBoost recorded the highest accuracy, having testing accuracy score of 97.9%, precision of 98%, F1-score of 98%, and recall of 98%. The determinants of under-five mortality identified in this study were antenatal care, child gender, wealth index, total number of alive children, preceding child alive, physical healthy ,birth place and weight of baby. In the end, the XGBoost algorithm emerged as the best among other models, proving to be the most reliable predictive model for under-five mortality. This study has shown the potential of machine learning providing helps in tackling critical public health challenges by leveraging diverse datasets to enhance decision-making and interventions. Household-level climate data were not utilized in this thesis, which would be taken into account by future researchers.

Keywords: Under-five mortality, Machine learning, Health care system.

CHAPTER 1

1. INTRODUCTION

1.1. Background

Health is a state of complete well-being, including our physical, mental, and social states (WHO, 2024). Maintaining good health is principal, as a healthy individual leads to a healthy community, where individuals are more likely to contribute positively and create a supportive environment for all (Health, 2024). In almost the last three decades, health has been identified as one of crucial element to international development and significant investments has been made to mitigate either universally or specific population morbidity and mortality, with a focus on vulnerable populations like the poor, women, and children (Buse & Sarah, 2015). The Sustainable Development Goals (SDG) encompass a comprehensive framework for achieving human, social, and environmental development objectives, including 17 goals with 169 targets and 232 unique indicators (D.Moyer & Steve, 2020). Three specific goals targeting health outcomes by 2015, such as reducing child mortality (goal 4), improving maternal health (goal 5), and combating the spread of HIV/AIDS, tuberculosis, and malaria (goal 6), are included in the Millennium Development Goals (MDGs) established in 2000 targeting to improve global health. With targets that include ceasing the epidemics of AIDS, tuberculosis, malaria, and neglected tropical diseases and combating hepatitis, water-borne diseases, and other communicable diseases; reducing maternal and children's mortality; and achieving universal health coverage, SDG 3 (healthcare as good health and well-being) is declared as one of the seventeen basic development goals in SDG to ensure healthy lives and promote well-being for all at all ages (WHO ,2017).

Under-five mortality is the death of children under the age five years old and is one of the major public health problems in developing country. Under-five mortality is an important indicator of child health and overall development of a nation, as it reveals the social, economic, and environmental conditions in which children (and others in society) reside, including their healthcare (Gebretsadik & Emmanuel, 2016). Under-five mortality rate refers to the probability that a newborn would die before reaching exactly 5 years of age, expressed per 1,000 live births. The under-five mortality rate is one of the significant measures of global health which is declared as a major development goal and one of the

key targets of the SDG 3 target 2. In SDG 3.2, call on all countries to reduce under-five mortality to at least as low as 25 per 1,000 live births (WHO ,2017). So achieving SDG 3 target of under-fivemortality rate to 25 per 1000 live births by 2030 needs understanding and identifying of the determinants of under-five mortality for the implementation of proper intervention.

Globally, 4.9 million children under the age of 5 years died in 2022. The under-five mortality rate fell to 37 deaths per 1000 live births in 2022. In the world, under five mortality rate decreased from 93 deaths per 1000 live births in 1990 to 37 deaths per 1000 live births in 2022. Despite the significant improvements made over the past decades, children in sub-Saharan Africa countries continued to have the highest mortality rate in the world, 71 deaths per 1,000 live births, which is 1 in 14 children in sub-Saharan Africa before reaching their fifth birthday in 2022. Which is 15 times higher than the risk of children born in high-income countries and 20 years behind the world average, which achieved a 1 in 14 rate by 2001 (UNICEF, 2024). Ethiopia happens to have the fifth highest deaths of children under the age of five years' in the globe, following Nigeria, India, Pakistan and the Democratic Republic of the Congo and third highest deaths of under-five mortality in Africa (UNICEF, 2024). Ethiopia is one of the countries that achieved the target of millennium development goal (MDG) by reducing Under-five mortality rate by two third, from 204 deaths per 1000 live births in 1990 to58 deaths per 1000 live births in 2016 (Fikrewold, Samuel, Lloyd, & Corey, 2020) . Recently, in 2019, the under-five mortality rate was reported as 55 deaths per 1,000 live births, according to the Ethiopian Mini DHS (Ethiopian Public Health Institute, Federal Ministry of Health, & Icf, 2021) . However, Ethiopia is still far from meeting SDG 3's second target, which is to reduce under-five mortality to at least 25 deaths per 1,000 live births (Ethiopian Public Health Institute, Federal Ministry of Health, & Icf, 2021) . The result in minimizing the under-five mortality is not similar between and within the regions, like under-five mortality in Ethiopia in 2019 were varied from 29 in Addis Ababa, to 74 per 1000 live births in Afar Region (United Nations Inter-Agency Group for Child Mortal, 2020) . The study by (Dheresa, et al., 2022) showed that under-five mortality rate in Kersa is increasing annually from 2015 to 2020 having the rate of 27.9 per 1000 live births in 2015 to 54.7 per 1000 live births and the overall rate of 46.3 per 1000 live births.

According to (Fikrewold, Samuel, Lloyd, & Corey, 2020) the risk factors for the under-five mortality are household size, time to the source of water, breastfeeding status, number of births in the preceding 5 years, sex of a child, birth intervals, antenatal care, birth order, type of water source, and mother's body mass index

Climate change is greatly affecting the health and lives of children around the world. Ethiopia is extremely sensitive to the effects of climate change, including increases in average temperature and variations in precipitation, especially in the water, agricultural, infrastructure, forestry, and public health organizations (Belay, et al., 2016).

Now a day massive data is being generated and collected due to the development of advanced technology. Thus, effectively understanding, interpreting and developing insight from the massive and comprehensive data requires machine learning techniques which can effectively change data into meaningful information. In recent year, machine learning methods have demonstrated encouraging outcomes when it comes to predicting under-five mortality. Machine learning models can automatically identify interactions and find the linear and nonlinear relationship between the dependent variable and the independent variables, which is not apparent to traditional linear model (Carlos, et al., 2023).

In this study, supervised machine learning approaches were used to develop scalable and accurate prediction model for identifying determinant factors of under-five mortality by integrating data from two different sources, such as climate data and health, socio-demographic data of children younger than five years.

1.2. Statement of the Problem

Under-five mortality is one of the potential health problems and the most commonly used indicator to measure the health status of the children. It is the death of a child younger than five years old. The under-five mortality rate has decreased in Ethiopia in the past decades, from 166 deaths per 1,000 live births in 2000 to 55 deaths per 1,000 live births in 2019, according to the Ethiopia Demographic and Health Survey (EDHS). However, the country is still a long way to achieve SDG 3.2 target, which is to decrease under-five mortality rate to at least 25 deaths per 1,000 live births. This leads to a practical challenge for policymakers, planners, and health care providers working in healthcare disease prevention and control activities.

Under-five mortality is a complex problem with multiple contributing factors, including socioeconomic factors, demographic factors like place of residence, and health-related factors such as lack of access to healthcare and infectious diseases. Traditional statistical methods have been used to identify and analyze these factors, but they have limitations in terms of their ability to handle large, complex datasets and identify nonlinear relationships between variables (Khosravi, et al., 2023). For evidence-based decision-making, especially in resource-limited settings, the Health Demographic Surveillance System (HDSS) occurred as the best source of representative data, and unlike cross-sectional studies or Demographic and Health Surveys, HDSS provides longitudinal data on a well-defined and stable population (Dheresa, et al., 2022). Thus, it is important to use comprehensive and representative data on these factors to develop a scalable and accurate predictive machine learning model. Exploring the determinants of under-five mortality in order to reduce the vulnerability of a child's survival is essential. Additionally, to design effective intervention measures, decision-makers must be aware of the most important predictors of under-five mortality, utilizing a suitable analytical framework and representative data. So machine learning algorithms are one of the powerful tools that can be used to address the potential factors and challenges of under-five mortality, as they can learn from large datasets and identify complex patterns that may not be apparent to traditional statistical methods (Morera, Juan, José, Jingjing, & Sergio, 2021).

The research work conducted by most of the researchers have employed machine learning approaches to identify the determinants that best contribute to the deaths of under-five children, using socio-demographic cross-sectional survey data (Adegbosin, B, & J, 2019; Carlos, et al., 2023; Fikrewold, Samuel, Lloyd, & Corey, 2020; Saroj, Yadav, Rajneesh, & Chilyabanyama, 2022; Solomon, Angela, Oluwafemi, Christabel, & Ignace, 2023). Even though the researchers employed machine learning algorithms to identify determinants of under-five mortality, they were limited to only socio-demographic factors. But today climate change is the biggest global health threat of 21 centuries (Hanna & Paulina, 2016). Children are more vulnerable than adults to environmental climate change factors (Dr.Frederica & Dr.Kari, 2022) . Because of their compromised thermoregulatory function at extreme temperatures, children are more vulnerable than adults to severe heat. Climate change can have a major risk to the health and wellbeing of children. There are two ways of effects of climate change on human health, direct effects of climate change include temperature changes

(heatwaves and more rapidly changing temperatures), changing precipitation patterns with increased risk of floods, droughts, and wildfires. And indirect effects include ecosystem disruption, changing vector patterns, air pollution, and aeroallergens (D. H., et al., 2021) . Some of the way that climate change is affecting the children include: maximized exposure to extreme weather events, increased risk of infection disease, malnutrition and mental health problem. Thus, integrating health, socio-demographic data from the Hararghe Health Demographic Surveillance system, and climate data from National Meteorology Agency (NMA), and identifying determinants of under-five mortality using machine learning approaches is the main interest of this research work. Then compare the performance of machine learning algorithms in predicting under-five mortality. As far as the researcher's knowledge is concerned, no studies have been conducted to predict under-five mortality by integrating health, socio-demographic data and climate data using machine learning algorithms. At the end of this study, the research aims to answer the following research questions (RQ) from the obtained experimental result.

RQ1. What health, socio-demographic, and climate factors are the significant determinants of under-five mortality?

RQ2. Which machine learning algorithm is most effective and appropriate for predicting under-five mortality?

RQ3. To what extent can the proposed machine learning model accurately predict under-five mortality?

1.3. Objective

1.3.1. General Objective

The general objective of this study is to develop a predictive model for child under-five mortality using machine learning algorithms on health, socio-demographic, and climate data.

1.3.2. Specific Objective

To achieve the general objective of the study, the following specific objectives should be performed.

- To integrate health, socio-demographic data, and climate data of three sites (Kersa, Harar and Haramaya).

- To develop a scalable and accurate predictive machine learning model for under-five mortality prediction.
- To evaluate the effectiveness of machine learning techniques in predicting child under-five mortality using metrics such as precision, recall, accuracy, and F-measure.
- To compare the performance of machine learning algorithms in predicting under-five mortality.
- To determine health, socio-demographic, and climate variables that are a significant risk factors for child under-five mortality in eastern Hararghe, Ethiopia.

1.4. Scope and Limitation of the Study

In this research work, we mainly focus on developing a predictive model for child under-five mortality on health, socio-demographic, and climate data using machine learning algorithms. This study encompasses the entire process, from collecting and preparing the dataset to preprocessing and ultimately making prediction and identification of factors influencing under-five mortality. Different machine learning algorithms are applied to the dataset for prediction and determinants identification purposes. These algorithms are Naïve Bayes, K-nearest neighbor, support vector machine, decision tree, random forest, extreme gradient boosting, Tabnet, and convolutional neural network. The climate data used is at the site level in eastern Hararghe (namely Kersa, Harari, and Haramaya), so we did not use the household level climate data on this research.

1.5. Significance of the Study

This research has made efforts to provide the following significance:

- This study contributes to the mitigation of under-five mortality in order to achieve the SDG target by 2030, which is to reduce the under-five mortality rate to at least 25 deaths per 1,000 lives.
- This study provides guidance in the planning, implementing, monitoring, and evaluating of child health programs in the eastern Hararghe, Ethiopia.
- To support researchers who are motivated in developing the model with high performance and interpretability on other health-related problems.

- ❖ Government and policymakers are advantageous of the predictive model to identify effective interventions for the core causes of child mortality depending on the identified factors.
- ❖ This study made a simple way of identification of the impacts of climate change on child health as a result of integrated data which is health, socio-demographic, and climate data.

1.6. Thesis Organization

This research paper is organized into six chapters. Chapter 1 presents an introduction, statement of the problem, objectives, significance, scope and limitation of the study. Chapter 2 discusses an overview of under-five mortality and trends and impacts of climate change on child health and the different machine learning approaches used in under-five mortality prediction. And related works of the research are also discussed in this chapter. Chapter 3 presents the employed research methodology. Chapter 4 discusses the general architecture and design of the under-five mortality prediction model. Chapter 5 shows experiment conducted and performance evaluation of the model with analysis of results. Chapter 6 presents conclusions, contributions and recommendations of the study.

CHAPTER 2
2. LITERATURE REVIEW

2.1. Overview

In this chapter, an overview of under-five mortality and the trends of under-five mortality has reviewed. Numerous studies carried out on under-five mortality are reviewed to identify the strengths, limitations, gaps, and findings of the research. And to have deep insight into the concepts, techniques, methodology and supervised machine learning approaches utilized in under-five mortality are discussed. Other health-related work was also reviewed.

2.2. Overview of under-five mortality and the trends

Under-five mortality rate is the probability of death in children younger than five years, expressed per 1,000 live births (Saroj, Yadav, Rajneesh, & Chilyabanyama, 2022). Under-five mortality is one of the leading indicators and significant measures of the health status of children. And also, it is one of the measures of the general development of the countries. Sustainable development goal 3.2 targets all countries to reduce the under-five mortality rate to at least 25 deaths per 1000 live births. The world has made significant progress in reducing deaths of children younger than five years in the last three decades (UNICEF, Under-five mortality, 2024) . In 2022, the likelihood of children surviving before exactly reaching the age of five has significantly improved compared to 1990, with a decrease in the mortality rate from 1 in 11 children to 1 in 27. This indicates that millions of children now have a higher chance of survival. Globally, 4.9 million children under the age of five died in 2022. Worldwide, the under-five mortality rate declined from 93 deaths per 1,000 live births in 1993 to 37 deaths per 1,000 live births in 2022. Even though child mortality is unequally distributed around the world, the majority of countries globally have reduced under-five mortality rates by at least half since 1990. This demonstrates that enhancing child survival is achievable even in settings with limited resources. Despite the significant improvements made over the past three decades, sub-Saharan Africa countries remain the region with the highest under-five mortality rate globally, 71 deaths per 1,000 live births. Which is 1 in 14 children in sub-Saharan Africa before reaching their fifth birthday in 2021. Which is 15 times higher than the risk of children born in high-income countries and 20

years behind the world average, which achieved a 1 in 14 rate by 2001 (UNICEF, 2024). The risk of a child dying before the age of five in the country with the highest mortality rate is approximately 80 times greater than in the country with the lowest mortality rate. Sub-Saharan Africa and South Asia are the two regions where more than 80 percent of deaths of children younger than the age of five occurred out of all under-five deaths occurred in the world in 2019. Ethiopia is home to highest death of under-five mortality in the world following Nigeria, India, Pakistan, and the Democratic Republic of the Congo (WHO, 2020) . Ethiopia is one of the countries that achieved the target of millennium development goal (MDG) by reducing Under-five mortality rate by two third, from 204 deaths per 1000 live births in 1990 to 58 deaths per 1000 live births in 2016 (Fikrewold, Samuel, Lloyd, & Corey, 2020). Recently, in 2019, the under-five mortality rate was reported as 55 deaths per 1000 live births, according to the Ethiopian Mini DHS (Ethiopian Public Health Institute, Federal Ministry of Health, Icf,2021). However, Ethiopia is still far from meeting SDG 3's second target, which is to reduce under-five mortality to at least 25 deaths per 1,000 live births (Ethiopian Public Health Institute E, Federal Ministry of Health F, Icf,2021). The result in minimizing the under-five mortality is not similar between and within the regions, like under-five mortality in Ethiopia in 2019 were varied from 29 in Addis Ababa, to 74 per 1000 live births in Afar Region (United Nations Inter-Agency Group for Child Mortality Estimation, 2020). The study by (Dheresa, et al., 2022) showed that under-five mortality rate in Kersa is increasing annually from 2015 to 2020 having the rate of 27.9 per 1000 live births in 2015 to 54.7 per 1000 live births and the overall rate of 46.3 per 1000 live births. So, achieving Sustainable Development Goal 3 target two requires an understanding of the trends and determinants that best contribute to the deaths of children younger than the age of five to design and implement appropriate interventions.

2.3. Impacts of climate on child health

Climate change remains an urgent and continuous global issue that requires immediate attention (Emmanuelle, et al., 2021) . Climate change is identified as one of the most significant global health risks of the 21st century. Global warming arises from the interactions among greenhouse gases, the earth's atmosphere, and the sun. The impacts of global warming are widely recognized as a major risk to global health and overall well-being, with children being identified as susceptible to its consequences. According to (WHO, 2023) around 3.6 people are living in areas that are highly vulnerable to climate change. Low-income countries and small island developing states (SIDS) face the most severe health impacts, even though they make minimal contributions to global emissions. The death rate in vulnerable region from extreme weather events in the past decade was 15 greater than in less vulnerable ones. Elderly, children, individuals with underlying health conditions and populations in LMICs are the populations that are vulnerable to climate changes. Climate change could increase the vulnerabilities of children and other vulnerable subpopulations, that could significantly risk the future progress and potentially alter the improvements made in child survivals and well-being in the recent years (Helldén, et al., 2021) . Focusing on children, children are considered as susceptible subpopulation due to their developing physiology and the potential for long-term exposure. The range of childhood conditions worsened by the direct and indirect impacts of climate change include illnesses transmitted by vectors, water, and food-born infectious disease, as well as mental health issues (Emmanuelle, et al., 2021) . Climate change is anticipated to have adverse impacts on human health in Africa in many ways by worsening malnutrition, causing air pollution, contaminating drinking water, changing the distribution of disease-causing pathogens and carriers, and leading to large-scale human displacement (Sarah, Wendemagegn, Mark D. P. D, & Louise, 2020).

Ethiopia is among the countries vulnerable to climate change, with an increasing number of reports of massive losses of human lives because of flooding, domestic water supply shortages, malnutrition, and the expansion of malaria transmission (Andualem, Benedict, & Helmut, 2014) . Flooding, a climate-related threat, impacts human health in different ways, including causing mortality, injuries, waterborne diseases, malnutrition, and mental

health issues. Different parts Ethiopia reported massive losses of human lives and damage of properties in 1988, 1993, 1994, 1995, 1996 and 2006. Thus, flooding took 256 human lives in Dire Dawa Town and 364 lives in South Omo Zone in 2006. And the properties that worth millions of dollars were damaged due to the flood in 2006 in Dire Dawa. Climate-related impact, such as malnutrition, mainly hamper the growth of children. In Ethiopia, the distribution of malnutrition among children has been linked to drought occurrences. For instance, the average annual temperature has risen by 1.3°C from 1960 to 2006, estimated to an average rate of temperature increase of 0.28°C per decade (Simane, et al., 2016). Thus, children are the least to blame for the variability of climate change and highly vulnerable to the impacts of climate change. The adverse impacts of climate change on child health are high, particularly in Ethiopia, where agriculture is highly dependent on rain-fed agriculture.

2.4. Data Preparation

Before proceeding into data analysis, data must be organized into an appropriate format to fit and evaluate machine learning models. Data preparation is a process of manipulating and organizing data before proceeding with data analysis. It is an iterative process of manipulating raw data, which perhaps converting unstructured and messy data into more structured, actionable and useful data that allows for further analysis using machine learning models (Zahraa, Lan, Geoffrey,2017). For predictive models such as classification and regression, raw data cannot be directly utilized because machine learning expects data to be in specific format specifically, the algorithms require data to be numbers. So raw data must be pre-processed before being used to fit and evaluate machine learning model. Data preparation process is one of the most difficult activities in any machine learning tasks. The data preparation process consists of a series of main activities including data cleaning, feature selection, data transformation and feature engineering.

2.4.1. Data Cleaning

Data cleaning is the process of identifying and correcting (removing) missing values, outliers, inconsistencies and errors in the data. Data cleaning is very significant steps and time consuming tasks in any machine learning projects (Abidin, Siti, Malik, Erwin, Errissya , Yovi, 2020). It involves in processing the data that contains the noise data, missing data and duplication of data.

2.4.2. Handling Missing Value

Handling missing value in data is one of the major activities in the data cleaning process. Though it is vital to check or determine whether missing values are there in the datasets, if exists we need to ensure that the appropriate measure has been taken to allow the learning system to handle it. Most of the existing datasets contains missing values because it was not introduced or not recorded during the data collection process or data encoding process. Categories of missing values in datasets are presented in the following table (Tamboli, 2023).

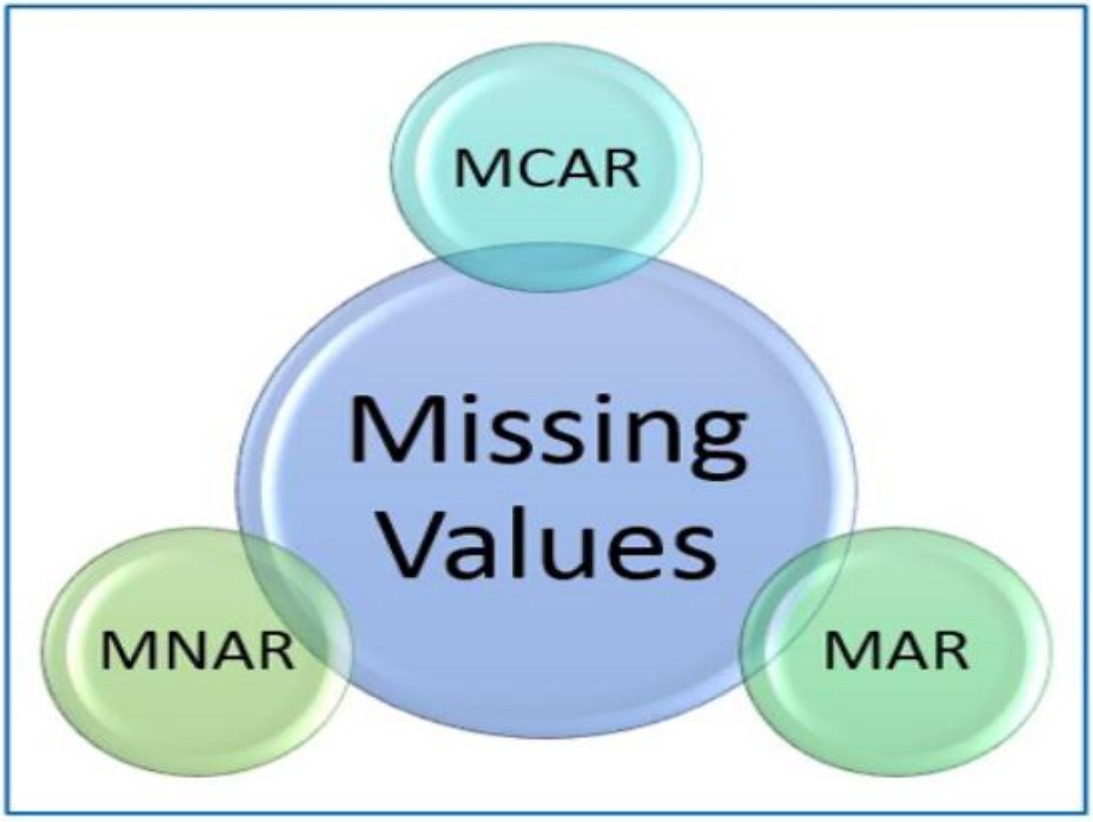


Figure 2- 1 Categories of missing values in datasets (Tamboli, 2023)

Missing Completely At Random (MCAR)

Missing completely at random is the probability of data being missing is equal for all observations. This means that there is no connection between the missing data and any other values that are observed or unobserved in the dataset. In other words, the missing values are completely unrelated to the rest of the data, and there is no discernible pattern (Tamboli, 2023). In instances where data follows the MCAR pattern, missing values could be a result of various factors such as human error, system or equipment failures, loss of samples, or unsatisfactory

technical issues during the recording process.

Missing At Random(MAR)

MAR data refers to situations where the missing values can be accounted for by variables for which complete information is available. This suggests that there is a connection between the missing data and other values or variables. Unlike in MCAR, the data is not missing across all observations. Instead, it is missing only within specific subsets of the data, and there is a discernible pattern in the missing values (Tamboli, 2023).

Missing Not At Random (MNAR)

If the missing values in a dataset exhibit a pattern that cannot be explained by the observed data, it is considered Missing Not At Random (MNAR). This occurs when there is a relationship between the missing data and unobserved variables. For example, in a survey, certain respondents may choose not to answer specific questions, leading to missing values that are not random (Tamboli, 2023).

Generally Handling missing values in datasets is very crucial because most of the machine learning algorithms fail to perform if the datasets contain missing values. The machine learning model we build would be biased because of not handling missing values in datasets. And it end up with incorrect results because not appropriate measure has been taken for missing values in datasets. There are various techniques that can be used to handle missing values in datasets, the most commonly used techniques for handling missing features or values typically involve imputation. The fundamental concept behind imputation is that if a significant feature or value is missing for a specific instance, it can be estimated from the available data. Imputation techniques can be categorized in to two: statistical (mean/mode/median, linear regression and least square) and machine learning based techniques (DT, KNN, RF) (Wei-Chao & Chih-Fong, 2019). In this study we used iterative imputation approaches. Iterative imputation is an approach where each feature is modeled as a function of the other features, e.g. a regression problem where missing values are predicted. Each feature is imputed sequentially, one after the other, allowing prior imputed values to be used as part of a model in predicting subsequent features. It is iterative because this process is

repeated multiple times, allowing ever improved estimates of missing values to be calculated as missing values across all features are estimated.

2.4.3. Data Integration

Data integration is the process of merging the data from two or more sources, one socio-demographic data from Hararghe Health Demographic Surveillance System and climate data from three sites. Provide the models with unified dataset, thus we can evaluate the effects of climate change for the child younger than year of five. To integrate data from health, socio-demographic, and climate data, we have common variable called site and based on this common variable we would join the two data. So to join the data from two sources we have used two possible options, one we can join using SQL query language and get a unified comprehensive data. Second, we used merge methods in Python to join data from two sources based on the common variables and get unified data.

We used python function called merge to join data from two sources on common variables, so first we need to read each data frame using read_csv function, then we proceed with joining data based on the common variables.

```
Dataframe1= pd.read_csv("sociodemgraphic.csv")
Dataframe2=pd.read_csv("climate.csv")
Unified_data=Dataframe1.merge(Dataframe2,left_on='site',right_on='site')
```

Table 2- 1 Joining two datasets

2.4.4. Feature Selection

Feature selection is a process of selecting a subset of input features that are most important to dependent variable that is being predicted. It is critical as irrelevant and duplicate input variables can mislead learning algorithms, possibly end up in lower predictive performance.

Feature selection, also known as a process of removal of irrelevant variables and focus on relevant variables, serves several important purposes in machine learning. It helps in gaining a better understanding of the data, as it allows for the identification of the most relevant and informative features. By selecting a subset of features, it also reduces computational requirements and mitigates the impact of the curse of dimensionality, which refers to the challenges that arise when working with high-dimensional data. Additionally, feature selection contributes to improving the performance of predictors by focusing on the most influential features, leading to more accurate and efficient models (Girish & Ferat, 2014).

Feature selection techniques can be categorized as supervised or unsupervised. Supervised

techniques can be further divided into embedded methods, which automatically select features during model fitting, wrapper methods, which incorporate feature selection within the learning algorithm itself. The algorithm automatically selects the most relevant features during the model training process. It explicitly choose features for optimal model performance. And filter methods, which score each input feature independently and allow for the selection of a subset of features (Pradip & Chandrashekhar, 2021). The overview of feature selection is presented in the below figure.

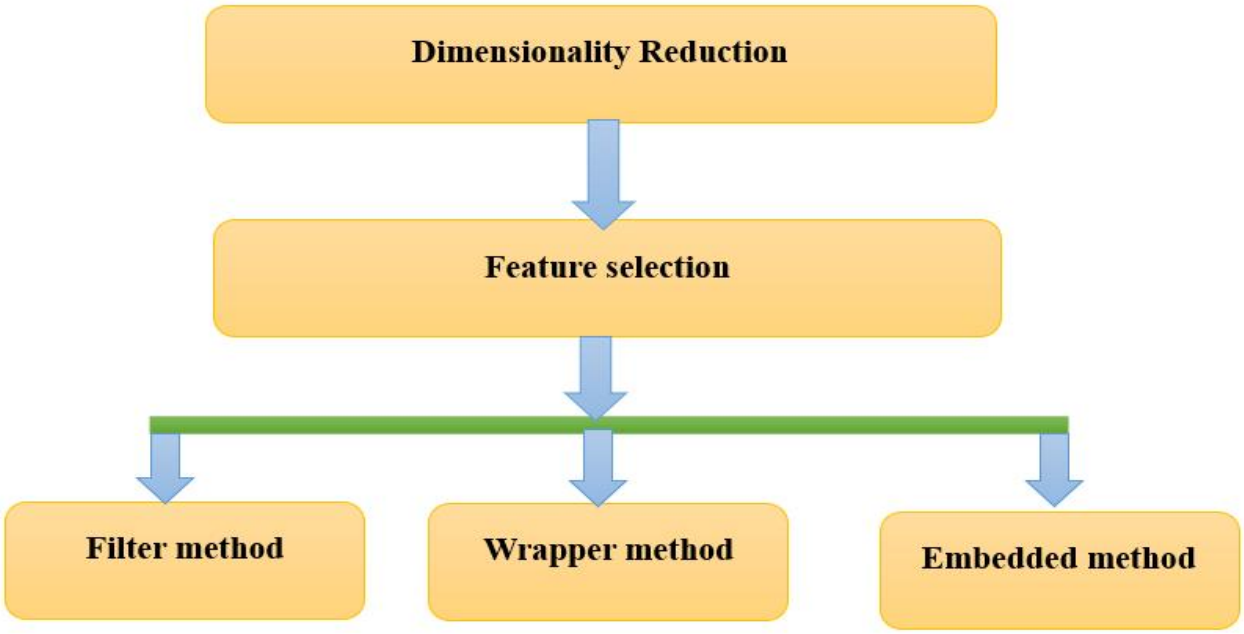


Figure 2- 2 Feature Selection

In this research work, we used the feature selection techniques like recursive feature elimination from wrapper method, random forest importance, TabNet feature importance and XGBoost feature importance from embedded method for predicting the most relevant features or variables of under-five mortality.

Recursive feature elimination (RFE): is an iterative procedure that involves removing one feature at a time and reevaluating the importance of the features after each elimination. It is popular type of wrapper methods. RFE aims to enhance the generalization performance of a model by eliminating the least important features that have minimal impact on training errors *(Puneet & Yadav, 2020)*.

Embedded feature selection: is a technique in machine learning where feature selection is integrated into the model training process. Rather than performing feature selection as a separate step before or after training, embedded feature selection methods incorporate feature selection directly into the learning algorithm. The model itself determines the importance or relevance of features during the training process. Random forest and decision tree are well known embedded feature selection methods (Chih-Wen, Yi-Hong , Fang-Rong , & Wei-Chao, 2020).

2.4.5. Handling Imbalanced Data

When the distribution of classes in a dataset is unequal, it is referred to as technical imbalance. However, if there is a substantial or extreme disproportion in the number of examples for each class in the problem, the dataset is considered imbalanced. Imbalanced data classes are a common occurrence in many real-life scenarios, such as mortality data, where there is a significant disparity between the number of survivors and the number of mortality cases. In the case of imbalanced datasets where one class significantly outnumbers the other, machine learning algorithms, including random forests, can struggle to accurately classify the data. These algorithms are sensitive to the proportions of different classes and tend to exhibit biased behavior by favoring the majority class while treating the minority class with less emphasis (Waititu1, Koskei1 , & Onyango, 2020) .

This imbalance in the dataset leads to a higher rate of misclassification in the minority class samples. As a result, the predictive accuracy of the minority class is weakened, while the majority class tends to have artificially high predictive accuracies due to correct classification. Thus, to overcome this problem, it is very significant to use balanced classification for machine learning algorithms.

To provide a solution for the problem related to imbalance class, different techniques have been suggested. Techniques like data level (external), algorithm level (internal) techniques, cost-sensitive learning techniques and ensemble-based methods are suggested for imbalance class in dataset. In this research work we employe data level preprocessing techniques (Waititu1, Koskei1 , & Onyango, 2020).

Random under-sampling

The main objective of random under-sampling is to do balancing by randomly removing or eliminating the records from the majority class until when dataset is balanced. It helps to

create a more balanced dataset, allowing machine learning algorithms to better handle the data and improve overall classification performance. The major disadvantages of random under-sampling techniques is that there is high possibility of deleting or avoiding potentially useful data belongs to majority class, and it leads to loss of necessary information (Waititu1, Koskei1 , & Onyango, 2020).

Random over-sampling

In case of random under-sampling, we remove the records from the majority class up to when data balanced but, random over-sampling techniques generate new sample from minority class until dataset get balanced. Thus, to balance dataset using this technique the records or observations from the minority class is reduplicated. Here the main advantage of this techniques is there is no data loss while ensuring the balancing of the dataset, but it increase the observation(exact copy) from the minority class which leads to overfitting problem because the model train with same data and cannot give accurate prediction on the new(test) dataset. Additionally, over-sampling techniques may increase computational work and execution time when dealing with large imbalanced datasets (Waititu1, Koskei1 , & Onyango, 2020).

Synthetic Minority Oversampling technique (SMOTE)

It is a hybrid technique where random under-sampling and random over-sampling are combined aiming to overcome their challenges or disadvantages. The key idea in SMOTE is to produce new samples of the minority class artificially (Waititu1, Koskei1 , & Onyango, 2020). This helps to avoid overfitting brought about by reduplication of minority class instances. Additionally, the majority of class examples are under-sampled, giving rise to a more balanced dataset. SMOTE is one of the most preferred techniques in data balancing fields (Fernández, et al., 2018). So on this research work we employed SMOTE for data balancing because of the significant advantages it offer in order to overcome the drawbacks of both under-sampling and over-sampling approaches. Here the unique process or algorithm of SMOTE is as follows.

Step 1: Setting the minority class set **A**, for each $x \in A$, the k-nearest neighbors of **x** are obtained by calculating the Euclidean distance between x and every other sample in set **A**.

Step 2: The sampling rate N is set according to the imbalanced proportion. For each $x \in A$, N examples (i.e $x1$, $x2$, … xn) are randomly selected from its k-nearest neighbors, and they construct the set $\boldsymbol{A1}$

Step 3: For each example $xm \in A$ where (m=1,2,..N), then an artificially constructed minority sample is calculated by

$\mathbf{p_x} = x + rand(0,1) * (x - xm)$

where rand(0,1) is a random number uniformly distributed within the range of [0,1].

2.5. Machine Learning Approaches

Machine learning is the sub-field of artificial intelligence that offers a computer to learn without being explicitly programmed but learn from a given labeled example data or experience (Mahesh, 2018). Depending on the given problem and available data machine learning methods are categorized into supervised, unsupervised, and reinforcement methods.

2.5.1. Supervised Learning

Supervised learning is one of the most important methodologies in machine learning, where a model is trained on the labeled dataset, with the objective of making prediction on new or unseen dataset (Badillo, et al., 2020) . The model is given the input (features) and output (dependent variable i.e. 0 low risk and 1 high risk) pairs of labeled data and the algorithm learns to map input to the corresponding outputs. Finally, the model is tested on a new dataset called as testing set, to determine the performance of the model on unseen data.

2.5.1.1. Support Vector Machine (SVM)

SVM is one of the powerful supervised learning that works on smaller and complex datasets. It can be used both for regression and classification. However, mostly used in classification problems. New instances are predicted based on class and the side of the partition they fall in. The SVM is the nearest data point to the hyperplane that divides the classes (Saroj, Yadav, Rajneesh, & Chilyabanyama, 2022). In SVM there are two terminology that would be used.

Support Vectors: These are the points that are closest to the hyperplane. A separating line would be defined with the help of these data points.

Margin: it is the distance between the hyperplane and the observations closest to the hyperplane (support vectors). In SVM large margin is considered a good margin.

SVM does the classification tasks by building hyperplanes in a multidimensional space that separates cases of different class labels. Hyperplanes are used as decision boundaries that help classify the data points (Boateng, Joseph, & Daniel, 2020) . It is a technique for the classification of both liner and non- liner data. Additionally, it is also an algorithm that uses nonlinear mapping to transform the original training data into a higher dimension. The working principles of SVM are based on the concept of decision planes that define the decision boundaries. A decision plane is one that separates between a set of objects having different class memberships. An appropriate nonlinear mapping to a sufficiently high

dimension, data from two classes can always be separated by a hyper plane. Support vector machine have various characteristics such as ability to handle large feature space, ability to prevent of over fitting and information dense in a given data set. So, we can see that there exist multiple lines that offer a solution to the classification problem. But the problem here is that which of lines is better than the others. A line is not a good classifier if it passes too close to either of the points because it would be noise sensitive and not accurately classify all set of points. Therefore, the main objective should be finding optimal hyper plane which classify all set of points at optimal margins.

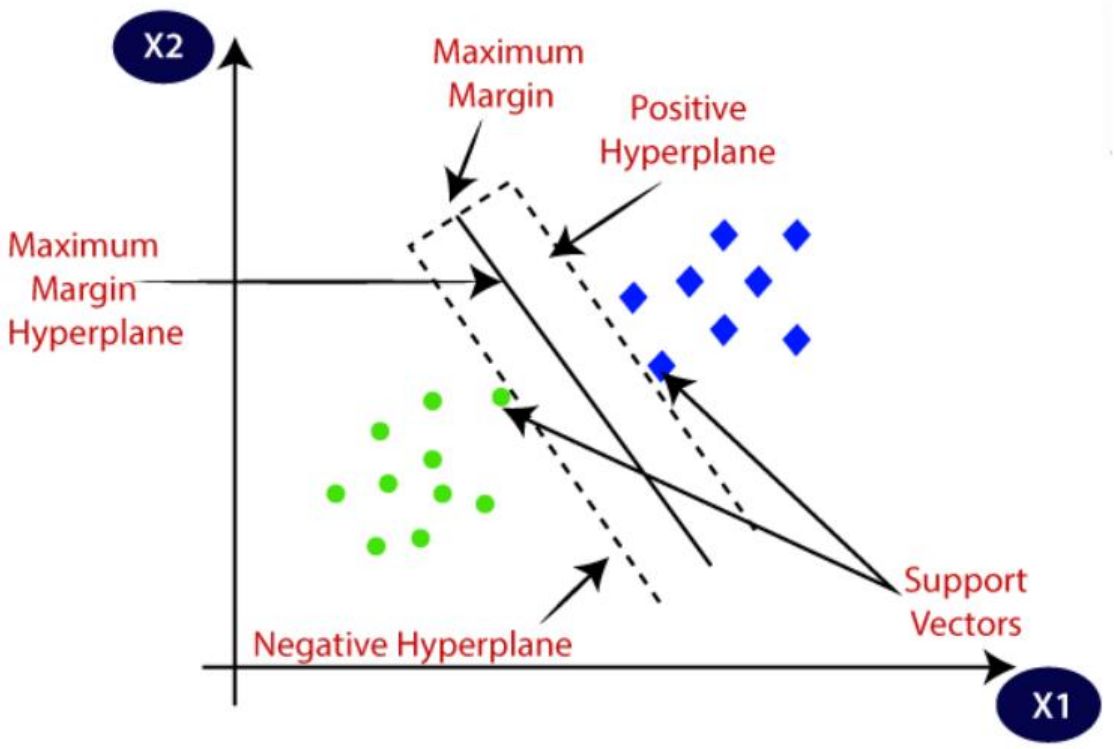


Figure 2- 3 Support Vector Machine(SVM) algorithm (Saroj, Yadav, Rajneesh, & Chilyabanyama, 2022)

2.5.1.2.Naïve Bayes (NB)

Naïve Bayes is a supervised learning algorithm, based on bayes theorem and it relies on the essential assumption that the characteristics are independent for the given class (Saroj, Yadav, Rajneesh, & Chilyabanyama, 2022) . Naïve Bayes Classifier is one of the simple and most effective Classification algorithms which helps in building fast machine learning models that can make quick predictions. The Naïve Bayes algorithm contain two words Naïve and Bayes,

which can be described as:

Naïve: It is called Naïve because it assumes that the occurrence of a certain feature is independent of the occurrence of other features.

Bayes: It is called Bayes because it depends on the principle of Bayes' Theorem.

Bayes' theorem is also known as Bayes' Rule or Bayes' law, which is used to determine the probability of a hypothesis with prior knowledge. It depends on the conditional probability.

$$P(A/B) = \frac{P(B/A)P(A)}{P(B)} \quad (1)$$

Where,

P(A|B) is Posterior probability: Probability of hypothesis A on the observed event B.

P(B|A) is Likelihood probability: Probability of the evidence given that the probability of a hypothesis is true.

P(A) is Prior Probability: Probability of hypothesis before observing the evidence.

P(B) is Marginal Probability: Probability of Evidence.

Naive Bayes assumes that all of the features in the dataset are equally important and independent. The Naive Bayes algorithm describes a simple method to apply Bayes' theorem to classification problems. The Naive Bayes algorithm was: Simple, fast, and effective and well with noisy and missing data.

2.5.1.3.K-Nearest Neighbors (KNN)

The k-Nearest Neighbors (KNN) technique is a non-parametric method for classification that is simple yet effective in various cases (Gongde, Hui, David, Yaxin, & Kieran, 2003). To classify a data point t, its k nearest neighbors are identified, creating a neighborhood around t. Typically, the majority voting among the data points in the neighborhood is used to determine the classification for t, with or without considering distance-based weighting. However, selecting an appropriate value for k is crucial when using KNN, as the classification success heavily relies on this choice. Essentially, the KNN approach is influenced by the value of k. There are multiple methods for determining the optimal k value, but a common approach is to execute the algorithm with different k values multiple times and select the one that yields the best performance.

The k-Nearest Neighbors (KNN) algorithm employs Euclidean distance metrics to locate the nearest neighbor (Boateng, Joseph, & Daniel, 2020). The Euclidean distance is the distance

between two points, it can be calculated as following equation.

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \quad (2)$$

2.5.1.4. Decision Tree (DT)

Decision Tree is a supervised machine learning approaches that are most commonly used for the problems of regression and classification (Harikumar & S R, 2019) . Being simple, interpreting, and visualize is one of the key properties of DT. Decision Tree has a hierarchical tree structure consisting of a root node, branches, internal nodes, and leaf nodes. It is a hierarchical model used in decision support that depicts decisions and their potential outcomes, incorporating chance events, resource expenses, and utility. This model utilizes conditional control statements and is non-parametric, supervised learning. Decision tree algorithm pseudocode are as follows:

- Keep the best feature of the input attributes at the root portion of the tree.
- Then make a splitting of training dataset into subsections.
- These divided subsets can be done by making each subset with data having the similar value for a input attribute.
- Then repeat the step 1, 2 and step 3 on each subset till the leaf portion in every branch of the tree is found.

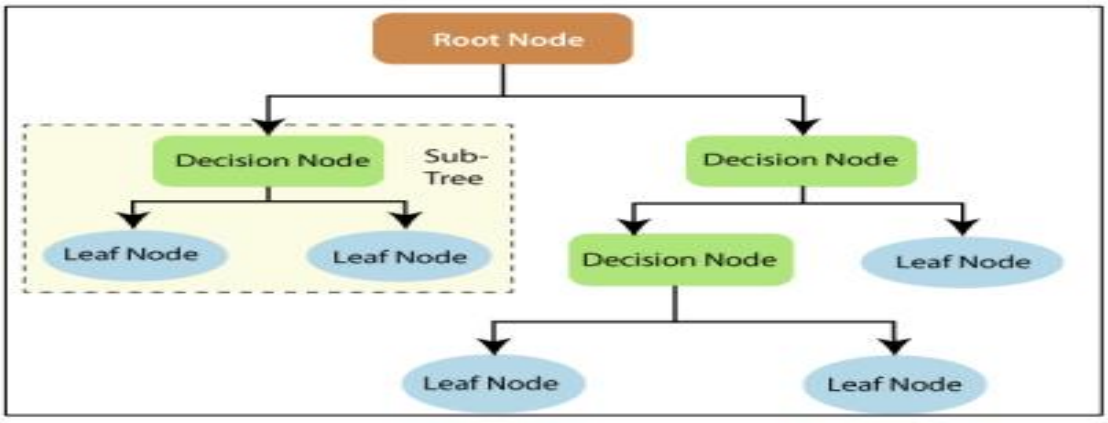


Figure 2- 4 Decision Tree (Harikumar & S R, 2019).

Important terminologies in Decision Tree (DT)

Root Node: The initial node at the beginning of a decision tree, where the entire population or

dataset starts dividing based on various features or conditions.

Decision Nodes: Nodes resulting from the splitting of root nodes are known as decision nodes. These nodes represent intermediate decisions or conditions within the tree.

Leaf Nodes: Nodes where further splitting is not possible, often indicating the final classification or outcome. Leaf nodes are also referred to as terminal nodes.

Sub-Tree: Similar to a subsection of a graph being called a sub-graph, a sub-section of a decision tree is referred to as a sub-tree. It represents a specific portion of the decision tree.

Pruning: The process of removing or cutting down specific nodes in a decision tree to prevent overfitting and simplify the model.

Branch / Sub-Tree: A subsection of the entire decision tree is referred to as a branch or sub-tree. It represents a specific path of decisions and outcomes within the tree.

Parent and Child Node: In a decision tree, a node that is divided into sub-nodes is known as a parent node, and the sub-nodes emerging from it are referred to as child nodes.

Attribute selection measure (ASM) techniques would be used while implementing decision tree to select the best attributes for the root node and sub-nodes (Malti, Apoorva, & Apoorva, 2022). There are two well-known techniques for ASM, those are:

- Information Gain
- Gini Index

Information Gain: is the measurement of changes in entropy after the segmentation of a dataset based on an attribute (Malti, Apoorva, & Apoorva, 2022) . It calculates how much information a feature provides us about a class. Based on the value of information gain, we split the node and create a decision tree. A decision tree algorithm always tries to maximize the value of information gain, and a node/attribute having the highest information gain would go first to split. It can be calculated as follows.

Information Gain= Entropy(S)-[(Weighted Avg)* Entropy(each feature)] (3)

Entropy: is a measure of impurity in a given attribute. To specify randomness in data. It can be represented as follows.

$$\textbf{Entropy(S)} = -P(yes)\log_2 P(yes) - P(no)\log_2 P(no) \quad (4)$$

Where,

S= Total number of samples

P(yes)= probability of yes

P(no)= probability of no

Gini index: is a measure of impurity or purity used at the time of creating decision tree in the classification and regression algorithm (Malti, Apoorva, & Apoorva, 2022). Compared to the attribute with high Gini index, the attribute with small Gini index should be preferred. It can be calculated as follows.

$$\text{Gini index} = 1 - \sum_{3} p_j^2 \quad (5)$$

2.5.2. Ensemble Learning

Ensemble learning is a technique that create two or more machine learning algorithms and combine them to obtain better performance compared to when the machine learning algorithms are used individually (Mienye & Yanxia, 2022). Rather than depending on single model, the predictions from the single learners are combined using a combination rule to obtain a single prediction that is more accurate. Ensemble algorithms have produced improved results and minimized the overfitting problem, With the combination of multiple learners and advantage of these learners. Three well know ensemble learning algorithms are bagging, boosting and stacking algorithms. Bagging is an ensemble method that involves merging several models trained on separate subsets of the original data (Zhang, Jingjing, & Wenjuan, 2022) . To create a bagging model, multiple datasets are created through bootstrapping the training data. Models are then constructed based on these individual datasets, and predictions are made using these models. The resulting predictions are aggregated to generate a representative value, such as the mean, median, or majority vote for classification, and averaging for regression, depending on the specific problem being addressed. Boosting algorithm transforms weak learners into strong learners using forward stagewise process by increasing the weights of training samples that were wrongly calculated in a successive iteration. The final result of boosting is obtained by combining the outputs from all iterations using a weighted vote for classification or a weighted sum for regression. So in this study, bagging and boosting algorithms would be used specifically random forest from bagging and extreme gradient boosting model from boosting algorithm.

2.5.2.1.Random Forest (RF)

Ensemble learning is one of the methods that generate numerous classifiers and aggregate their results (Boateng, Joseph, & Daniel, 2020). Boosting and bagging classification trees are two common methods of ensemble learning. Successive trees give extra weight to points incorrectly predicted by earlier predictors in the case of boosting. Each tree is independently constructed using a bootstrap sample of the data set, and successive trees do not rely on earlier trees in cases of bagging. Finally, a majority vote would be taken for prediction.

RF is a machine learning algorithm based on ensemble learning, and it is easy and flexible to use. The random forest algorithm was developed by Breiman and Cutler (Andronicus & Aderemi, 2014). RF classifier consists of a number of trees, on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset. It combines many decision trees that result in a forest of trees, resulting in improved results. Rather than depending on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output. The working steps of random forest are described as below:

1: Select random K data points from the training set.

2: Build the decision trees associated with the selected data points (Subsets).

3: Choose the number N for decision trees that you want to build.

4: Repeat Step 1 & 2.

5: For new data points, find the predictions of each decision tree, and assign the new data points to the category that wins the majority votes.

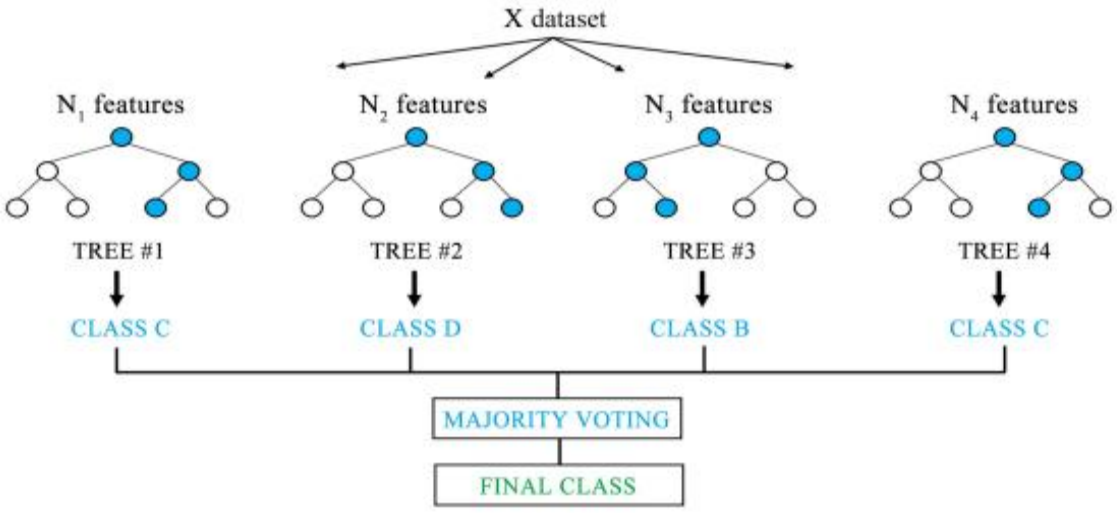


Figure 2- 5 Overview of random forest (Boateng, Joseph, & Daniel, 2020)

2.5.2.2.Extreme Gradient Boosting (XGBoost) Algorithm

XGBoost is a scalable machine learning algorithm and one of the ensemble learning algorithm, specifically the gradient boosting framework. XGBoost integrate multiple weak learners into one strong learner to improve the prediction accuracy (Xuzhi, et al., 2022) . It employs decision trees as base learners and employs regularization techniques to enhance model generalization. XGBoost offers significant advantages including high flexibility, strong predictability, strong generalization ability, high scalability, high model training efficiency, and great robustness. Known for its computational efficiency, feature importance analysis, and handling of missing values, XGBoost is mostly used for tasks such as regression, classification, and ranking.

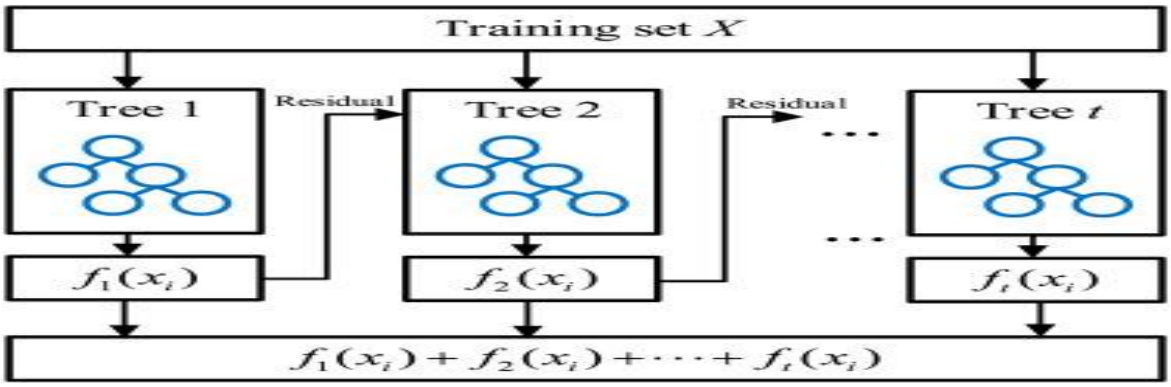


Figure 2- 6 Structure diagram of XGBoost algorithm (Xuzhi, et al., 2022)

The difference between simple gradient boosting algorithm and XGBoost algorithm is that differ in gradient boosting, the process of addition of the weak learners does not happen sequentially(one after the other); it takes a multi-threaded approach whereby proper utilization of the CPU core of the machine are utilized, leading to greater speed and performance (S, Nishant, Sunil, & Shatadeep, 2016).

Some features of XGBoost are as follows:

Regularization: XGBoost has an option to control complex models through both L1 and L2 regularization. Regularization helps in preventing overfitting (Xuzhi, et al., 2022). **Handling sparse data**: XGBoost Classifier integrates a sparsity-aware split finding algorithm to handle different types of sparsity patterns in the data. **Weighted quantile sketch**: XGBoost has a distributed weighted quantile sketch algorithm to effectively handle weighted data.

Block structure for parallel learning: To enhance computational speed, the XGBoost

Classifier can leverage multiple CPU cores by utilizing a block structure within its system architecture. Data is organized and stored in memory units known as blocks, allowing for the reuse of data layout across iterations, rather than recalculating it each time. **Out-of-core computing**: This feature optimizes the available disk space and maximizes its usage when handling huge datasets that do not fit into memory.

2.5.3. Deep Learning

Deep learning is subset of machine learning, which based on learning and improving on its own by evaluating computer algorithms. It is a network model with neurons having numerous parameters and layers in between input and output (Sharma, Reecha, & Neeru, 2021) . At different levels, deep learning offers automatic learning of features and their representation in a hierarchical manner. Because of this powerful process deep learning is robust compared with traditional machine learning. It is architecture is used for feature extraction and alteration process. Deep learning is appropriate for dealing with larger data and complexity, as the first layers perform simple processing of input data or learn the easy features and that output goes to the upper layers which perform complex features learning.

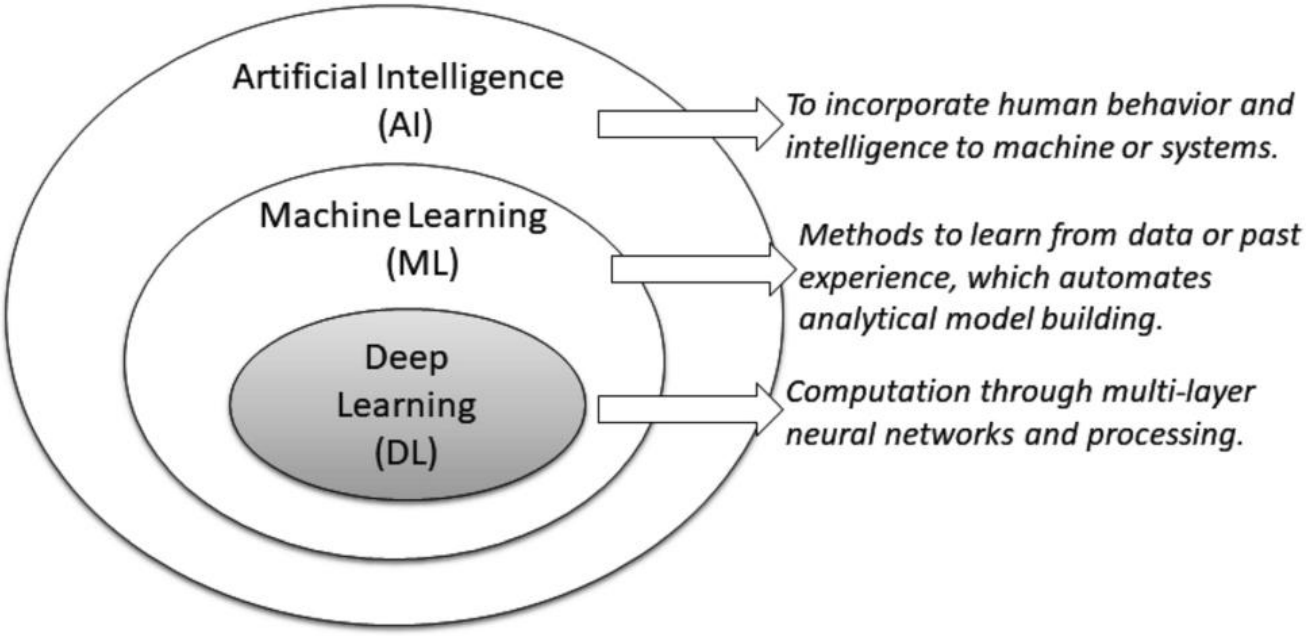
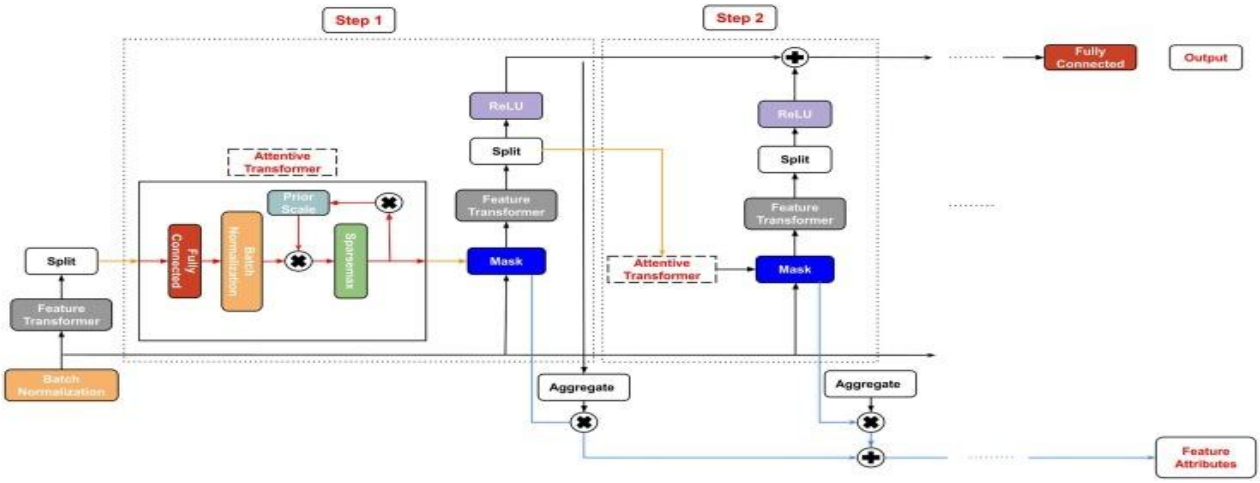


Figure 2- 7 Position of deep learning (Sarker, 2021).

2.5.3.1.TabNet Algorithm

TabNet is a deep learning model that offers the advantage of high performance and interpretability(both local and global interpretability) on tabular data learning (Arık & Tomas,

2021). Local interpretability illustrates the importance of features and their combinations, and global interpretability, which measures the impact of each feature to the model. TabNet classifier contains two blocks, the encoder and the decoder. The encoder block consists of a feature transformer, an attentive transformer and feature masking. The following figure presents the encoder architecture.



2- 8 TabNet Encoder Architecture *(McDonnell, Finbarr, Barry, Leandr, & German, 2023)*.

Feature transformer in the encoder consists of various layers. Every layers are composed of Fully-connected (FC) layers, a Batch Normalization (BN) layer, and a Gated Liner Unit (GLU) which is linked to a normalization residual connection, which helps in stabilizing the variance across the network.

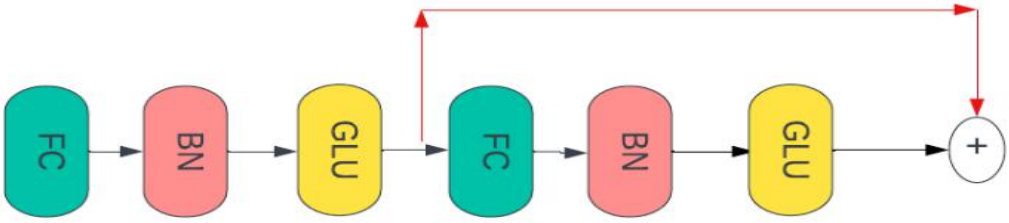


Figure 2- 9 A feature transformer block (Arık & Tomas, 2021).

At the end of transformation of the features the attentive transformer and the mask can continue for the task of robust feature selection. The function for sparsemax is:

a[i − 1]: M[i] = sparsemax(P[i − 1] · hi(a[i − 1])) (6)

where a[i − 1] is the previous step, P[i − 1] is the priori scale and hi some trainable function.

Sparsemax activation function and the prior scale are the two elements of attentive transformer. Sparsemax normalization promotes sparsity by transforming the Euclidean projection onto the probabilistic simplex, which is known to enhance performance and align with the objective of sparse feature selection for interpretability. The prior scale term, P[i], denotes the saliency of a feature throughout the previous steps and is defined as:

$$\mathbf{P[i] = \pi_{j=1}(\gamma\text{-}M[j])} \quad (7)$$

where γ is a relaxation parameter and defines the relationship between enforcement of a feature at one decision step or multiple steps. When γ=1 the feature is enforced at the given step and when γ=0 the feature is enforced multiple steps. The Attentive Transformer selects the most relevant features to form the transformed feature vector and passes these features to the learnable Mask M[j]. The Mask enables interpretability and further improves upon feature selection from the Attentive Transformer. The Attentive Transformer block is composed of a fully connected layer (FC), batch normalization (BN) and sparsemax normalization as shown in figure below. Each layer has a historical knowledge of how much each feature has been utilized in the previous layer. The sparsemax inside this block is being used to normalize the coefficients and enhance the robustness. Figure 2-10 shown below presents the attentive transformer.

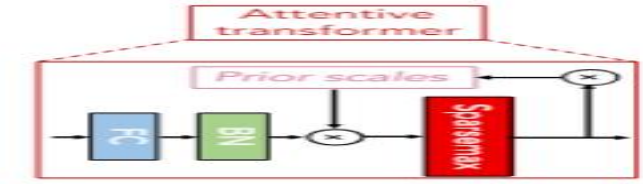


Figure 2- 10 An attentive transformer *(Arık & Tomas, 2021)*.

In this research work, we used the TabNetClassifier function from pytorch documentation which is suitable for binary classification tasks. The classifier is sickit-compatible, so we can leverages the use of many modules from sickit library while implementing the model. TabNet leverages sequential attention mechanism to determine the features to focus on during each decision step, enhancing interpretability and optimizing learning by prioritizing the most relevant features, thus making the learning process more efficient. This deep architecture helps in feature selection and enhance the capacity to learn high-dimensional features.

2.5.3.2. Convolutional Neural Network (CNN)

CNN is the well-known and commonly used model in the field of deep learning (Alzubaidi, et al., 2021). CNNs have been mainly applied in a range of different fields, including computer vision, speech processing, Face Recognition, etc. The structure and operation of the human visual cortex served as an inspiration for the pattern connection between its neuron (Li, Fan, Wenjie, Shouheng, & Jun, 2021). There are three major components of convolutional neural network such as Convolutional layer, pooling layer, and fully connected layer.

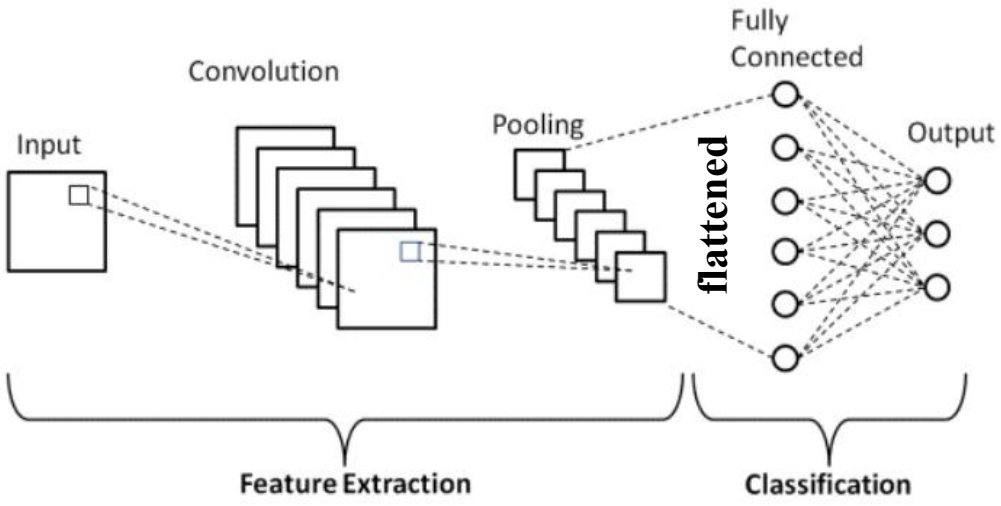
Convolution layer: is a fundamental component of a CNN (Bharadiya, 2023). It consists of several filters, also known as kernels, whose parameters are learned during training. The size of the filters is typically smaller than the input image. The image is convolved with each filter and resulting in an activation map. Convolution layers extract features using a combination of linear and nonlinear operations known as convolution operations and activation functions.

Pooling layer: pooling layers are used to reduce the dimensions of the feature maps. Thus, it reduces the number of parameters to learn and the amount of computation performed in the network (Bharadiya, 2023). This layer takes input and shrink them down while preserving the most important information in them. It keeps the maximum value from each window, it preserves the best fits of each feature within the window. Three common functions used in the pooling operation are:
average pooling, maximum pooling and sum pooling.

Fully Connected Layer: The final layer is the fully connected layers which represent the high-level filtered features data (Bharadiya, 2023). The input to the fully connected layer is the output from the final Pooling or Convolutional Layer, which is flattened and then fed into the fully connected layer.

Rectified Linear Unit Layer (Activation layer): The ReLU layer replace every negative number of the pooling layer with 0 and keep positive number as it is (Bharadiya, 2023). This helps the CNN stay mathematically stable by keeping learned values from getting stuck near 0 or blowing up toward infinity.

Dropout layer: Overfitting is one of the main challenges in deep neural networks with a large number of trainable parameters (Lee & Chulhee, 2020). Dropout is one of the most popular regularization methods used for preventing a neural network model from overfitting in the training phase. It is generally used to avoid overfitting for achieving a better prediction model. The basic CNN architecture is shown in the below figure.



CNN Architectures (Gurucharan, 2024)

2.5.4. Unsupervised Learning

Unsupervised learning is one of the machine learning, in which the algorithms learn patterns in the data without any explicit labeled data. Unsupervised learning is differed from supervised learning in the way that it is not given a target output rather it tries to draw the underlying pattern in the data itself. So, in unsupervised learning no specific target output is available for the data in the training data set and the aim ML is to fetch important information from the input values.

2.5.5. Reinforcement Learning

Reinforcement learning is a learning framework where agents are self-trained on reward and punishment mechanisms (Matsuo, et al., 2022). It improves the policy in terms of the given objective through interaction with an environment where an agent perceives the state of that

environment. Reinforcement learning is about taking the best possible action or path to gain maximum rewards and minimum punishment through observations in a specific situation. So, the main goal of reinforcement learning is simply to choose actions to make decisions that maximize future rewards as much as possible.

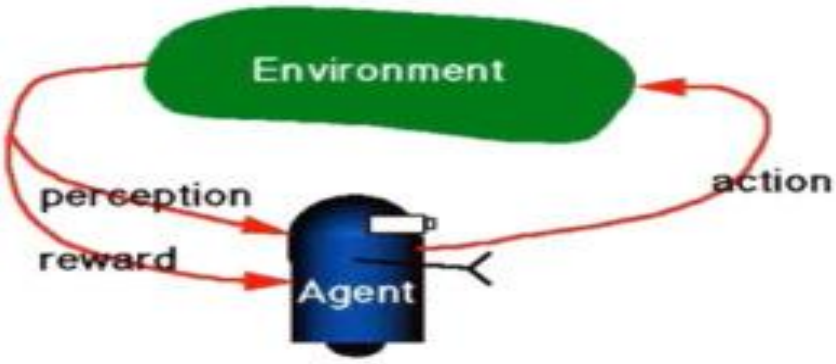


Figure 2- 11 Reinforcement Learning (Matsuo, et al., 2022).

2.6. Related Work

(Saroj, Yadav, Rajneesh, & Chilyabanyama, 2022) proposed machine learning algorithms for predicting/understanding the determinants of under- five mortality. For child health and social development of countries, under-five mortality is matter of concern that need to be considered. The researchers mainly focused to find the accuracy of machine learning models and recommend the model with better predictive performance and to identified the significant risk factors that best contribute to the deaths of children under the age of five. In the process researcher used data from the National Family Healthy Survey(NFHS-IV) of Uttar Pradesh. The total dataset used was about 41,751 records, out of those 41,751 records, 2,830 were deaths of the children under the age of five. For data analysis, multivariate logistic regression used to find significant factors and they used machine learning algorithms such as decision tree, random forest, Naïve Bayes, K-nearest neighbor (KNN), logistic regression, Support Vector Machine (SVM), neural network and ridge classifier. The accuracy of each selected models was evaluated by confusion matrix, accuracy, precision, recall, F1 score, Cohens Kappa and AUC-ROC. Thus, machine learning algorithms are applied on dataset and obtained results for each model. Based on results obtained, neural network was found to be the best predictive model compared with other

predictive models having the accuracy (95.29% to 95.96%) and significant risk factors of under-five mortality were identified such as number of living children, survival time, wealth index, child size at birth, birth in the last five years, the total number of children ever born, mother's education level, and birth order.

(Fikrewold, Samuel, Lloyd, & Corey, 2020) proposed a machine-learning models used to predict determinants of under-five mortality in Ethiopia. Ethiopian Demography and Health Survey data of 2016 were used in this study. The researchers employed three machine learning algorithms such as random forest, KNN, and logistic regression. The models were evaluated based on accuracy and the Receiver Operating Characteristic (ROC) curve. Based on the evaluation metrics, the result of the model shows between 46.3% to 67.2%. Random forest scores 67.2% showing the best performance from the other model. In this study spatial variations of under-five mortality were also identified. The random forest models identified that household size, time to the source of water, breastfeeding status, number of births in the preceding 5 years, sex of a child, birth intervals, antenatal care, birth order, type of water source, and mother's body mass index are important risk factors that contribute to under- five mortality levels in Ethiopia.

(Adegbosin, B, & J, 2019), proposed a predictive model for child under-five mortality for low and middle-income countries to explore the efficiency of deep learning techniques in predicting the under-five mortality and identify significant predictors of under-five mortality. Deep learning techniques which are deep neural networks, convolutional neural networks, logistic regression, and hybrid CNN-DNN are used to build a prediction model to predict child under-five mortality and compare which DL model achieves better performance from listed techniques. The researchers found the duration of breastfeeding, household wealth index, and the level of maternal education are the most important determinants of under-five mortality and from the models, Hybrid CNN-DNN scores 0.71 sensitivity and 0.83 specificities better than the others.

(Carlos, et al., 2023) proposed machine learning approaches for understanding the social determinants of child mortality in Latin America. In the process the researcher applied random forest machine learning algorithm on aggregated data of health data

and socioeconomic data from three countries namely Brazil, Ecuador and Mexico. On this work the researchers abled to identify the significant risk factors contributing to the death of children under the age of five. Poverty, illiteracy, and the Gini index were the most important variables for predicting U5M based on the random forest machine learning algorithms. To reduce under-five mortality rate in Latin America, the researcher recommended long-term public policies to give focus on reducing poverty, illiteracy, and socioeconomic inequalities. The research produced the knowledge on the relationship between social determinants and child mortality.

(Solomon, Angela, Oluwafemi, Christabel, & Ignace, 2023) employed two machine learning algorithms namely Logistic Regression(LR) and Deep Neural Network (DNN) to two main objectives. One is to examine the trends of under-five mortality in Nigeria from 2003 to 2018. Second, to identify or determine the significant risk factors (determinants) that best contribute to under-five mortality in Nigeria. The research was drawn from Nigeria Demographic and Health Survey (NDHS) data. The survey data were collected in 2003, 2008, 2013, and 2018. Thus, the researchers used these four survey data points to evaluate the trends of under-five mortality during the specified period. The researchers divided the dataset into 70% for training and 30% for testing and used a random forest classifier to select significant under-five mortality variables. The trend results revealed that the under-five mortality rates were 200.72, 156.86, 128.05, and 132.02 per 1,000 live births in 2003, 2008, 2013, and 2018, respectively. So after applying LR and DNN machine learning algorithms, DNN happened to be the best performer compared to LR in terms of the accuracy produced. The accuracy of logistic regression on the test data set was about 60%, while the accuracy of deep neural networks was 74%. Breastfeeding had the highest contribution to the deaths of under-five children in Nigeria, based on the variable importance of random forest classification. Mothers' marital status, frequency of reading newspapers (media exposure), delivery by caesarean section, and smoking of cigarettes were found to be the lowest contributing factors for the deaths of children younger than five years.

No	Research Title	Algorithm	Author	Limitation

1	Machine Learning Algorithms for understanding the determinants of under-five Mortality(2022).	decision tree, random forest, Naïve Bayes, K-Nearest Neighbor (KNN), logistic regression, Support Vector Machine (SVM), neural network and ridge classifier	Saroj, Yadav, Rajneesh, & Chilyaban yama.	• Limited to identify only socio-demographic factors that are determinants for U5M • Dataset used is less representativ e.
2	Machine learning approach for predicting under-five mortality determinants in Ethiopia (2020).	Random forest, KNN, and logistic regression	Fikrewold , Samuel, Lloyd, & Corey	• Limited to identify only socio-demographic factors that are determinants for U5M • Dataset used is less representativ e. • Achieved low accuracy
3	Predicting Under-five	Deep Neural Networks,	A. E. Adegbosi	• Limited to identify only

	mortality across 21 Low and Middle countries.	Convolutional Neural Networks, Logistic Regression, and Hybrid CNN-DNN	n, B. Stantic, & J. Sun.	socio-demographic factors that are determinants for U5M
4	Understanding the social determinants of child mortality in Latin America over the last two decades: a machine learning approach (2023)	Random Forest	Carlos et.al.	• Limited to identify only socio-demographic factors that are determinants for U5M.
5	Trend Analysis and Determinants of under5 Mortality in Nigeria: A Machine Learning Approach.	Logistic Regression(LR) and Deep Neural Network (DNN)	Solomon, Angela, Oluwafemi, Christabel, & Ignace	• Limited to identify only socio-demographic factors that are determinants for U5M • Achieved moderate

				accuracy

Table 2- 2 Summary of related works

Generally, most of the researchers employed machine learning algorithms for prediction of under-five mortality to identify the determinants that contribute to the death of children younger than five years using cross-sectional data. However, they were limited to identify socio-demographic factors, and the data used were also less representative. In this research work, we did not only address the factors of socio-demographics but also identified the climate change factors on under-five children, as the children are likely to be vulnerable to the climate change. In this study, we have integrated health, socio-demographic data and climate data of three sites(Haramaya, Kersa, and Harar) and developed the model that better predict under-five mortality using machine learning approaches.

CHAPTER 3
3. RESEARCH METHODOLOGY

3.1. Overview

In this chapter, the selected research methodology to undergo the research, the designed process model and its components are discussed. Section 3.2 presents the research methodology used and presents how each parts of design science research methodology is applied in this study.

3.2. Research Design

The choice of the appropriate research methodology is a significant step in conducting research and should be based on the statement of the problem. Methodology is a system of principles, practices, and procedures that are applied to a particular field of study (PEFFERS, TUURE, ROTHENBERGE, & SAMIR, 2007). In Design Science research, a methodology comprises three components: conceptual principles that clarify the essence of DS research practice rules, and a process of conducting and presenting the research. Design Science is an outcome-based methodology in information technology which offer a specific guide for evaluation and iteration within research work or projects (Bisandu, 2016). Design science creates and evaluates IT artifacts intended to solve identified organizational problems (Hevner, Sudha, March , & Jinsoo, 2004). It includes a thorough process for creating artifacts to resolve identified problems, to contribute to research, to evaluate the designs, and to share the findings with appropriate audiences. These artifacts may encompass constructs, models, methods, and instantiation (PEFFERS, TUURE, ROTHENBERGE, & SAMIR, 2007) . The beginning point of any scientific study is identifying the reason for conducting the research study. A research methodology that best fit the research would be selected, to achieve the aims and objectives of the research (data collection, analyze, interpretation and provide the result or solution for identified problems). Thus, design science research methodology is ideal for this study. The main aim of this study is to design and develop a predictive model for under-five mortality based on health, socio-demographic and climate data in eastern Hararghe, Ethiopia using machine learning approaches.

Design Science Research (DSR) is a problem-solving method that aims to improve human knowledge by creating innovative artifacts. In simple terms, DSR aims to enhance technology

and science knowledge bases by developing inventive solutions that address problems and enhance the environment in which they are implemented (Brocke, Alan, & Alexander, 2020). In this research work, design science approach by (PEFFERS, TUURE, ROTHENBERGE, & SAMIR, 2007) is adopted in which design science process includes six steps in nominal sequence: problem identification and motivation, definition of the objectives for a solution, design and development, demonstration, evaluation, and communication. problem-centered initiation, objective-centered solution, design and development-centered initiation and client/context are four possible entry points in DS approaches. In this research work, Design Science Research Methodology (DSRM) process is used to create an artifact to efficiently address prediction of under-five mortality and identification of factors best contribute to the death of child younger than age of five in eastern Hararghe, determine objectives for a solution, design and develop an artifact that can be provide in a solution, demonstrate the use of the artifact, evaluate the artifact and to finally communicate the process to others.

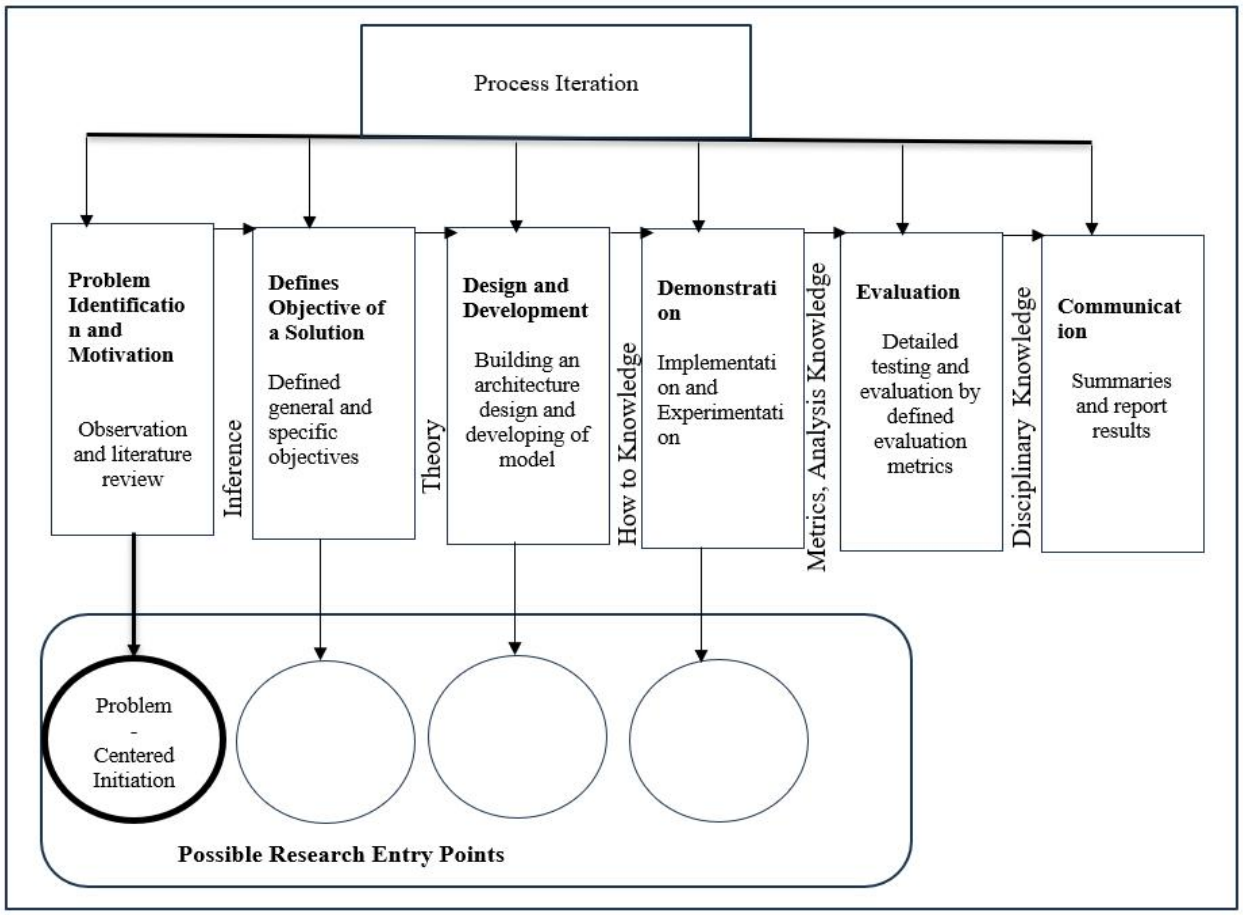


Figure 3- 1 Design Science Research Process Model (PEFFERS, TUURE, ROTHENBERGE, & SAMIR, 2007)

For this research work, the first approach a problem-centered and initiation approach is selected as the entry point of the research, because of the problem observed in under-five mortality prediction that utilize integrated and comprehensive health, socio-demographic and climate data to address both the health, socio-demographic factors and the impacts of climate on children younger than the age of five.

3.2.1. Problem Identification and Motivation

Problem identification is the first step of design science research methodology. The research problem is identified and the relevance and motivation of the research is justified with the help of the researcher's detail investigation of the literature review and prior observation. To get deeper insight into under-five mortality prediction and detail knowledge on the techniques utilized in under-five mortality prediction, the researcher conducted a literature review. The details of problem identification and motivation is discussed in 1.2 section.

3.2.2. Define Objective of Solutions

Objective definition is the second activity of design science research methodology. The objective of the solution is inferred from the problem identified in previous activity which is problem identification and motivation. The main objective of this study is to develop a model that predict under-five mortality on health, socio-demographic and climate data in eastern Hararghe, Ethiopia.

3.2.3. Design and Development

Design and development are the third steps of design science research methodology, which involves model design and development. This step is achieved through the architecture of the proposed system discussed in chapter 4. It involves artefact design in order to predict under-five mortality based on health, socio-demographic and climate data in eastern Hararghe using machine learning approaches to address the factors (health, socio-demographic and climate) affecting under-five children. The main activity in this step is to design and develop the architecture for under-five mortality prediction.

Python programming with anaconda distribution is used in order to develop and implement the model. Python is a simple and powerful programming language with excellent features and extensive support libraries. Anaconda is a free software that gives you a toolkit that is tailored for research and science. Installing Anaconda gives you access to different environments that allow you to code in either Python or R. These environments, also known as integrated

development environments (IDEs). Python has powerful libraries or packages like NumPy, Pandas, Matplotlib, SciPy, Scikit-learn, TensorFlow and Keras.

Scikit-learn is a library for python machine learning library, which contains simple and efficient tools for data mining and data analysis algorithms for both supervised and unsupervised problems. TensorFlow is an open-source deep learning library that provides a flexible framework for building and training neural networks, especially for tasks like image and text recognition. Keras is a high-level neural network library that runs on top of TensorFlow, offering a user-friendly API for building and training deep learning models (Damien , Matt, Thadd´e, & Kinsey, 2020). We also used computers with specifications: Dell PC of Intel(R)Core(TM) i7-12800H CPU @ 1.9GHz 1.9 GHz,16 GB of RAM, 1TB of hard disk capacity, with Microsoft Windows 10 Pro 64-bit operating system and also used Microsoft word 2019 for writing the report.

3.2.4. Demonstration

Demonstration is a process of proving to what extent the artifact solves the problem identified. This involve its use in experimentation, simulation and proof. In this activity, to demonstrate or present the performance of the system and its acceptance by the end users, the research uses different test cases in the actual process. Demonstrating or presenting how the designed artifact (research result) solves problem of the domain area utilizing a selected optimal model is the main task we have in this activity.

3.2.5. Evaluation

The Evaluation activity involves analyzing and evaluating the results of the proposed model in predicting under-five mortality based on socio-demographic and climate data. We employed the most commonly used evaluation metrics such as accuracy, precision, recall, F-score, and AUC-ROC score. The detailed results of each evaluation metric are discussed in Section 5.

Precision: is a metric that measures the proportion of selected data items that are relevant. It measures the proportion of correctly predicted positive instances out of all instances that were predicted as positive. Precision answers the question: "Out of the observations that the model predicted to be positive, how many of them are actually positive?" It focuses on the accuracy of the positive predictions (Meysam, Mohammad, & Masoumeh, 2020). It can be calculated by dividing the number of true positives by the sum of true positives and false positives.

$$\textbf{Precision} = \frac{TP}{TP+FP} \quad (9)$$

Accuracy: is a commonly used metric in machine learning to evaluate the performance of a classification model. It measures the proportion of correctly predicted instances out of all the instances in the dataset. It answers the question, "How many of the predictions made by the model are correct?" It provides an overall assessment of the model's ability to correctly classify instances into their respective classes (Meysam, Mohammad, & Masoumeh, 2020) . The formula we use to calculate accuracy is given as:

$$\textbf{Accuracy} = \frac{TP + TN}{TP+TN+FP+FN} \quad (10)$$

Recall: reveals "how many relevant data items are identified or selected". Recall is also known as sensitivity or the true positive rate, It measures the ability of the model to correctly identify all positive instances out of the total actual positive instances in the dataset (Meysam, Mohammad, & Masoumeh, 2020). The formula we use to calculate recall is given as:

$$\textbf{Recall} = \frac{TP}{TP+FN} \quad (11)$$

F1-score: is also known as f-score or f-measure, which takes both precision and recall into account in order to calculate the performance of an algorithm. It is the harmonic mean of precision and recall which is a score computed from both the precision and the recall. The F1 score ranges from 0 to 1, where a score of 1 represents perfect precision and recall, while a score of 0 represents the reverse result (Sikha & Kunqi, 2021) . The formula we use to calculate F1-score is given as:

$$\textbf{F1-score} = 2 * \frac{Precision * Recall}{Precision + Recall} \quad (12)$$

True positives (TP): Predicted positive and are actually positive.

False positives (FP): Predicted positive and are actually negative.

True negatives (TN): Predicted negative and are actually negative.

False negatives (FN): Predicted negative and are actually positive (Sikha & Kunqi, 2021).

Confusion matrix: is one of the most intuitive and descriptive metrics used to find the accuracy and correctness of a machine learning algorithm. Its main usage is in classification problems where the output can contain two or more types of classes (Meysam, Mohammad, & Masoumeh, 2020).

Area Under Receiving Operating Characteristic(AUC-ROC): is a widely used evaluation metric in machine learning. It measures the performance and discriminative power of a classification model by analyzing the trade-off between true positive rate (TPR) and false positive rate (FPR) (Meysam, Mohammad, & Masoumeh, 2020).

True Positive Rate (TPR) is a synonym for recall or sensitivity and is therefore defined as follows:

$$\mathbf{TPR} = \frac{\boldsymbol{TP}}{\boldsymbol{TP+FN}} \quad (13)$$

False Positive Rate (FPR) measures the proportion of actual negative instances that are incorrectly classified as positive by the model. It is calculated as:

$$\mathbf{FPR} = \frac{\boldsymbol{FP}}{\boldsymbol{FP+TN}} \quad (14)$$

The AUC-ROC score ranges from 0 to 1, where 0.5 indicates random guessing, and 1 indicates perfect performance.

3.2.6. Communication

Communication is the last step of the design science research methodology, and it involves summarizing the test results, presenting conclusion and recommendation from the experimentation results, the initial description of this research work is presented in the thesis work for scholarly sessions. The result of this research work will be submitted to the department of computer science as partial fulfilment of MSc. degree in Computer Science. Further, the research can also be published in a conference or journal.

CHAPTER 4

4. SYSTEM ARCHITECTURE AND DESIGN

4.1.Overview

In this chapter, the overall of the design of the model we proposed for under-five mortality prediction using machine learning approaches in eastern Hararghe, Ethiopia is discussed in detail. First, the general overview of the proposed model is discussed, second, how data preprocessing from data cleaning to feature selection has been performed is described. Thirdly, how machine learning algorithms are employed for prediction is discussed.

4.2. System Architecture

The proposed system architecture for the prediction of under-five mortality on health, socio-demographic, and climate data consists of data cleaning, data integration, data balancing, feature scaling, feature selection, dataset splitting, training, testing, and evaluation .

System architecture for Under-five mortality prediction based on health, socio-demographic, and climate data using machine learning approaches is shown as below.

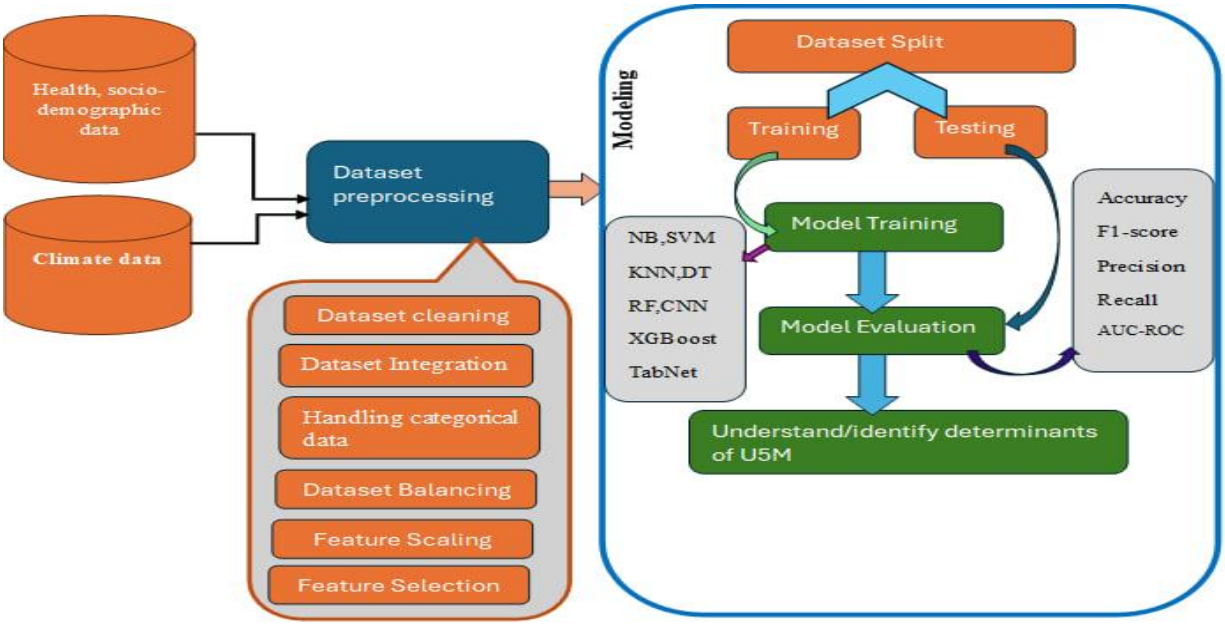


Figure 4- 1 Proposed System Architecture

4.3. Data Source

In this study, we have used a secondary dataset of Hararghe Health Demographic Surveillance and climate data of three sites (Harari, Haramaya, and Kersa). Health Demographic Surveillance System (HDSS) is a longitudinal, population-based health and vital event registration system that monitors demographic and health events in a geographically defined population with timely production of quality data. For evidence generation, specifically in rural areas and the countries with weak vital event registration, HDSS is considered to be one of the best sources of representative data (Dheresa, et al., 2022) . Hararghe HDSS was established by Haramaya University and is currently running on three different sites, namely, Kersa (Rural), Harari (Urban), and Haramaya (Rural). Initially it was established on the Kersa site and called Kersa HDSS. Kersa HDSS is located in the eastern part of Ethiopia, and Kersa is 476 km away from Addis Ababa. It was established in 12 sub-villages (called Kebeles, locally) of the Kersa district, Eastern Hararghe, Oromia Region, Ethiopia. The site is primarily rural with two small towns (Kersa and Weter). At baseline, a total of 10,522 households and 50,830 people were registered.

The baseline census was conducted in 2007, and since then the households that were captured during baseline have been updated every 6 months (twice a year), with registration of demographic and vital health events such as birth, death, morbidity (child and adult), immunization, pregnancy observation, in-migration, out-migration, family planning, house condition, and economic information. Currently, the Kersa HDSS site has been expanded to 24 villages by doubling the initial number of catchment areas. Having the total household 26,148 and total resident population 146,414.

Harar is the capital of the Harari region and is located 520 km east of Addis Ababa. The Harar HDSS was established in 2012. Currently working on 12 villages, having a total household of 12,830 and a total resident population of 51,361. Haramaya HDSS was established in October 2018 by Haramaya University to be one of the broad and sustainable representative data sources for monitoring health and socio-demographic data of the population in the Haramaya district. Haramaya HDSS is currently running in 12 villages, having a total resident population of 118,483 and a total of 20,582 households. Therefore, Hararghe HDSS is totally working on 48 villages under three sites (Nega, et al., 2016). Climate data of three sites, namely Harari, Kersa, and Haramaya, was taken from the Ethiopian National Meteorology Agency (NMA).

4.4. Data Preparation

Before proceeding into data analysis, data must be organized into an appropriate format to fit and evaluate machine learning models. Data preparation is a process of manipulating and organizing data before proceeding with data analysis. It is an iterative process of manipulating raw data, which perhaps converting unstructured and messy data into more structured, actionable and useful data that allows for further analysis using machine learning models (Zahraa, Lan, Geoffrey,2017).

4.4.1. Dataset Cleaning

Data cleaning is one of the pre-processing steps and the time-consuming tasks in machine learning projects. It involves identifying and removing or correcting the outliers, noise, missing values, duplication and inconsistence from the data. Health, socio-demographic data has some variables with missing values which reduce the performances of the machine learning models. Thus, handling the missing values in the dataset improves the performance of the machine learning models.

4.4.1.1. Handling Missing Values

Currently, missing values are a well-known challenge in a real-world dataset. Handling missing values is one of the main tasks in the data cleaning process (Pratishtha & Navneet, 2023) . In most cases, the raw data obtained is unorganized, duplicated and noisy. This can affect accuracy while predicting the outcomes. Since higher accuracy is crucial for the deployment of a correct model, data cleaning becomes important to prepare a precise machine learning algorithm. And involves checking and identifying the existence of missingness in each variable of the dataset. So, if the missing values exist in the dataset, we need to make sure the appropriate measure has been taken to handle the missingness in the dataset. To handle missing values in the dataset, we used different imputation techniques, and we found the iterative imputation with the Random Forest algorithm performs better compared with others. The algorithm for handling missing values is given below.

Input: dataset with missing values

Output: dataset with no missing values

For all columns in the dataset

- Check the columns with null or missing values
- Initialize for initial temporary placeholder.
- Model building for each column with missing values, a model is built using the other columns as predictors.
- Iteratively impute the values for each column.
- The process continuous until the imputed values stabilize.
- Return the dataset with no missing values

Algorithm 4- 1 Algorithm to handle the missing values

4.4.2. Dataset Integration

Dataset integration is the process of combining datasets from multiple sources to create a unified and comprehensive dataset. For under-five mortality prediction we used data from two sources, one is health, socio-demographic data from Hararghe Demographic Surveillance System and climate data of three sites (Harar, Haramaya and Kersa) from Ethiopia national meteorology agency. To integrate the two datasets, we used a common variable from both datasets and combined the datasets based on the common variables identified to generate the unified and comprehensive dataset. The algorithm for dataset integration is give below.

Input: health, socio-demographic dataset, and climate dataset.

Output: one unified dataset

For all records in the dataset:

- Find a common variable in both dataset such as site, year and month
- Join the two datasets based on the common variables
- Return the unified dataset

Algorithm 4- 2 Data Integration

4.4.3. Handling Categorical Data

Categorical data in Machine Learning is the data that consists of categories or labels, rather than numerical values. Handling categorical data is the crucial part of machine learning preprocessing, as many algorithms require numerical input. In this step, the collected information is converted into a format that represents categories numerically. In this process, nominal data, which includes values like 'Yes' or 'No,' is transformed into numerical values, typically 0 and 1. For example, in the case of the "attended ANC" attribute, 'No' represented as 0, and 'Yes' represented as 1. This numerical representation allows machine learning models to work with categorical data effectively. In handling categorical data and after preprocessing the data should be in CSV file forms which hold the entire integer and float values data.

4.4.4. Dataset Balancing (Handling Imbalanced Dataset)

Imbalanced datasets occur if the sample of one class significantly outnumbers the sample of

another class (Danquah, 2020). The class with the highest number in the data is referred to as the majority sample class and the class with the least number in the data is minority sample class. Imbalanced data is one of the main problems in terms of solving classification problems up to date. By default, majority of machine learning models perceive that all data distribution is equal, or all data is balanced, the algorithms do not take into consideration the distribution of the data sample class. The outcomes tend to be biased and skewed towards the majority sample class distribution. So, if machine learning is favoring the majority sample class, that implies we are getting the opposite result of what we expect to get. Imbalanced data classes are a common occurrence in many real-life applications, such as mortality data, where there is a significant difference between the number of survivors and the number of mortality cases.

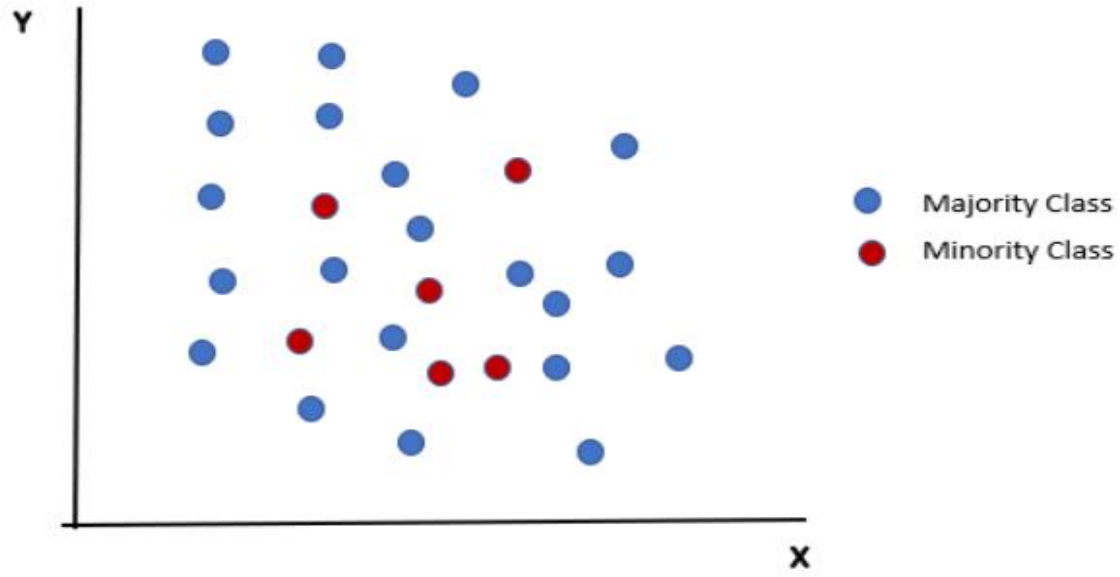


Figure 4- 2 Imbalanced class representation of binary data

Handling imbalanced dataset is very important in binary or multi-class classification for machine learning models. In this research work we have used Synthetic Minority Oversampling Technique (SMOTE), Adaptive Synthetic (ADASYN) and SMOTE+TOMEKlinks to handle the imbalanced dataset.

4.4.5. Synthetic Minority Oversampling Technique (SMOTE)

In SMOTE technique, minority class is over sampled by generating synthetic samples rather than by oversampling with replacement for simple random oversampling (Danquah, 2020) .

To avoid the issue of over fitting when increasing minority sample class, SMOTE creates synthetic data points by working within the current feature space. New synthetic data points are extracted from interpolation, so the original data set still has significance. SMOTE interpolates values using a K - nearest neighboring technique for each minority class instance and generates attribute values for new data instances. A new synthetic data point is created for each minority sample data by taking the difference between the a minority sample class feature vector and the nearest neighbor belonging to the same sample class and multiplying it by a random number between 0 and 1 and then adding the results back to the minority sample class feature vector. This creates a random line segment between every pair of existing features. This results in a new instance generated within the data set. The algorithm for the SMOTE techniques is given as below.

Input: imbalance dataset

Output: balanced dataset

Let x1, x2,...,xn be the minority class feature vectors in the n dimensional space of X

Let N be the number of synthetic instances to generate

Let K be the number of nearest neighbour

Synthetic set of artificial instances

1. For i in range (N) do,
2. Select randomly a minority class feature vector xi
3. From xi's K-nearest minority class neighbors, randomly select a neighbor xˆi
4. diff = xˆi – xi δ =random number between 0 and 1
5.new Sample= xi + diff * δ
6. Synthetic ⇐ newSample

Algorithm 4- 3 SMOTE Algorithm

4.4.6. Adaptive Synthetic (ADASYN)

Adaptive Synthetic sampling is one of the data balancing techniques primarily used in machine learning to address the issues of imbalanced datasets. It is an extension or improvement of the SMOTE algorithm. ADASYN has similarity with SMOTE but differ in

the sample created. ADASYN works by calculating the degree of class imbalance, then calculating the number of synthetic data examples that are needed to be generated. It attempts to generate more synthetic instances on the region with underrepresented instances than one with well represented instances to increase the recognition of positive class. This algorithm uses the number of negative neighbors in K-nearest neighbors of each positive instance to form a distribution function. The distribution function determines how many synthetic instances are generated from that positive instance. The algorithm for ADASYN sampling approach is given below.

Input:

Let m be the number of minority samples

Let n be the number of majority samples

Let β be the ratio of the balance level of the synthetic samples. NB: $\beta \in (0, 1]$

Let xi for i=1,2,3...m be the minority class feature vectors in the n dimensional space of X

Let G be the number of synthetic instances to generate

Let gi for i=1,2,3...m be the number of synthetic data generated for each xi

Let K be the Number of Nearest Neighbour

Let $\delta \in [0,1]$

Output:

Synthetic set of artificial instances $G = \beta \times (n - m)$

For i in range (m), Find K for every xi Calculate ri = fk/K , where fk is the number of feature vectors in the K nearest neighbors belonging to the majority class

Calculate $\hat{r}i = ri / \sum_{i=1}^{m} ri$, so that ($\sum_{i=1} \hat{r}i = 1$)

Calculate $gi = \hat{r}i \times G$

For i in range (gi) and for j in range (m), do

From xi's K-nearest minority class neighbors, randomly select a neighbor $\hat{x}ij$

diff = $\hat{x}ij$ - xi

$NewSample_{ij} = xi + diff * \delta$

Synthetic ⇐ $NewSample_{ij}$

Algorithm 4- 4 ADASYN Algorithm

4.4.7. SMOTE+TOMEKlinks

SMOTE+TOMEKlinks is a hybrid technique used for handling imbalanced dataset. Tomek link was initially developed by Tomek, which was basically designed for two different classes (one majority and one minority), where, if the majority and minority classes are xa and xb, then the distance between them would be $d(xa,xb)$ and is known as the Tomek link, provided that no other class xz such that

$d(xa,\ xz)<d(xa,xb)$ or $d(xb,\ xz)<d(xa,xb)$. T-link works by removing the majority class instances that are closer to the minority class by applying the nearest neighbor rule to select instances (Swana, Wesley, & Pitshou, 2022) . So, while the SMOTE technique generates synthetic samples for the minority class by interpolating between existing minority class instances and their nearest neighbors in the majority class, TOMEKlinks identifies and removes pairs of instances that are likely to cause noise or bias in a machine learning model.

4.5. Feature Scaling

Feature scaling in machine learning adjusts or transforms the numerical features of a dataset to enhance the performance of models, in both accuracy and the speed of convergence (Alshaher, 2021) . It is a data preprocessing technique used to standardize the range of features or variables in a dataset. Feature scaling is a crucial step in the data preprocessing step of machine learning, as features with different scales can have a disproportionate impact on the model's performance. Feature scaling is crucial for machine learning models that compute distances between data, such as SVM and KNN, because without scaling, features with larger numerical values have a greater effect on the distance between data and dominate other features when calculating distances. On the other hand, feature scaling has no effect on machine learning techniques like XGBoost, RF, and NB. In this study, we have utilized the StandardScaler module to normalize the categorical features and rescale the data in order to have values between a mean of 0 and a standard deviation of 1 and it is the most common feature scaling technique.

4.6. Feature Selection

Feature selection is the process of selecting a subset of relevant features from the larger set of features to use it in machine learning algorithms. It is a common way to minimize the problem of excessive and irrelevant features. Generally, feature selection methods reduce the

dimensionality of the training data by excluding a feature that has low or negligible predictive power and features that are redundant to each other. For this research work we have used an advanced feature selection methodology that leverage the benefit of both forward and backward feature selection techniques called Recursive Feature Elimination (RFE) from the wrapper feature selection techniques. RFE is the process of eliminating the least important features whose deletion will have less effect on the training errors, leading to an improved set of features that highlights the important patterns and variations in the data (FILIOU, 2023). The function from the sklearn package is utilized for this purpose, and in order to evaluate the feature importance, this function used an estimator. Usually tree-based models are effective since they have the ability to adjust the importance of scores by themselves. Random forest classifier is used as the predictor because it is tree-based model. This method helps in extracting crucial features from the dataset for training the machine learning models. And also we have applied embedded method feature selection. In an embedded method, feature selection is integrated or built into the classifier algorithm. Embedded methods are an intermediate solution between filter and wrapper methods in the sense that the embedded methods combine the qualities of both methods. Decision tree, random forest and gradient boosting are some examples of embedded methods.

4.7. Data Splitting and Model Training

In this research work, we have used data fusion (feature level fusion), which is the process of combining data from multiple sources to create a more comprehensive and accurate representation of a phenomenon (Pereira, Addisson, & Luis, 2023). We have integrated health, socio-demographic data, and climate data from two different sources, such as the Hararghe Health Demographic Surveillance System and the National Meteorology Agency Ethiopia, to create a unified and comprehensive dataset.

Data splitting is a significant step in machine learning where the dataset is divided into two subsets for training and testing purposes. It helps to prevent overfitting, ensure the robustness of the model, and assess the model's generalization ability. In this step, we split the dataset into training and testing. For the purposes of avoiding overfitting, underfitting, and enhancing the efficiency of the model, the training dataset should be greater than the test dataset. So, 80% of the dataset is used for training purposes, which was eventually used for 5-fold cross-validation to tune the model parameters for optimal performance, and 20% of the dataset is

used for evaluation (testing) purposes. The dependent (outcome) variable is transformed into binary format, where zero represents low risk and 1 represents high risk. Independent features, specifically categorical features are transformed to numerical data types.

Model training is the process of teaching a machine learning model to learn patterns and relationships from the given input data. It involves feeding the input dataset to machine learning algorithms and allowing it to adjust its internal parameters to minimize the error between its predictions and the actual output values. This is done by iteratively adjusting the model's parameters until it produces the best possible, accurate predictions on that data. Training a model can involve much iteration, with the aim of improving model accuracy by minimizing predictive errors. The final goal of training a model is to produce a model that can accurately predict outcomes on unseen data. We have used important machine learning models like Naïve bayes, decision tree, random forest, support vector machine, k-nearest neighbor, extreme gradient boosting, convolutional neural network and tabnet model.

4.7.1. Hyperparameter Tuning

Hyperparameter tuning is the process of selecting the optimal hyperparameters for a machine learning model (Aryo & Salsa, 2021). Hyperparameters are variables that are chosen before the learning process and cannot be discovered through direct data learning. They control various aspects of the model's learning process and can have a significant impact on the model's performance. Hyperparameter configuration can be done manually or automatically. The first approach involves manually setting and experimenting with various groupings of hyperparameters. This is tedious and may not be practical in cases where there are many hyperparameters to try and a large search space. Automatic hyperparameter tuning, on the other hand, uses algorithms and techniques to automate the process of finding the best hyperparameter configuration. Random search and Grid search are two commonly used algorithms for this purpose. Grid Search exhaustively searches through a predefined subset of the hyperparameter space. It evaluates the model's performance for every possible combination of these values. Random search selects a random subset of combinations to evaluate the model performance, instead of checking every possible combination. In this research work, we have utilized the hyperparameter tuning techniques with supervised machine learning models to obtain an optimal result in the prediction of under-five

mortality. The hyperparameter optimization can be done using tuning to learn an algorithm based on existing data. The optimal performance of an algorithm depends on the hyperparameter in supervised learning and in some scenarios (Aryo & Salsa, 2021). We have used random search and grid search as tuning techniques and also used five parameters to be tuned in xgboost model, namely, learning_rate, gamma, max_depth, subsample, and colsample_bytree. Learning_rate is responsible for controlling the step size taken in each boosting iteration , while gamma is responsible in reducing loss to determine the leaf node of a tree (Aryo & Salsa, 2021). The depth of the tree would be determined by max_depth, while e colsample_bytree is the ratio of the attributes when constructing each tree and the fraction of the training data that is randomly sampled for each tree is handled by subsample (Aryo & Salsa, 2021) . For deep learning model specifically convolutional neural network, we used Rectified Linear Unit (ReLU) and sigmoid activation function. **ReLU**: is the simple and the most commonly used function in the CNN context. It returns the input values if it is positive numbers else convert the input to zero (Alzubaidi, et al., 2021). The main advantage of ReLU is Lower computational. ReLU mathematical representation is as follows.

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

sigmoid: is an activation function that is a smooth, S-shaped curve that maps any real-valued number to a value between 0 and 1 (Alzubaidi, et al., 2021). And mathematically represented as following.

$\sigma(x)=1+e-x$

Dropout regularization was utilized for the purpose of preventing overfitting by ensuring that no units are codependent with one another in the deep learning model. Adaptive Moment Estimation (Adam) was used as an optimizer in the deep learning model. And it is one of the widely used optimization techniques. It combines the advantages of both momentum and Root Mean Squared Propagation (RMSprop). It has reduced oscillation, a more smoothed path, and adaptive learning rate capabilities.

4.8. Performance Evaluation Technique

Performance evaluation is the process of measuring the effectiveness of the machine learning models on the given dataset. It involves using different metrics to measure how well the model performs on a given dataset. We have employed different metrics for evaluating the machine learning models in predicting under-five mortality using health, socio-demographic, and

climate data, such as accuracy, precision, recall, f1-score, and Area Under Receiving Operating Characteristic (AUC-ROC). The detailed explanation of all metrics with a formula for evaluation of machine learning models is described in Chapter 3, Section 3.2.5.

CHAPTER 5
5. Experiment and Result Discussion

5.1. Overview

In this chapter, the results obtained and experimental evaluation of different machine learning models on the prediction of under-five mortality on health, socio-demographic, and climate data would be demonstrated. The process of using the artifact to solve one or more instances of the problem is demonstration (PEFFERS, TUURE, ROTHENBERGE, & SAMIR, 2007). The dataset used, description, implementation, results, evaluations, and discussion of the proposed model are discussed in detail. Finally, test results obtained would be presented and compared with other state-of-the-art models.

5.2. Dataset Preparation

In this study, the researchers used health, socio-demographic dataset from Hararghe Demographic Surveillance System (HDSS) and climate data of three sites (Harar, Haramaya and Kersa sites). Health Demographic Surveillance System (HDSS) is a longitudinal population-based health and vital event registration system that monitors demographic and health events in a geographically defined population with timely production of quality data. For evidence generation, specifically in rural areas and countries with weak vital event registration, HDSS is considered to be one of the best sources of representative data (Dheresa, et al., 2022). The two datasets were integrated based on the common variable, as the climate data provided was on site level and monthly based for each year. In this study, a total of 44,933 children were included, and out of 44,933 children, 42,406 children are low risk (coded as 0) and 2,227 children high risk (coded as 1).

Summary of the dataset used is presented in a table below.

Gender	Low risk	High risk	Total size
Male	22,504	1,202	23,706
Female	20,202	1,025	21,227
Total	42,706	2,227	44,933

Table 5- 1 Size of the dataset

On this research work, the researcher used a total of 22 independent features and the description of each variables are given in the below Table.

No	Variable name	Variable description	Data type
1	c_gender	Gender of the child	bool
2	wealthindex	Wealth index of the household	categorical
3	site	Site of the child residence	categorical
4	TMPMIN	Minimum Temperature	numerical
5	TMPMAX	Maximum Temperature	numerical
6	Humidity	Humidity	numerical
7	Precipitation	Precipitation	numerical
8	physicly_normal	Whether the child is physically normal during born	bool
9	weight_sizeofbabycompared	Weight or size of the baby	categorical
10	birthplace	Where did the birthing take place?	categorical
11	occupation	Occupation of the mother	categorical
12	attendant_atbirth	Attendant at birth	categorical
13	preg_duration	Duration of pregnancy	categorical
14	ATTENDED_ANC	Did the mother attend ANC clinics during your pregnancy	bool
15	mother_ageatfirstdeliv	Mother age at first delivery	numerical
16	Ageatbirth	Mother age at current birth	numerical
17	gradecomp	Grade completed by mother	numerical
18	litracy	Mother Literacy	categorical
19	physcly_healthy	was the child Physically Health	bool
20	preciding_childalive	Preceding child alive	bool
21	TOTAL_NUMBER_CHILDREN_STILL_LIVING	Total number of children living	numerical
22	gravidity	Number of pregnancy	numerical

Table 5- 2 Description of the variables

5.3. Handling Missing Values

Missing values are one of the most common problems in the real-world dataset. Missing values are the values not stored or captured for some variables of interest. In this study, the under-five datasets (health, socio-demographic dataset and climate dataset) contain columns with missing values, including occupation, litracy, gradecomp, mother_ageatfirstdeliv, preg_duration, wealthindex, physcly_healthy, and physicly_normal. The missing values that occurred in this dataset are presented in the table below.

Variables/features	Total missing values	Missing values in %
gradecomp	2120	4.78%
litracy	1617	3.64%
occupation	1456	3.28%
wealthindex	542	1.22%
mother_ageatfirstdeliv	83	0.19%
preg_duration	59	0.13%
physicly_normal	4	0.01%
physcly_healthy	2670	6.01%

Table 5- 3 Missing Values

Handling missing values is one of the processes in data preparation or preprocessing step in machine learning. The accuracy and efficiency of machine learning algorithms are based on the quality of data utilized for the analysis. Therefore, missing values should be handled to help machine learning algorithms to perform optimally. The missing values that occurred in the health, socio-demographic dataset are missing completely at random. Thus, the researchers used iterative imputer impute the missing values in the dataset. Figure 5-1 shows handling missing values using iterative imputer.

```
In [62]: from sklearn.experimental import enable_iterative_imputer
         from sklearn.impute import IterativeImputer
         imputer = IterativeImputer(random_state=42)
         iteratedf[missedcolumns] = imputer.fit_transform(iteratedf[missedcolumns])
```

Figure 5- 1 Handling missing values

5.4.Exploratory Data Analysis (EDA)

Exploratory data analysis is the data analytics process that helps to better understand the data in depth and learn its different characteristics, with some statistics and visual representations (Poian, et al., 2023). EDA is a tool used to understand the dataset in depth and is an initial and fundamental step to discover patterns from the data that could potentially be used to develop machine learning algorithms to predict our dataset better. In this study, we used exploratory data analysis to further discover the patterns and understand the nature of the datasets before developing any machine learning algorithms. We have used some statistics and visualization tools to understand and summarize data. The researcher's, commonly utilized Python libraries like pandas, NumPy, Matplotlib and Seaborn for this purpose.

Figure 5-2 presents the distribution of child sex by the survival status. It shows that out of high risk children, 2.66% were male in gender.

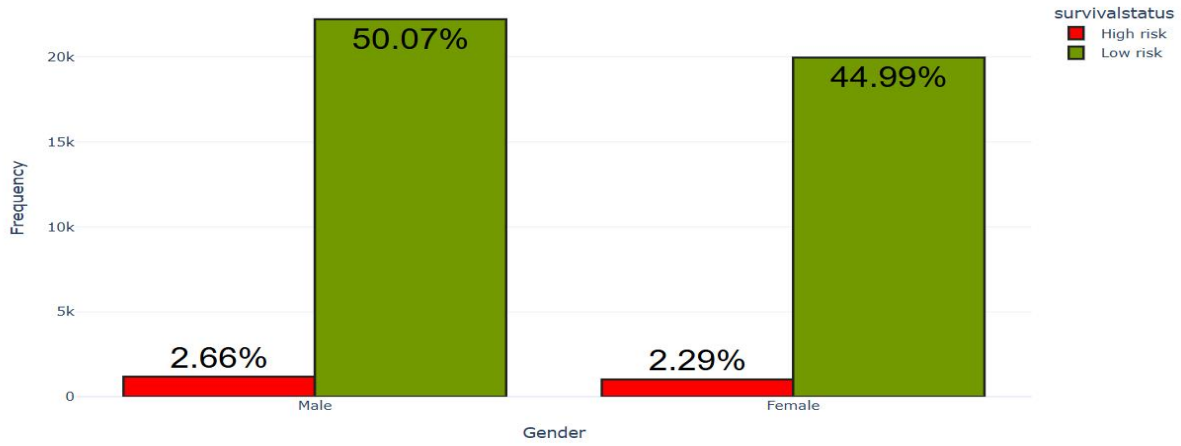


Figure 5- 2 Distribution of child sex by survival status

Figure 5-3 describes the distribution of ANC usage by mother to the low risk and high risk children. It shows that out of the high risk children, 3.16% were at high risk because the children's mothers did not receive the ANC.

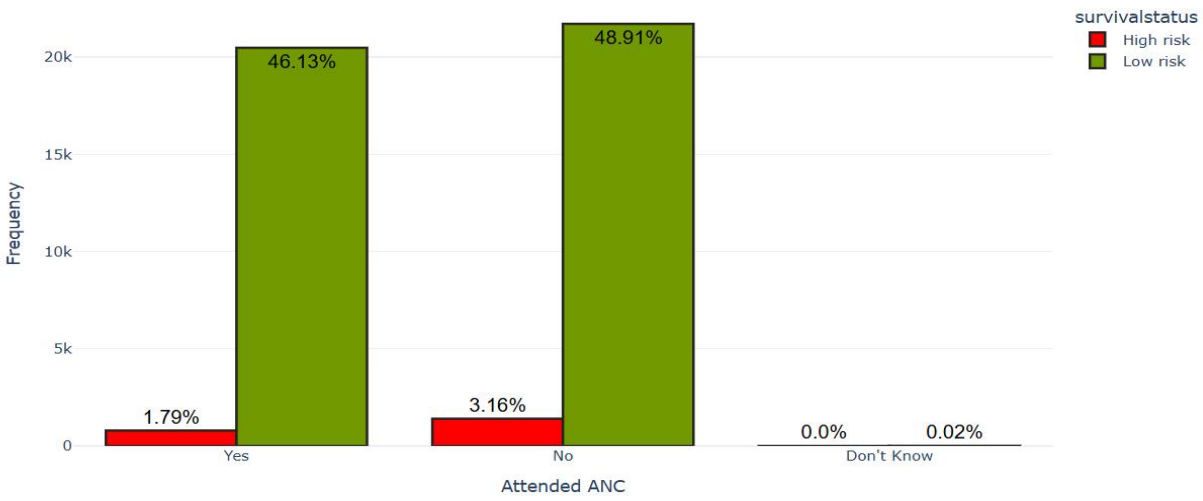


Figure 5- 3 Distribution of ANC usage

Figure 5-4 shows the distribution of children's birthplace to the low risk and high risk children. The majority of the high risk of deaths occurred among children born at home. Out of the children who are at high risk of death, 3.6% were born at home.

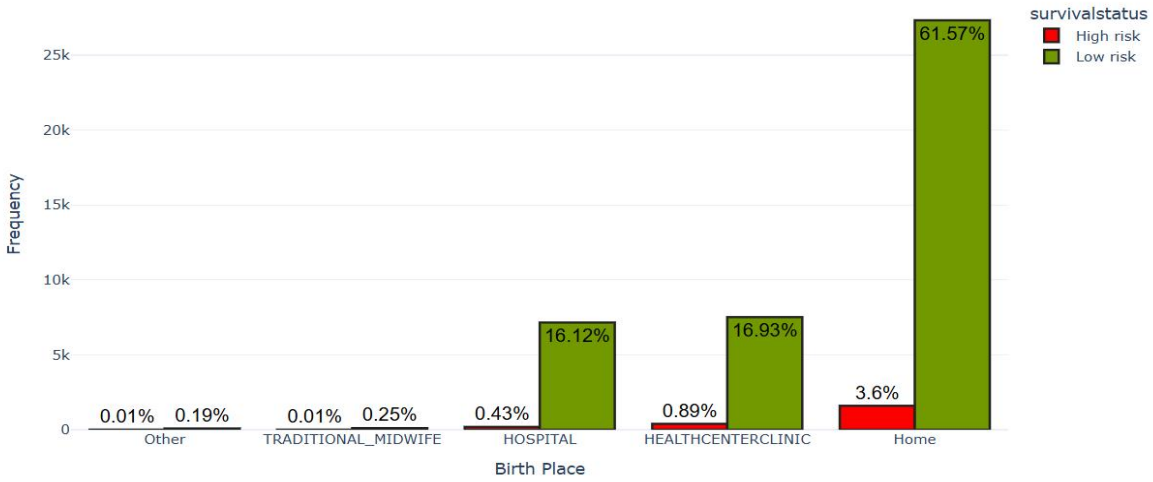


Figure 5- 4 Distribution of children birth place by survival status

5.5. Handling Imbalanced dataset

Imbalance dataset is when the sample of one class is higher than the sample of other class. So if you train the model with an imbalanced dataset, the model would be biassed toward the majority class because machine learning models learn from the examples, and most of the examples in your dataset belong to a single majority class. In this study, the number of low risk children is higher than the number of high risk children. This can lead to biased models that perform poorly on the minority class which is high risk children. Handling imbalanced dataset is very important to improve the accuracy of the model in the prediction of under-five mortality. After removing the duplicates, we addressed the imbalanced dataset on the unique dataset. We have used different data balancing techniques such as SMOTE, ADASYN, and SMOTE+TOMEKlinks. Table 5-4 presents the summary of the size of the imbalanced dataset and balanced dataset with different techniques.

Imbalanced and balanced	Low risk	High risk	Total
Imbalanced dataset	42,197	2,195	44,392
Balanced by SMOTE	42,197	42,197	84,394
Balanced by ADASYN	42,197	42,835	85,032
Balanced by SMOTE+TOMEKlinks	42,197	42,149	84,346

Table 5- 4 imbalanced and balanced dataset size

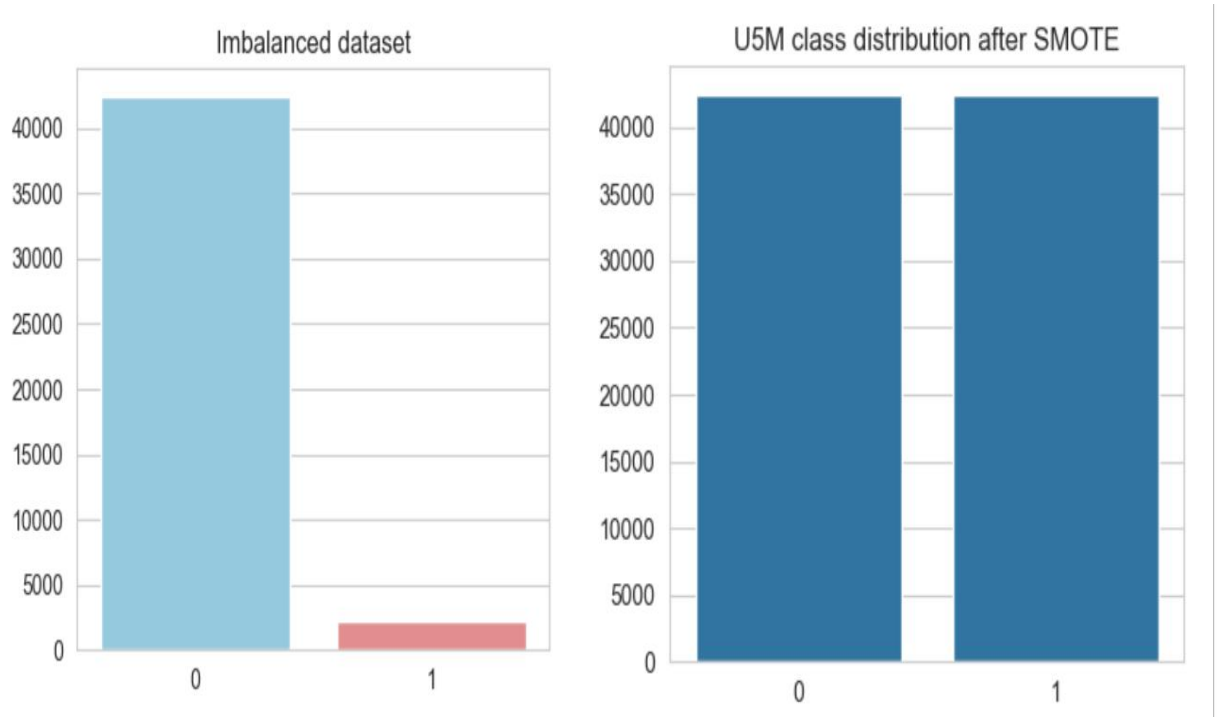
The code for handling imbalanced dataset is shown in the figure 5-5

```
In [100]: #The function that balance the dataset using all techniques and return balanced dataset.
def data_balance(X,Y):
    #SMOTE DATA BALANCING
    smt=SMOTE(random_state=42)
    x_smoteresampled,y_smoteresampled=smt.fit_resample(X,Y)
    #plot_resampled_result(y_smoteresampled,'U5M class distribution after SMOTE')

    #ADASYN
    adasyn=ADASYN(random_state=42)
    x_adasynresampled,y_adasynresampled=adasyn.fit_resample(X,Y)
    #plot_resampled_result(y_adasynresampled,'U5M class distribution after ADASYN')
    #TOMEK LINK
    tom=TomekLinks()
    x_tomresampled,y_tomresampled=tom.fit_resample(X,Y)
    #plot_resampled_result(y_tomresampled,'U5M class distribution after TomeLinks')
    #SMOTE+TOMELINKS
    tomlink=TomekLinks()
    x_smotetomelinkresampled,y_smotetomelinkresampled=tomlink.fit_resample(x_smoteresampled,y_smoteresampled)
    #plot_resampled_result(y_smotetomelinkresampled,'U5M class distribution after SMOTE+TOMELINKS')
    #Random Over Sampling
    ros=RandomOverSampler(random_state=42)
    x_rosresampled,y_rosresampled=ros.fit_resample(X,Y)
    #plot_resampled_result(y_rosresampled,'U5M class distribution after ROS')
    # Random Under Sampling
    rus=RandomUnderSampler(random_state=42)
    x_rusresampled,y_rusresampled=rus.fit_resample(X,Y)
    #plot_resampled_result(y_rusresampled,'U5M class distribution after RUS')
    return x_smoteresampled,y_smoteresampled,x_adasynresampled,y_adasynresampled,x_tomresampled,y_tomresampled,x_rosresampled,y_r
```

Figure 5- 5 code for dataset balancing

Once we run the above code to handle the imbalanced dataset, then the dataset will get balanced. Figure 5-6 a, presents the imbalanced dataset or the original dataset and Figure 5-6 b, shows the balanced dataset after we applied the SMOTE data resampling method to balance the representation of the minority class with that of the majority class.



a b

Figure 5- 6 a) Represent imbalanced dataset b) presents balanced dataset

5.6. Implementation and experimental result

In this research work, python programming language with different packages is used in the process of experimenting this study. Python is an object-oriented, interpreted, mid-level programming language that is easy to learn and use while being versatile enough to tackle a variety of tasks. Python version 3.12.4 with Anaconda distribution is used in this research. Anaconda is a free software that gives with a toolkit that is tailored for research and science. The python libraries or packages that have been used includes NumPy, Pandas, Matplotlib, SciPy, Scikit-learn, TensorFlow and Keras. NumPy is a library for python that solves scientific computation easily. Pandas is an open-source library that is used to read CSV files

and perform different operations on the CSV files. Scikit-learn is an open-source machine learning library for Python. Which supports a wide range of tools for building and training various machine learning models. TensorFlow is an open-source deep learning library that provides a flexible framework for building and training neural networks, especially for tasks like image and text recognition. Keras is a high-level neural network library that runs on top of TensorFlow, offering a user-friendly API for building and training deep learning models (Damien , Matt, Thadd´e, & Kinsey, 2020).

Figure 5-7 shows the important libraries importing for preprocessing and building or developing machine learning algorithms.

```
import pandas as pd
import pymysql
import numpy
from sqlalchemy import create_engine
import warnings
warnings.filterwarnings('ignore')
%load_ext sql

from scipy.stats import randint, uniform
from sklearn.feature_selection import SequentialFeatureSelector

from sklearn.model_selection import cross_val_predict
import numpy as np
from sklearn.model_selection import StratifiedKFold,RandomizedSearchCV, cross_val_score
import matplotlib.pyplot as plt
import seaborn as sns
import xgboost as xgb
from sklearn.model_selection import train_test_split, GridSearchCV
from sklearn.metrics import confusion_matrix, ConfusionMatrixDisplay, classification_report
from sklearn.metrics import balanced_accuracy_score, accuracy_score, precision_score, recall_score, f1_score
from sklearn.utils.class_weight import compute_sample_weight
from sklearn.ensemble import RandomForestClassifier
from sklearn.svm import SVC
from sklearn.linear_model import LogisticRegression
from sklearn.impute import SimpleImputer
from sklearn.preprocessing import OneHotEncoder, LabelEncoder
import itertools
import joblib
import sys
sys.modules['sklearn.externals.joblib'] = joblib
%matplotlib inline
```

Figure 5- 7 Important libraries imported for building machine learning algorithms

As once the necessary libraries are imported, then we loaded data from the database and saved it as a CSV file. The CSV file is loaded as a Data frames from the disk using pandas as shown in Figure 5-8. The pandas method read_csv reads the file in the tab separated file and load it as DataFrames, where instances of the data are accessible via column name.

```
df=pd.read_csv("C:/Users/fgelashe/paperunderfive/Feyisathesis/underfiveclimatecleanddatafinal.csv")
```

Figure 5- 8 To load csv file from disk

After the data is loaded as a DataFrames, the next step is to preprocess the data which involves removing duplicates, handling missing values, handling imbalanced dataset, and changing the

data to some format that machine learning models require (changing categorical values to numerical). To check and handle the duplicates we have used the code shown in Figure 5-9 and to change categorical variable to numerical, we have used label encoder and replace function. Handling missing values and imbalanced dataset is already discussed in detail in the above section.

```
df.duplicated().sum()
df=df.drop_duplicates()
```

Figure 5- 9 Code for checking and removing duplicates

once the dataset is loaded, preprocessed and balanced, the next step would be splitting the dataset into training and testing using train_test_split function from scikit-learn library. The dataset is split with the ratio of 80% for training and 20% for testing, the code to split the dataset into training and testing is shown in the Figure 5-10.

```
X_train,X_test,y_train,y_test=train_test_split(X_resampled, y_resampled,test_size=0.2,random_state=42,stratify=y_resampled)
```

Figure 5- 10 Presents the code for dataset split

After that, eight machine learning models (Naïve Bayes, Support Vector Machine, Decision Tree, Random Forest, K-Nearest Neighbor, Extreme Gradient Boosting, convolutional neural network, and TabNet) trained using training dataset and evaluated using testing dataset.

For implementing the Naïve Bayes, we used MultinomialNB() function of sklearn, as it suitable for categorical and discrete features. The following figure shows the implementation of Naïve Bayes.

```
nb_classifier = MultinomialNB()
# Train the classifier
nb_classifier.fit(X_train, y_train)
y_pred = nb_classifier.predict(X_test)
```

Figure 5- 11 Naive Bayes model

We have used SVC() function of the sklearn package for building the SVM model. Figure 5-12 shows the implementation of the support vector machine.

```
model_svm=SVC(kernel='rbf',random_state=42)
model_svm.fit(X_train,y_train)
svm_pred=model_svm.predict(X_test)
```

Figure 5- 12 SVM model

We used KNeighborsClassifier() function of the sklearn package for implementing the KNN model. The figure 5-13 shows the implementation of the K-Nearest Neighbor.

```
classifier_knn=KNeighborsClassifier(metric='minkowski',p=2)
classifier_knn.fit(X_train,y_train)
knn_pred=classifier_knn.predict(X_test)
```

Figure 5- 13 KNN model

DecisionTreeClassifier() is function of sklearn package is used for building the decision tree model.

Figure 5-14 shows the implementation of Decision Tree model.

```
def dt_modelpredict(X_train,X_test,y_train,y_test):
    dt_model = DecisionTreeClassifier(criterion='entropy', random_state=42)
    param_grid = {
        'max_depth': [1,3, 5,7,9,11],
        'min_samples_split': [2, 5, 10],
        'min_samples_leaf': [1, 2, 4],
        'criterion': ['gini', 'entropy']
    }
    grid_search = GridSearchCV(dt_model, param_grid, cv=5)
    grid_search.fit(X_train, y_train)
```

Figure 5- 14 Decision Tree model

For implementing Random Forest, we used RandomForestClassifier() function of sklearn. Figure 5-15 shows the implementation of random forest model.

```
rf_model = RandomForestClassifier(criterion='entropy',random_state=42)
grid_search = GridSearchCV(rf_model, param_grid, cv=5)
grid_search.fit(X_train, y_train)
```

Figure 5- 15 Random Forest model

We used XGBClassifier() function of the XGBoost package to implement Extreme Gradient Boosting model. Figure 5-16 shows the implementation of Extreme Gradient Boosting.

```
xgb_model=xgb.XGBClassifier(n_estimators=xgb_bp["n_estimators"],
                            max_depth=xgb_bp["max_depth"],
                            learning_rate=xgb_bp["learning_rate"],
                            subsample=xgb_bp["subsample"],
                            colsample_bytree=xgb_bp["colsample_bytree"],
                            gamma=xgb_bp["gamma"],
                            )
xgb_model.fit(X_train, y_train)
y_pred = xgb_model.predict(X_test)
```

Figure 5- 16 Extreme Gradient Boosting

We used TabNetClassifier() function of the pytorch TabNet to build the TabNet model. Figure 5-14 shows the implementation of the TabNet model.

```
clf= TabNetClassifier(
                    scheduler_params={"step_size":20,
                                      "gamma":0.9},
                    scheduler_fn=torch.optim.lr_scheduler.StepLR,
                    mask_type='entmax'
                   )
```

Figure 5- 17 TabNet model

Sequential() function of the TensorFlow package is used for implementing the convolutional neural network. Figure 5-18 shows the implementation of Convolutional Neural Network.

```
modelt = Sequential()
modelt.add(Conv1D(32, 3, activation='relu', input_shape=(X_train.shape[1], 1)))
modelt.add(MaxPooling1D(2))
modelt.add(Flatten())
modelt.add(Dense(128, activation='relu'))
modelt.add(Dropout(0.5))
modelt.add(Dense(1, activation='sigmoid'))
```

Figure 5- 18 Convolutional Neural Network

5.7. Experimental Results

In this experimental test, we assessed the results and impacts of using and not using balanced datasets for the prediction of under-five mortality. To answer the research questions, we have conducted two experiments, one with the balanced datasets and the other with the imbalanced datasets. Then, the results were compared to determine which technique is effective or more accurate in predicting under-five mortality. As already discussed, missing values were handled for both experiments, and the data was split into 80% of training, which was later on used for 5-fold cross-validation to tune the model parameters for optimal performance and 20% of testing data. The evaluation metrics used were accuracy, precision, recall, F1-score, AUC-ROC score. Macro average, and weighted-average were calculated for each of the listed metrics.

3.7.2. Experiment One

In this experiment, the models were trained using the imbalanced datasets. Imbalanced data is a common problem in supervised machine learning and deep learning where there is a non-uniform distribution among the samples. In this research work, the number of low risk cases outnumbered the number of high risk cases, so the majority of the data belong to low risk child cases. The accuracy rate of the under-five mortality prediction would be determined in the absence of the techniques used to handle imbalanced data. This would provide useful information on the performance of the model using an imbalanced dataset and would assist the researchers in understanding the consequences of using the imbalanced dataset on the prediction of under-five mortality.

5.7.1.1. Naïve Bayes model with imbalanced datasets

The classification report below shows the performance of Naïve Bayes model in prediction of under-five mortality using imbalanced dataset.

NB:

	precision	recall	f1-score	support
0.0	0.97	0.33	0.49	8440
1.0	0.06	0.84	0.11	439
accuracy			0.35	8879
macro avg	0.52	0.58	0.30	8879
weighted avg	0.93	0.35	0.47	8879

Figure 5- 19 The Classification report of NB model on imbalanced dataset.

From the results shown in Figure 5-19, the NB model obtained an overall accuracy score of 35%. So the overall accuracy obtained was too low, and this indicates that the model poorly generalize on unseen datasets. The model achieved the precision score of 97% for the low risk children and 6% precision were for the high risk children; this shows if the dataset is imbalanced, the precision for the majority class would be high, and this is because the model is biassed toward predicting the majority class. The model achieved precision of 93%, recall of 35%, f1-score of 47%, and AUC-ROC score of 61%. Table 5-5 shows the confusion matrix of the NB model. In the first row of table 5-20, there were a total of 8,440 test datasets of the low risk children. Out of these 8,440 children, only 2,750 children were correctly predicted as the low risk children, while 5,690 cases were incorrectly predicted as the high risk children.

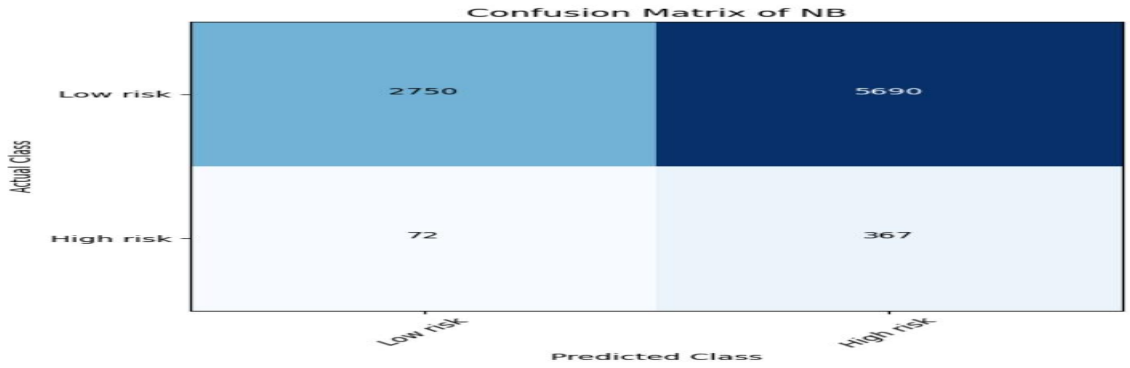


Table 5- 5 Confusion matrix of NB for imbalanced datasets

5.7.1.2. Support Vector Machine model with imbalanced datasets

The classification report in Figure 5-20 shows the performance of SVM model in prediction of under-five mortality using imbalanced datasets.

SVM:

	precision	recall	f1-score	support
0.0	0.95	1.00	0.98	8440
1.0	0.96	0.06	0.11	439
accuracy			0.95	8879
macro avg	0.96	0.53	0.54	8879
weighted avg	0.95	0.95	0.93	8879

Figure 5- 20 Classification report of SVM model using imbalanced datasets

From the result shown in Figure 5-20, the SVM model obtained an overall accuracy of 95%. And 100% recall for the children at the low risk and 6% recall for the children at the high risk, this shows the model is totally biassed toward the majority class. The SVM model achieved 95%, 95%, 93%, and 69% for the precision, recall, F1-score, and AUC-ROC scores, respectively. This indicates that the model is struggling to predict under-five mortality because of the lower AUC-ROC scores. Table 5-21 shows the confusion matrix of the SVM model. In the second row of table 5-6, there were a total of 439 test datasets of the high risk cases. Out of 439 cases, only 26 cases were correctly predicted as the high risk children, while 413 cases were incorrectly predicted as the low risk children.

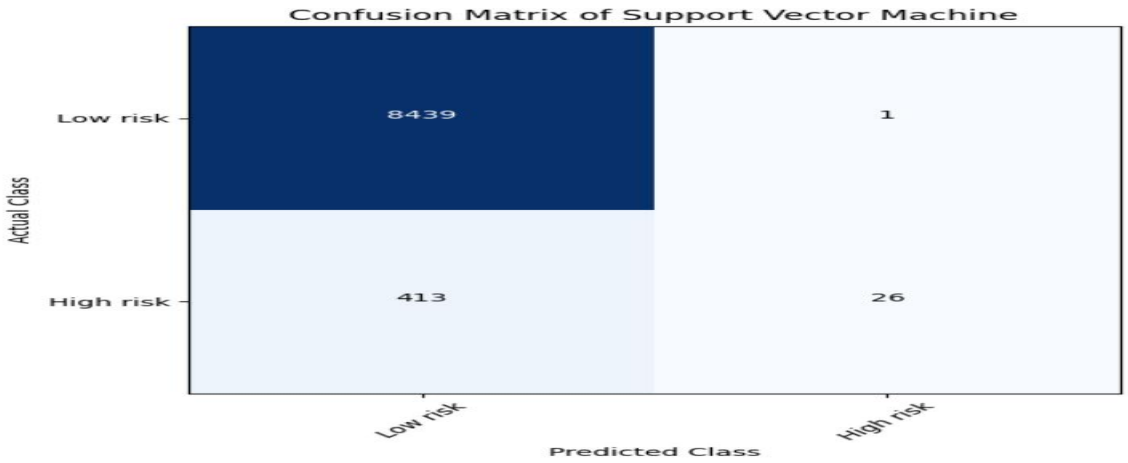


Table 5- 6 Confusion matrix of SVM model using imbalanced datasets

5.7.1.3. K-Nearest Neighbors model with imbalanced datasets

The classification report in Figure 5-21 shows the performance of KNN model in prediction of under-five mortality using imbalanced datasets.

KNN:

	precision	recall	f1-score	support
0.0	0.95	1.00	0.97	8440
1.0	0.24	0.01	0.03	439
accuracy			0.95	8879
macro avg	0.60	0.51	0.50	8879
weighted avg	0.92	0.95	0.93	8879

Figure 5- 21 Classification report of KNN model using imbalanced datasets

From the results shown in the Figure 5-21, the KNN model yields an overall accuracy of 95%. And the low risk f1-score of 97% and the high risk f1-score of 3%, which indicates the model is biassed toward predicting the majority class. The model achieved precision of 92%, recall of 95%, F1-score of 93%, and AUC-ROC score of 57%. Table 5-7 shows the confusion matrix of the KNN model. In the second row of table 5-22, there were a total of 439 test datasets of the high risk cases. Out of these 439 cases, only 6 cases were correctly predicted as the high risk children, while 433 cases were incorrectly predicted as the low risk children.

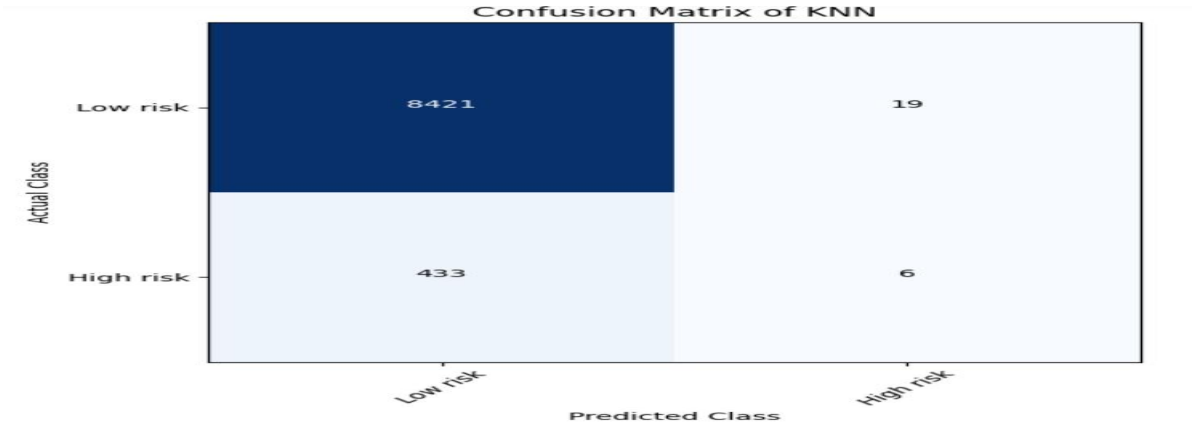


Table 5- 7 Confusion matrix of KNN model using imbalanced datasets

5.7.1.4. Decision Tree model with imbalanced datasets

The classification report in Figure 5-22 shows the performance of DT model in prediction of under-five mortality using imbalanced datasets.

Decision Tree:

	precision	recall	f1-score	support
0.0	0.96	1.00	0.98	8440
1.0	0.96	0.26	0.40	439
accuracy			0.96	8879
macro avg	0.96	0.63	0.69	8879
weighted avg	0.96	0.96	0.95	8879

Figure 5- 22 Classification report of DT model using imbalanced datasets

The results shown in Figure 5-22 show that the DT model obtained an overall accuracy of 96%. And the precision of 96%, recall of 96%, F1-score of 95%, and AUC-ROC of 75%. This implies that, even though the accuracy obtained by the model is impressive, the model is somehow struggling for the prediction of under-five mortality because of the lower AUC-ROC score. The model achieved moderate performance compared to the above-discussed models in terms of AUC-ROC. The table 5-8 shows the confusion matrix of the DT model. For instance, in the second row of table 5-8, there were a total of 439 test datasets of the high risk cases. Out of these 439 cases, only 112 cases were correctly predicted as the high risk cases, while 327 cases were incorrectly predicted as the low riskcases.

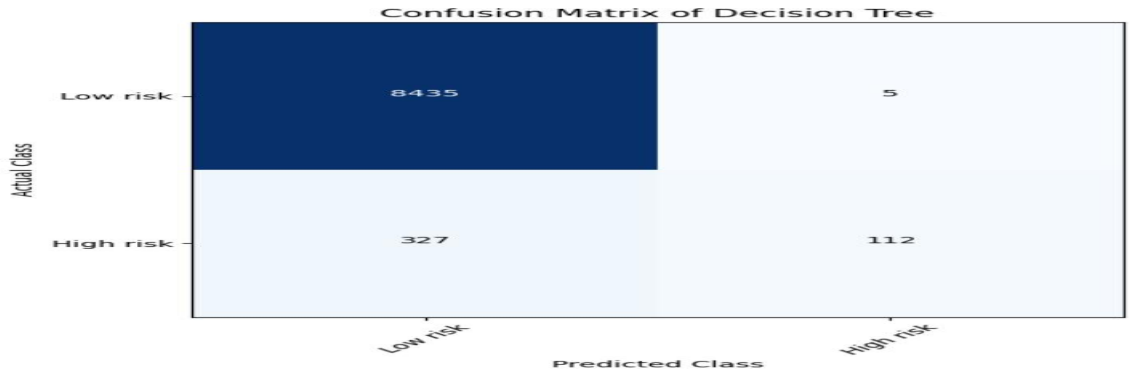


Table 5- 8 Confusion matrix of DT model using imbalanced dataset

5.7.1.5. Random Forest model with imbalanced datasets

The classification report in Figure 5-23 shows the performance of RF model in prediction of under-five mortality using imbalanced datasets.

Random Forest:

	precision	recall	f1-score	support
0.0	0.96	1.00	0.98	8440
1.0	0.96	0.26	0.41	439
accuracy			0.96	8879
macro avg	0.96	0.63	0.70	8879
weighted avg	0.96	0.96	0.95	8879

Figure 5- 23 Classification report of RF model using imbalanced datasets

The results in Figure 5-23 shows that the RF model obtained the overall testing accuracy score of 96%. And F1-score for the low risk is 98% and F1-score for the high risk is 41%, this indicate the model is predicting 98% for the low risk and only 41% for the high risk cases and there is big fluctuation between the low risk and high risk cases. Which implies as the model is biased toward the predicting of majority class. The RF model achieved 96%, 96%, 95%, and 78% for precision, recall, F1-scores and AUC-ROC score, respectively. This shows that the model moderately effective in prediction of under-five mortality on imbalanced datasets. The table 5-9 shows the confusion matrix of RF model. For example in the second row of table 5-9, there were a total of 439 test datasets of the high risk cases. Out of these 439 cases, only 116 cases were correctly predicted as the high risk cases, while 323 cases were incorrectly predicted as the low risk cases.

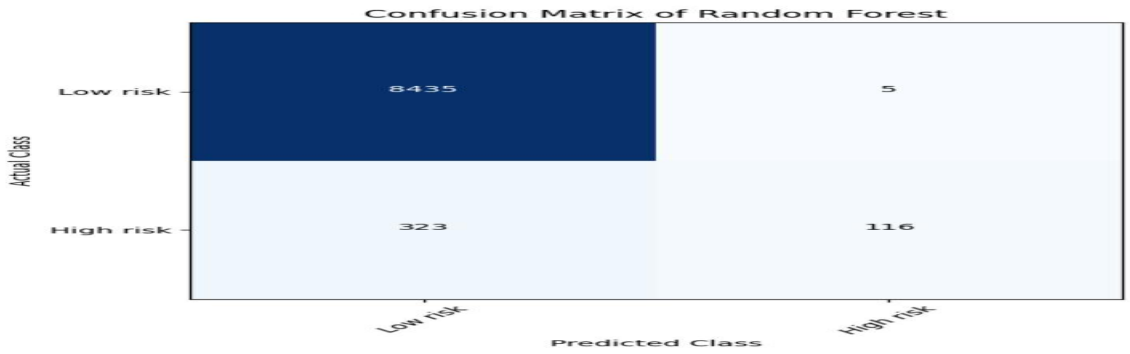


Table 5- 9 Confusion matrix of RF model using imbalanced datasets.

5.7.1.6. Extreme Gradient Boosting model with imbalanced datasets

The classification report on Figure 5-24 shows the performance of XGBoost model in prediction of under-five mortality using imbalanced datasets.

	precision	recall	f1-score	support
0.0	0.96	1.00	0.98	8440
1.0	0.95	0.26	0.41	439
accuracy			0.96	8879
macro avg	0.96	0.63	0.70	8879
weighted avg	0.96	0.96	0.95	8879

Figure 5- 24 Classification report XGBoost model using imbalanced datasets.

As results shown in Figure 5-24, the XGBoost model on imbalanced datasets yield the overall accuracy of 96%. This indicates that the model shows the significant performance on the prediction of under-five mortality using imbalanced data. The XGBoost model achieved, 96%, 96%, 95%, and 80% for precision, recall, F1-scores, and AUC-ROC, respectively. The AUC-ROC metrics is the best suit metrics for the imbalanced datasets because it is not significantly affected by imbalanced datasets where the instance in one class outnumbered the other class. Thus, compared with the models discussed above in experiment two, the XGBoost model performs better than the other models in terms of AUC-ROC score in prediction of under-five mortality. The table 5-10 shows the confusion matrix of XGBoost model. For example in the second row of table 5-10, there were a total of 439 test datasets of the high risk cases. Out of these 439 cases, 116 cases were correctly predicted as the high risk cases, while 323 cases were incorrectly predicted as the low risk cases.

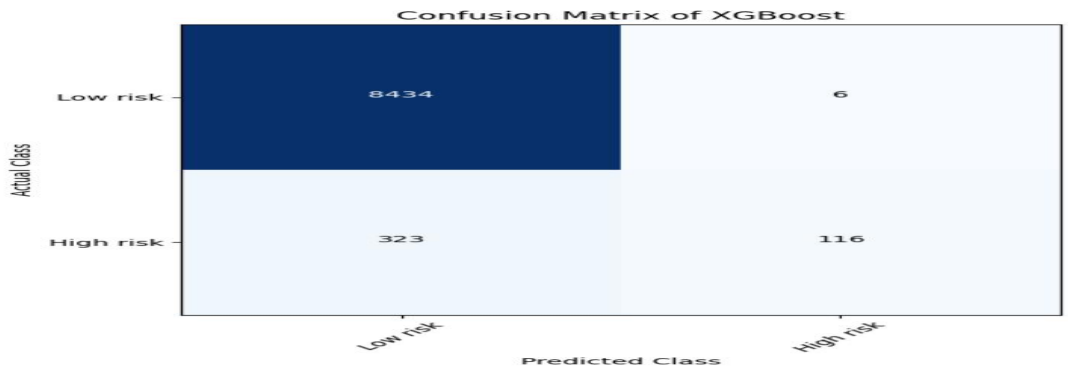


Table 5- 10 Confusion matrix of XGBoost model using imbalanced datasets.

5.7.1.7. TabNet model with imbalanced datasets

The classification report on figure 5-25 shows the performance of TabNet model in prediction of under-five mortality using imbalanced datasets.

	precision	recall	f1-score	support
0.0	0.98	0.60	0.75	8440
1.0	0.09	0.72	0.15	439
accuracy			0.61	8879
macro avg	0.53	0.66	0.45	8879
weighted avg	0.93	0.61	0.72	8879

Figure 5- 25 Classification report of TabNet model using imbalanced datasets.

As results shown in Figure 5-25, the TabNet model obtained the overall accuracy of 61%, this indicates that the accuracy obtained is low so the model may generalize poorly on unseen data. The TabNet model achieved 93%, 61%, 72%, and 75% for the precision, recall, F1-score and AUC-ROC scores respectively. The model achieved the low risk precision score of 98% and the high risk precision score of 9%, so this indicates that the model is biased toward the majority class. The Table 5-11 shows the confusion matrix of TabNet model. For example in the second row of table 5-11, there were a total of 439 test datasets of the high risk cases. Out of these 439 cases, 316 cases were correctly predicted as the high risk cases, while 123 cases were incorrectly predicted as the low risk cases.

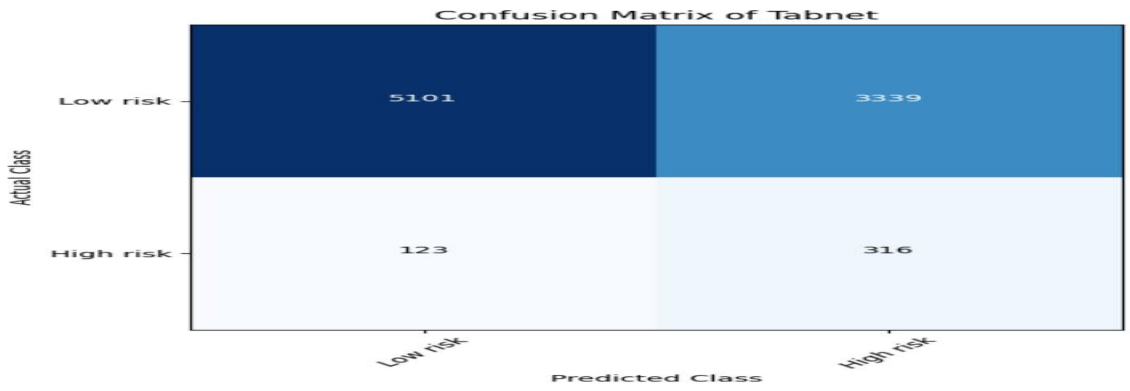


Table 5- 11 Confusion matrix of TabNet model using imbalanced datasets.

5.7.1.8. Convolutional Neural Network model with imbalanced datasets

The CNN model's performance in predicting under-five mortality is presented in Figure 5-26. This classification report demonstrates the model's accuracy and effectiveness when dealing with imbalanced datasets.

	precision	recall	f1-score	support
0.0	0.96	1.00	0.98	8434
1.0	0.91	0.23	0.37	445
accuracy			0.96	8879
macro avg	0.94	0.62	0.68	8879
weighted avg	0.96	0.96	0.95	8879

Figure 5- 26 Classification report of CNN model using imbalanced datasets.

As results shown in Figure 5-26, the CNN model obtained an overall accuracy of 96%. And the model achieved the precision of 96%, recall of 96%, F1-scores of 95%, and AUC-ROC of 80%. This implies that the model achieved promising accuracy to generalize well on unseen datasets, but the AUC-ROC score indicates that the model is moderately effective in identify positive and negative cases in the prediction of the under-five mortality. Table 5-12 shows the confusion matrix of the CNN model. For example, in the second row of table 5-12, there were a total of 445 test datasets of the high risk cases. Out of these 445 cases, only 104 case was correctly predicted as the high risk cases, while 341 cases were incorrectly predicted as the low risk cases.

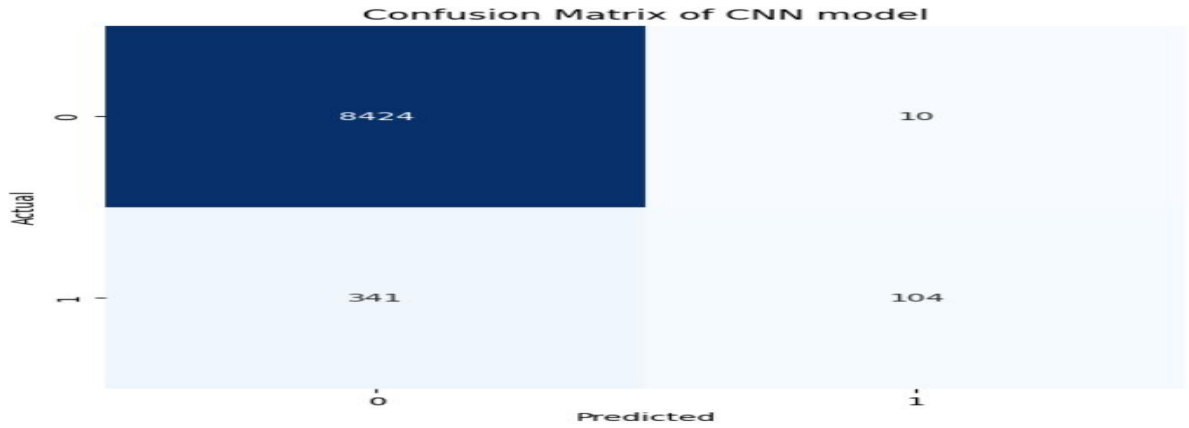


Table 5- 12 Confusion matrix of CNN model using imbalanced datasets.

5.7.2. Experiment Two

This is experiment is conducted with balanced datasets. As already explained, for this research work we have used three techniques of handling imbalanced dataset, such as SMOTE, ADASYN and SMOTE+TOMEKlinks. This would provide useful insight into the performance of the model while using the balanced dataset and would help the researchers to understand the impacts of these techniques on the model performance in prediction of under-five mortality on the health, socio-demographic, and climate data in eastern Hararghe, Ethiopia.

5.7.2.1.Naïve Bayes

The classification report below shows the performance of Naïve Bayes model in prediction of under-five mortality.

NB:

	precision	recall	f1-score	support
0.0	0.69	0.20	0.32	8440
1.0	0.53	0.91	0.67	8439
accuracy			0.56	16879
macro avg	0.61	0.56	0.49	16879
weighted avg	0.61	0.56	0.49	16879

Figure 5- 27 Performance evaluation of Naive Bayes model.

As shown in Figure 5-27, the NB model obtains an overall accuracy of 55.6% for prediction of under-five mortality. And the NB model achieved 61%, 56%, and 49% for precision, recall,

and F1-score, respectively. This indicates that the model results in lower testing accuracy; the lower testing accuracy suggests that the model may not generalize well on unseen data. The model scored a precision of 61%, which shows that out of all test cases, only 61% were correctly predicted of all positive instances. Table 5-13 presents the confusion matrix, which shows how many cases of all the test dataset were correctly predicted as children at the low risk or high risk of death before the age of five. In the first row, there are a total of 8,440 test datasets of the low risk cases. Out of these 8,440 cases, 1,723 were correctly predicted as the low risk, while 6,717 cases were incorrectly predicted as the high risk. This is due to the data complexity because Naive Bayes assumes that features are independent, which might not hold true for complex real-world problems like the integrated dataset (health, socio-demographic, and climate dataset).

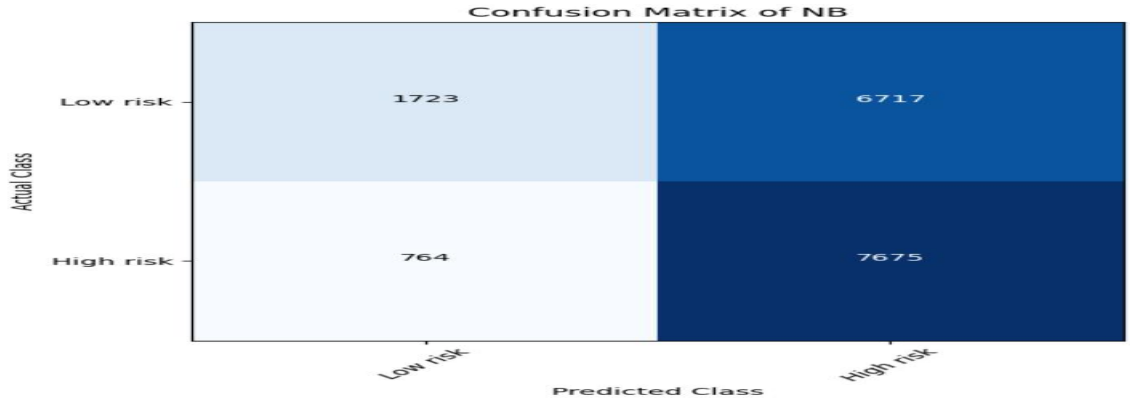


Table 5- 13 Confusion matrix of Naive Bayes model

Table 5-14 shows the results obtained from NB model based on different data balancing techniques.

Performance Evaluation Results of NB Model			
Metrics	SMOTE(%)	ADASYN(%)	SMOTE+TOMEKlinks(%)
Accuracy	56	56	55
Precision	61	61	61
Recall	56	56	55
F1-Score	49	49	49
AUC-ROC	60	60	60

Table 5- 14 Results of NB model

Table 5-14 illustrates the accuracy, precision, recall, F1-score, and AUC-ROC achieved in each of the three dataset balancing techniques for the NB model. And it is evident that NB

with a dataset balanced by SMOTE and ADASYN yields the better results with an accuracy score of 56% and AUC-ROC score of 60%.

5.7.2.2.Support Vector Machine

The classification report below shows the performance of the Support Vector Machine model.

SVM:

	precision	recall	f1-score	support
0.0	0.85	0.95	0.90	8440
1.0	0.94	0.83	0.89	8439
accuracy			0.89	16879
macro avg	0.90	0.89	0.89	16879
weighted avg	0.90	0.89	0.89	16879

Figure 5- 28 Performance evaluation of Support Vector Machine model

From the results shown in Figure 5-28, the SVM model obtained 89% of overall testing of accuracy score. This indicates that the model correctly predicted the under-five mortality 89% of the time on unseen data, and it is evident that the model generalizes well on unseen data. The SVM model performs better than the NB model in the prediction of under-five mortality. The model scored the precision of 90%, recall of 89%, and f1-score of 89%. This indicates that 90% were correctly predicted of all positive instances and 89% were correctly identified of all actual positive instances. Table 5-15 presents the confusion matrix of Support Vector Machine model. There are a total of 8,440 test datasets of the low risk children. Out of these 8,440 the low risk children, 8,000 were correctly predicted as the low risk children, while 440 children were incorrectly predicted as the high risk children in the first row.

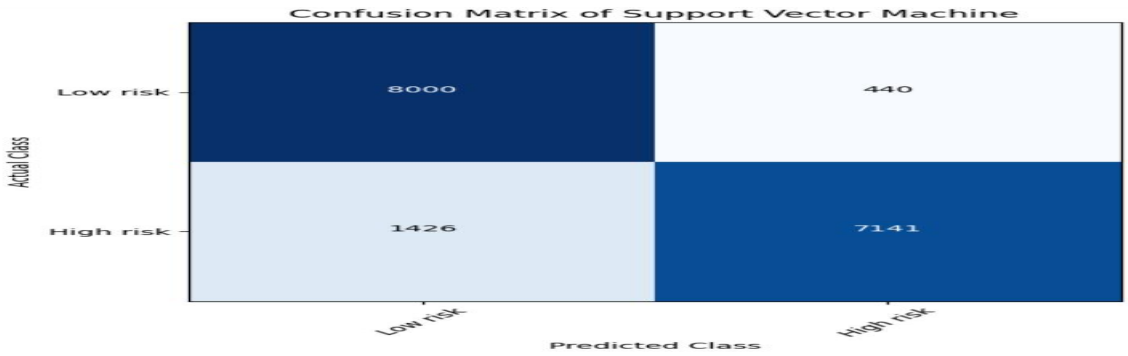


Table 5- 15 Confusion matrix of Support Vector Machine model

Table 5-16 presents the results of SVM model based on different data balancing techniques.

Performance Evaluation Results of SVM Model			
Metrics	SMOTE(%)	ADASYN(%)	SMOTE+TOMEKlinks(%)
Accuracy	89	89	89
Precision	90	90	90

Recall	89	89	89
F1-Score	89	89	89
AUC-ROC	96	96	96

Table 5- 16 Result of SVM model

Table 5-16 illustrates the accuracy, precision, recall, F1-score, and AUC-ROC achieved in each of the three dataset balancing techniques for the SVM model. And it is evident that SVM model with a dataset balanced by all techniques yields the same results with an accuracy score of 89% and a precision score of 90%.

5.7.2.3. K- Nearest Neighbours

The performance of K-NN model is shown by the classification report below.

KNN:

	precision	recall	f1-score	support
0.0	1.00	0.82	0.90	8440
1.0	0.85	1.00	0.92	8439
accuracy			0.91	16879
macro avg	0.92	0.91	0.91	16879
weighted avg	0.92	0.91	0.91	16879

Figure 5- 29 Classification report of KNN model

As results shown in the figure 5-29, the KNN model achieved an overall testing accuracy of 90.9%. This reveals the model moderately generalize well on the unseen data. The KNN model outperformed both of the above models (NB and SVM) in the prediction of under-five mortality. The model scored 92%, 91%, and 91% for precision, recall, and f1-score, respectively. This indicates that the model is able to correctly identify 92% of all predicted positive instances. The confusion matrix for a KNN model is shown in the below table. From Table 5-9 we understand how well the model predicted the cases correctly; there are a total of 8,439 test datasets of the high risk cases. Out of these 8,439 cases, 8,423 were correctly predicted as the high risk cases, while 16 cases were incorrectly predicted as the low risk children in the first row.

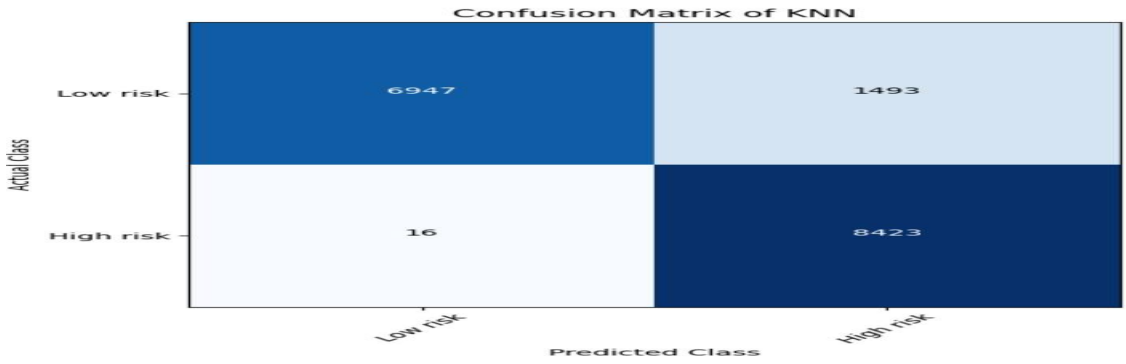


Table 5- 17 Confusion matrix of KNN model

Table 5-10 shows the results of KNN model based on different data balancing techniques

Performance Evaluation Results of KNN Model			
Metrics	SMOTE(%)	ADASYN(%)	SMOTE+TOMEKlinks(%)
Accuracy	91	91	91
Precision	92	92	92
Recall	91	91	91
F1-Score	91	91	91
AUC-ROC	98	98	98

Table 5- 18 Results of KNN model

Table 5-18 illustrates the accuracy, precision, recall, F1-score, and AUC-ROC achieved in each of the three dataset balancing techniques for the KNN model. And it shows that KNN with a dataset balanced by ADASYN yields better results with an accuracy score of 91% and AUC-ROC score of 98%.

5.7.2.4. Decision Tree

The classification report below shows the performance of the Decision Tree model in the prediction of under-five mortality prediction.

Decision Tree:

	precision	recall	f1-score	support
0.0	0.95	0.98	0.96	8440
1.0	0.97	0.95	0.96	8439
accuracy			0.96	16879
macro avg	0.96	0.96	0.96	16879
weighted avg	0.96	0.96	0.96	16879

Figure 5- 30 Classification Report of Decision Tree model

From the results shown in figure 5-30, the decision tree model obtained the overall testing accuracy of 96.2%. This indicates that the model performs well on unseen data in the prediction of under-five mortality. The testing accuracy obtained is better than the models discussed above. The model achieved the precision score of 96%, the recall score of 96% and f1-score of 96%. This shows that 96% were correctly identified of all positive instances. This reveals the model has a higher proportion of true positive predictions compared to false positive predictions. High precision and recall lead to lower false positive and false negative prediction. The table 5-19 shows the confusion matrix of the decision tree model. From the table 5-19, we can see the improvement on how the model predicted the cases correctly and the number of cases predicted incorrectly were decreased. For instance, in the first row there

are a total of 8,440 test datasets of the low risk cases. Out of these 8,440 cases, 8,230 were correctly predicted as the low risk children, while only 210 children were incorrectly predicted as the high risk children. It is evident that both false positive and false negative predictions have decreased.

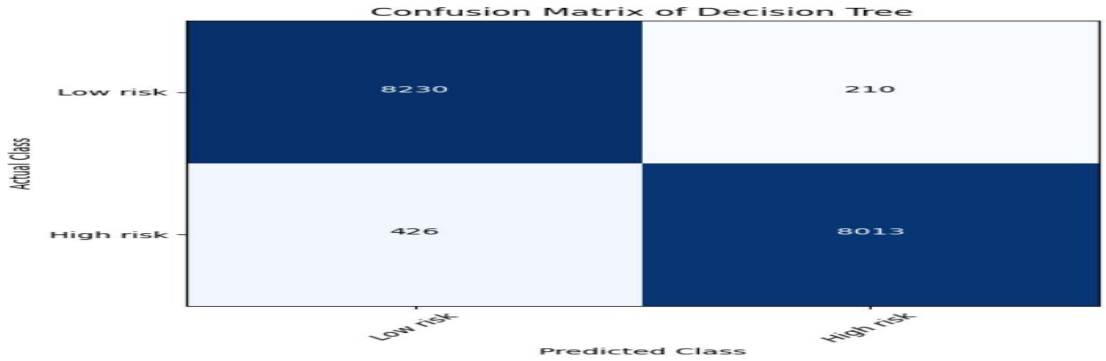


Table 5- 19 Confusion matrix of Decision Tree model

Table 5-20 below presents the results of Decision Tree with different data balancing techniques.

Performance Evaluation Results of Decision Tree Model			
Metrics	SMOTE(%)	ADASYN(%)	SMOTE+TOMEKlinks(%)
Accuracy	96	96	96
Precision	96	96	96
Recall	96	96	96
F1-Score	96	96	96
AUC-ROC	97	97	97

Table 5- 20 Results of Decision Tree Model

Table 5- 20 illustrates the accuracy, precision, recall, F1-score, and AUC-ROC achieved in each of the three dataset balancing techniques for the DT model. And it shows that DT with a dataset balanced by all techniques yields the same results with an evaluated accuracy score of 96% and a precision score of 96%.

5.7.2.5.Random Forest

The classification report shown in the Figure 5-31 presents the performance of the Random Forest model in the prediction of under-five mortality.

	precision	recall	f1-score	support
0.0	0.95	1.00	0.97	8440
1.0	1.00	0.94	0.97	8439
accuracy			0.97	16879
macro avg	0.97	0.97	0.97	16879
weighted avg	0.97	0.97	0.97	16879

Figure 5- 31 Classification report of the Random Forest model

The results in figure 5-31 shows, the random forest model obtained the overall testing accuracy score of 97.2%. This indicates that the model generalize very well on unseen data in the prediction of under-five mortality. The testing accuracy obtained from the Random Forest model is the best accuracy of all models discussed above. The model achieved the precision score of 97%, the recall score of 97%, and f1-score of 97%. This shows that the model is able to correctly identify 97% of all positive instances. A higher precision score indicates that the model is more cautious in identifying positive instances. And also the model achieved AUC-ROC score of 99%, which suggests that the model is accurate in distinguishing between positive and negative cases during the prediction of under-five mortality. The Table 5-21 shows the confusion matrix of the Random Forest model, and we observed the significant improvement on how the model predicted the cases correctly and the number of cases predicted incorrectly were decreased. For instance, in the first row there are a total of 8,440 test datasets of the low risk children. Out of these 8,440 children, 8,424 were correctly predicted as the low risk children, while only 16 cases were incorrectly predicted as the high risk children.

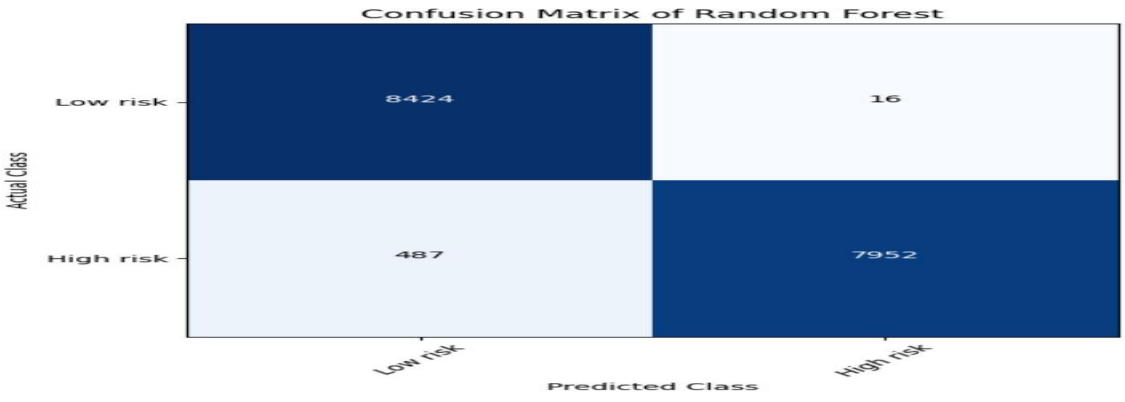


Table 5- 21 Confusion matrix of Random Forest Model

The results of Random Forest model based on data balancing techniques presented in the table below.

Performance Evaluation Results of Random Forest Model			
Metrics	SMOTE(%)	ADASYN(%)	SMOTE+TOMEKlinks(%)
Accuracy	97	97	97
Precision	97	97	97
Recall	97	97	97
F1-Score	97	97	97
AUC-ROC	99	99	99

Table 5- 22 Result of Radom Forest model

Table 5-22 illustrates the accuracy, precision, recall, F1-score, and AUC-ROC achieved in each of the three dataset balancing techniques for the RF model. And it shows that RF with a dataset balanced by all data balancing techniques yields the same results with an accuracy score of 97% and a precision score of 97%.

5.7.2.6.Extreme Gradient Boosting

The classification report in figure 5-32 shows, the performance of the Extreme Gradient Boosting model.

Extreme Gradient Boosting:

	precision	recall	f1-score	support
0.0	0.96	1.00	0.98	8440
1.0	1.00	0.96	0.98	8439
accuracy			0.98	16879
macro avg	0.98	0.98	0.98	16879
weighted avg	0.98	0.98	0.98	16879

Figure 5- 32 Classification report of Extreme Gradient Boosting model

From the results shown in figure 5-32, the Extreme Gradient Boosting model obtained an overall testing accuracy of 97.9%. This indicates that the model generalize very well on unseen data, and no sign of overfitting is observed from the result. The model achieved the precision score of 98%, the recall score of 98%, and the f1-score of 98%. This shows that the XGBoost model performs exceptionally well on both positive and negative instances. Based on the feature importance score of the XGBoost model, antenatal care, child gender, wealth index, total number of alive children, preceding child alive, physical healthy ,birthplace, weight of baby, site, gravidity, attendant at birth, mother age at first delivery, pregnancy duration, physically normal, literacy, occupation, grade completed, humidity, TMPMAX, TMPMIN, Precipitation and Ageatbirth were identified as factors influencing the under-five mortality. As shown in Figure 5-25, antenatal care, child gender, wealth index and total

number of alive children are the top 4 most significant factors influencing under-five mortality whereas TMPMAX, TMPMIN, Precipitation and Ageatbirth are less significant factors influencing under-five mortality. While these features still have some impact, their influence is considerably weaker compared to the most important factors influencing under-five mortality in eastern Hararghe Ethiopia. The feature importance of XGBoost model is shown in the figure 5-33.

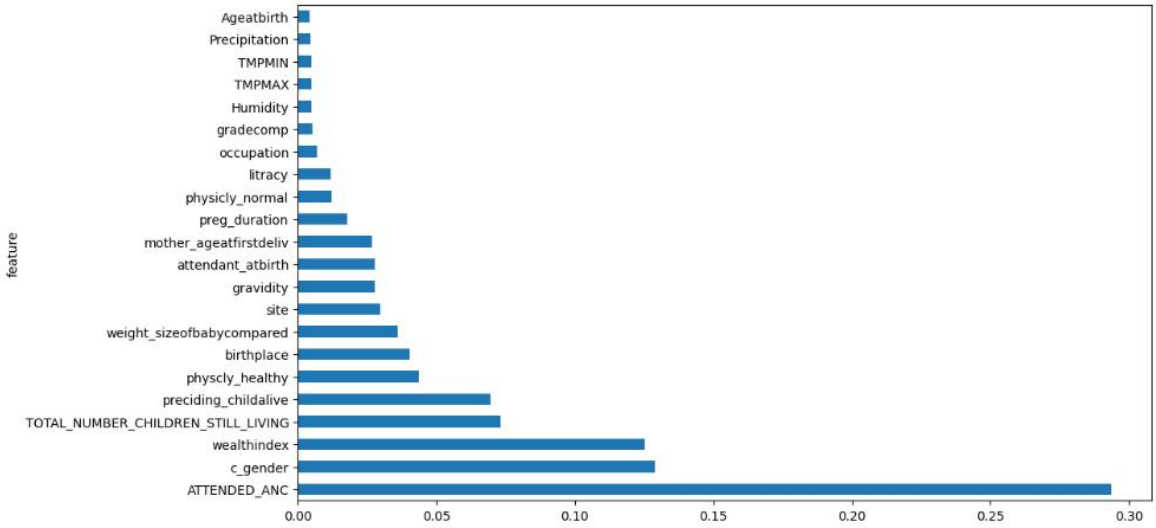


Figure 5- 33 Feature importance of XGBoost model

Table 5-23 shows the confusion matrix of XGBoost model. In the first row of Table 5-23, there are a total of 8,440 test datasets of the low risk cases. Out of these 8,440 the low risk children, 8,423 were correctly predicted as the low risk children, while only 17 cases were incorrectly predicted as the high risk children. This indicates that the model accurately identifying both positive and negative cases and the number of false negative and false positive have decreased.

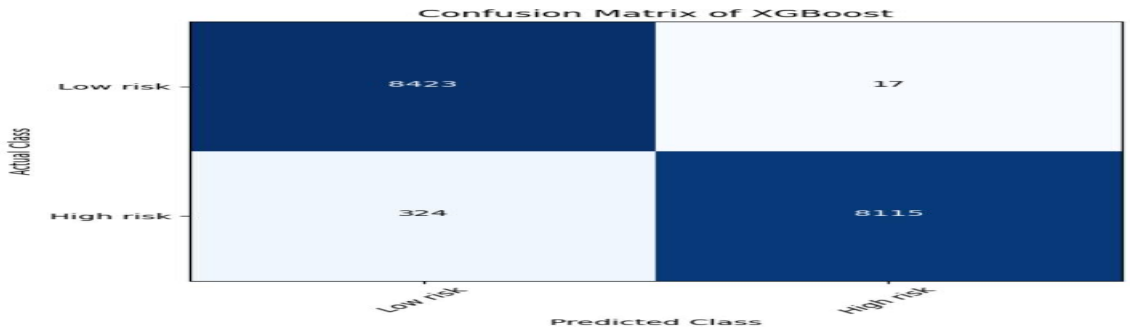


Table 5- 23 Confusion matrix of XGBoost model

The results of XGBoost model with different data balancing techniques is presented below in the table 5-24.

Performance Evaluation Results of Extreme Gradient Boosting Model			
Metrics	SMOTE(%)	ADASYN(%)	SMOTE+TOMEKlinks(%)
Accuracy	98	98	98
Precision	98	98	98
Recall	98	98	98
F1-Score	98	98	98
AUC-ROC	99	99	99

Table 5- 24 Results of XGBoost model

5.7.2.7.TabNet

Figure 5-34 presents the performance of TabNet model in prediction of under-five mortality. The TabNet model is trained for 100 epochs, 50 patience and obtained the overall testing accuracy of 92.6%.

	precision	recall	f1-score	support
0.0	0.89	0.98	0.93	8440
1.0	0.97	0.88	0.92	8567
accuracy			0.93	17007
macro avg	0.93	0.93	0.93	17007
weighted avg	0.93	0.93	0.93	17007

Figure 5- 34 Results of TabNet model

This indicates that the model performs well on unseen data in the prediction of under-five mortality. The TabNet model achieved a precision score of 93%, a recall score of 93%, and f1-score of 93%. This shows that the model is able to correctly identify 93% of the predicted positive instances and 93% of all actual positive instances. The table 5-25 shows the confusion matrix of the TabNet model. In the first row of table 5-25, there were a total of 8,440 test datasets of the low risk children. Out of these 8,440 children, 8,240 were correctly predicted as the low risk children, while 194 cases were incorrectly predicted as the high risk children. This indicates the model is moderately effective in identifying the positive and negative cases in the prediction of under-five mortality.

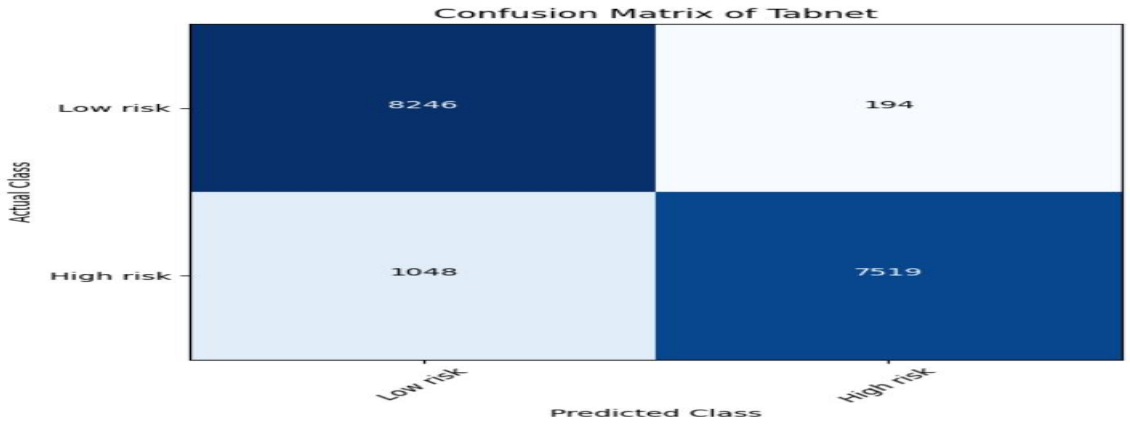


Table 5- 25 Confusion matrix of TabNet model

The results of TabNet model with different data balancing techniques presented in the table 5-26.

Performance Evaluation Results of Tabnet Model			
Metrics	SMOTE(%)	ADASYN(%)	SMOTE+TOMEKlinks(%)
Accuracy	87	93	88
Precision	87	93	89
Recall	87	93	88
F1-Score	87	93	88
AUC-ROC	95	96	95

Table 5- 26 Results of TabNet model

Table 5-26 illustrates the accuracy, precision, recall, F1-score, and AUC-ROC achieved in each of the three dataset balancing techniques for the TabNet model. And it shows that TabNet model with a dataset balanced by ADASYN yields the better results with an accuracy score of 93% and a precision score of 93%.

5.7.2.8.Convolutional Neutral Network

Figure 5-27 shows, the results obtained from the CNN model in prediction of under-five mortality. The model is trained for 100 epochs, 1024 batch size, and 0.5 dropout.

	precision	recall	f1-score	support
0.0	0.89	0.99	0.94	8416
1.0	0.99	0.88	0.93	8463
accuracy			0.94	16879
macro avg	0.94	0.94	0.94	16879
weighted avg	0.94	0.94	0.94	16879

Figure 5- 35 Classification report of the CNN model

As shown in the figure 5-35, the CNN model obtained the overall testing accuracy of 93.7%. This indicates the model produces reasonably good performance on unseen data in the prediction of under-five mortality. It suggests that the model is capable of making 94% accurate predictions on unseen data. The CNN model achieved the precision of 94%, recall of 94% and f1-score of 94%. This shows that the model is able to correctly identify 94% of all positive instances. Table below shows the results of CNN model with different data balancing techniques.

Model	Performance Evaluation Results of CNN		
Metrics	SMOTE(%)	ADASYN(%)	SMOTE+TOMEKlinks(%)
Accuracy	94	94	94
Precision	94	94	94
Recall	94	94	94
F1-Score	94	94	94
AUC-ROC	98	98	98

Table 5- 27 Results of the CNN model

Table 5-27 illustrates the accuracy, precision, recall, F1-score, and AUC-ROC achieved in each of the three dataset balancing techniques for the CNN model. And it shows that CNN with a dataset balanced by all techniques yields the same results with an accuracy score of 94% and a precision score of 94%.

The plot in the Figure 5-36 shows, the training and testing accuracy and loss of the CNN model in the prediction of under-five mortality. As observed in the figure, the lines are close to each other, indicating that the model is not overfitting.

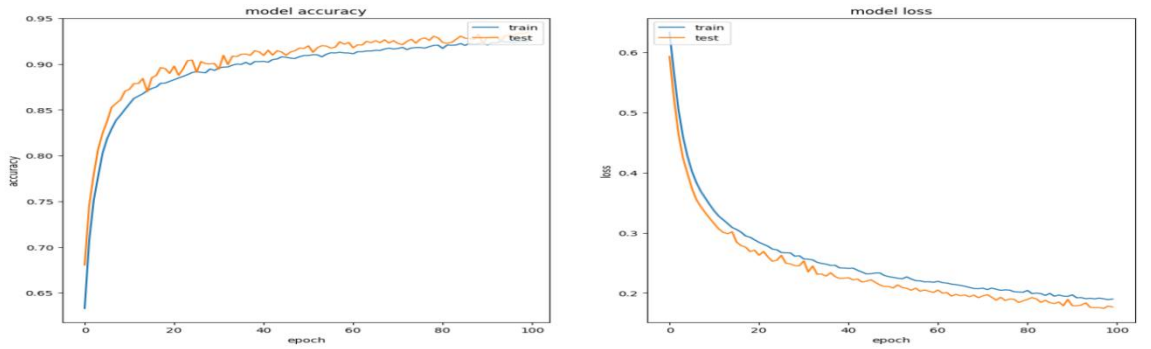


Figure 5- 36 Train, Test accuracy and loss graph of CNN model

The Figure 5-37 presents the comparisons of an overall testing accuracy score for each model used in the prediction of under-five mortality.

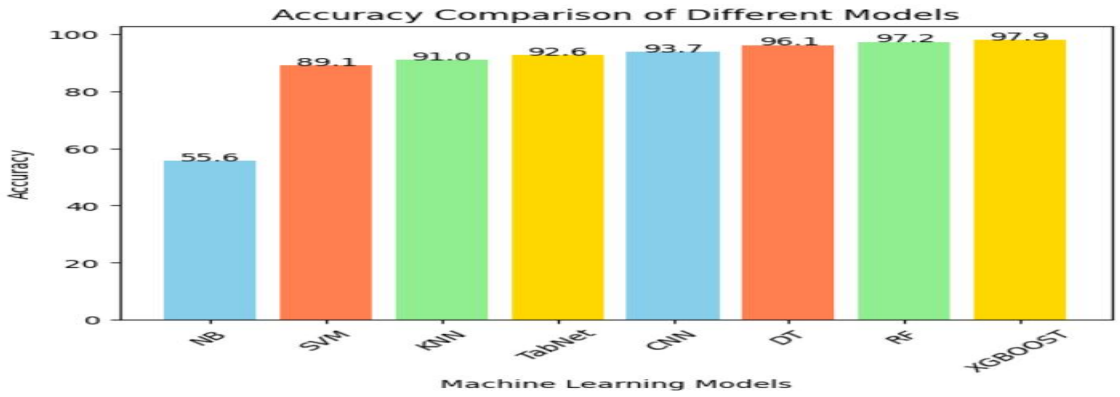


Figure 5- 37 Comparison of an overall accuracy of models used in under-five mortality prediction.

The accuracy of NB, SVM, KNN, TabNet, CNN, DT, RF, and XGBoost is 55.6%, 89.1%, 91.0%, 92.6%, 93.7%, 96.1%, 97.2%, and 97.9%, respectively. As shown in Figure 5-37, XGBOOST achieved the highest accuracy score of 97.9%, followed by RF with an overall testing accuracy score of 97.2%. This indicates that the XGBoost model outperformed the others with the highest testing accuracy score of 97.9%, surpassing the other models. The model generalize very well on unseen data. And Naïve Bayes yields poor results with a testing accuracy score of 55.6% because one of the limitations of the NB classifier is when there is a complex dataset.

The Figure 5-38 presents the comparisons of the AUC-ROC scores for each model used in the prediction of under-five mortality.

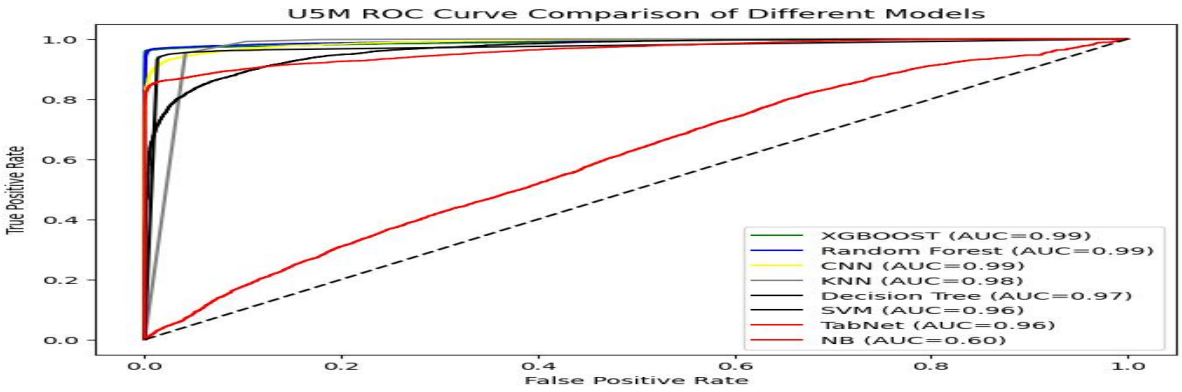


Figure 5- 38 Comparison of AUC-ROC scores of different models.

The AUC score of NB, TabNet, SVM, KNN, DT, CNN, RF, and XGBoost is 60%, 96%, 96%, 98%, 97%, 98%, 99%, and 99%, respectively. As shown in Figure 5-38, XGBoost and RF model achieved an impressive AUC-ROC score of 99%, showing exceptional performance in predicting under-five mortality. This suggests that the model is highly effective at distinguishing between which children were at risk of dying before their fifth birthday and not at risk of dying before the age of five.

5.8. Experimental Results: Summary of Experiments Comparison with Balanced and Imbalanced Data.

In this research work, we have conducted two experiments, one with balanced datasets and the other with imbalanced datasets, using different machine learning algorithms, namely, Naïve Bayes, Support Vector Machine, K Nearest Neighbours, Decision Tree, Random Forest, Extreme Gradient Boosting, TabNet, and Convolutional Neural Network. In this subsection, we have performed a comparative analysis of the machine algorithms used for prediction of under-five mortality. Summary of both experiments conducted in this study is presented in table 5-28. The table provides a summary of results obtained for the precision, recall, F1-score, AUC-ROC, and testing accuracy of the models used in prediction of under-five mortality.

From the two experiments conducted for the prediction of under-five mortality, we have realized that handling imbalanced dataset techniques enhanced accuracy for the prediction of under-five mortality. The degrees of improvement between the two experiments were significantly different. While some models revealed impressive accuracy when trained on an imbalanced dataset, their low AUC-ROC scores indicate that the models were struggling to

distinguish between children at risk of dying before the age of five or not at risk (low risk) of dying before their fifth birthday. The experiment that utilized the data balancing techniques showed a higher accuracy and AUC-ROC score than the other experiments that utilized an imbalanced dataset for the prediction of under-five mortality. Particularly, the second experiment achieved a higher AUC-ROC score compared with the first experiment, which used an imbalanced dataset for the prediction and yielded low AUC-ROC scores.

Table 5-28 presents the summary of the comparison of models using balanced and imbalanced datasets. The table illustrates the scores of performance evaluation metrics (precision, recall, f1-score, AUC-ROC, and accuracy) for each model used in this study. Based on the results presented in the table below, it is evident that the XGBoost model in experiment two exhibited high performance, achieving the highest testing accuracy score of 97.9%, precision score of 98, f1-score of 98, recall score of 98, and AUC-ROC score of 99%.

Experiment Number	ML model	Weighted Average				Training Accuracy	Testing Accuracy	Weighted Average testing Accuracy
		Precision	Recall	F1-Score	AUC			
Experiment One (Imbalanced Data)	NB	93%	35%	47%	61%	35%	35.1%	35%
	SVM	95%	95%	93%	69%	95%	95.3%	95%
	KNN	92%	95%	93%	57%	95%	94.9%	95%
	DT	96%	96%	95%	75%	96%	96.2%	96%
	RF	96%	96%	95%	78%	96%	96.3%	96%
	XGBoost	96%	96%	95%	80%	96%	96.2%	96%
	TabNet	93%	61%	61%	75%	63%	61.1%	61%
	CNN	96%	96%	95%	80%	96%	96.0%	96%
Experiment Two (balanced Data)	NB	61%	56%	49%	60%	57%	55.6%	56%
	SVM	90%	89%	89%	95%	90%	89.1%	89%
	KNN	92%	91%	91%	98%	94%	91.0%	91%
	DT	96%	96%	96%	97%	98%	96.2%	96%
	RF	97%	97%	97%	99%	98%	97.2%	97%
	XGBoost	98%	98%	98%	99%	98%	97.9	98%
	TabNet	93%	93%	93%	96%	93%	92.6%	93%
	CNN	94%	94%	94%	98%	95%	93.7%	93%

Table 5- 28 Summary of comparison of models using balanced and imbalanced dataset.

5.9. Prototype

In this research work, we have developed a graphical user interface using the Python programming language streamlit package to test the proposed model. It involves using a saved model from a training model and developing a prototype using the streamlit. Streamlit is a Python library that provides a simple and intuitive interface for building interactive user interfaces and deploying models. The main task is to predict under-five mortality using the inputs received from the users and predicting whether the input data is at high risk of dying or not. The sample of the prototype, which shows accepting the inputs from the user and displays the predicted result of the given input, is shown in Figure 5-39. The demo shows that the user can enter the input based on the provided forms on the user interface, and when the user clicks the "predict U5M using XGBoost" button, then the accepted input needs to be stored as a one-dimensional array and passes to the loaded model to predict under-five mortality, and the st.success() function displays the result as a success message.

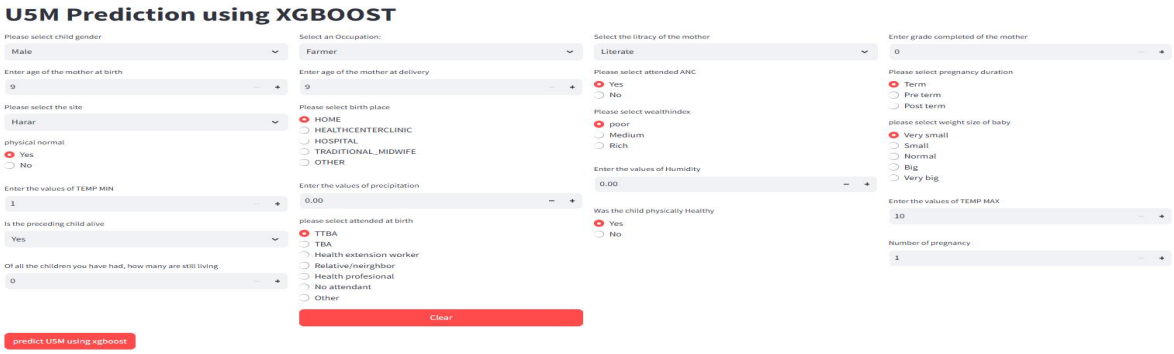


Figure 5- 39 Prototype of under-five mortality prediction

5.10. User Acceptance Testing

As once the experimentations were conducted and the prototype was developed based on the saved model then, the user acceptance testing evaluation is performed by the system's possible end-users to ensure that whether the performance of the system is accurate and the system is usable by the end-users. Thus, five users (3 from the health background and 2 from the IT background) were selected and had been given the chance to use and interact with the system. Therefore, to analyse the system performance with user evaluations, the selected users gave their answers based on Likert scale response options such as Excellent = 5, Very Good =4, Good =3, Fair =2 and Poor =1. This technique helped us to manually examine user acceptance based on the evaluator’s response. Table below shows, the evaluation criteria utilized in this

study along with the results of the end users.

No.	Criteria of evaluation	Excellent	Very Good	Good	Fair	Poor	Average	Percentage
1	Simplicity of the system	4	1	0	0	0	4.8	96%
2	Efficiency and Effectiveness of the system	2	3	0	0	0	4.4	88%
3	Attractiveness of the system	1	3	1	0	0	4	80%
4	Accuracy of the system in the prediction of under-five mortality.	2	3	0	0	0	4.4	88%
5	Importance of the system in the domain area	1	4	0	0	0	4.2	84%
6	Error tolerance of the system	2	1	2	0	0	4	80%
7	Total Average						**4.3**	**86%**

Table 5- 29 User acceptance testing evaluation results

As shown in Table 5-29, 80% of the evaluators assessed the prototype's simplicity as Excellent, 20% rated it as Very Good. In the second criteria of evaluation, the prototype effectiveness and efficiency 60% of the evaluators ranked it a Very Good rating and 40% ranked it an Excellent rating. The third criteria, which is Attractiveness of the prototype, 60% of the evaluators scored it a Very Good rating, 20% gave it a Good rating, and 20% gave it Excellent. In the fourth criteria, 40% of respondents ranked it Excellent for the accuracy of the predictive model, while 60% of the respondents ranked it as a Very Good. In the fifth criteria, which is regarding the importance of the developed prototype in the domain area, 80% of the evaluators gave it a Very Good rating, and 20% gave it an Excellent rating. The sixth and the final evaluation criterion is error tolerance of the system, in which 40% of respondents scored as a Very Good,20% of respondents scored as an Excellent and 40% of respondents ranked it as a Good. Finally, according to the user's evaluation results, the prototype average performance is 4.3 out of 5. This result revealed that the under-five mortality prediction prototype overall average performance is 86%, which is beyond Very Good.

5.11. Discussion of the Results

The main objective of this study is to develop a predictive model for under-five mortality

using machine learning algorithms on health, socio-demographic, and climate data in Eastern Hararghe, Ethiopia. In order to achieve the objective, the researchers evaluated the performance of eight machine learning algorithms, namely NB, SVM, KNN, DT, RF, XGBoost, TabNet, and CNN. We conducted two experiments, one using balanced datasets utilizing data balancing techniques and the other using imbalanced datasets without utilizing techniques of handling imbalanced datasets to test the models performance on both techniques. Our finding shows that the model trained with a balanced dataset achieves higher accuracy than the model trained with an imbalanced dataset. Among the models, the Extreme Gradient Boosting model achieved higher performance with an overall testing accuracy score of 97.9%, followed by RF, DT, CNN, TabNet, KNN, SVM, and NB models with an overall testing accuracy score of 97.2%, 96.2%, 93.7%, 92.6%, 91.0%, 89.1%, and 55.6%, respectively. Therefore, we conclude the XGBoost model is the most appropriate model for prediction of under-five mortality because of its higher accuracy and better performance than other models used in this research work. In this study, two experiments were designed and conducted to provide an answer for the formulated research questions.

The feature importance evaluation result shown in Figure 5-33 provides an answer for the RQ1[What are the factors (health, socio-demographic and climate factors) that are determinants for under-five mortality in eastern Hararghe, Ethiopia?], and the result shows, based on the feature importance score of the XGBoost model, antenatal care, child gender, wealth index, total number of alive children, preceding child alive, physical healthy ,birth place, weight of baby, site, gravidity, attendant at birth, mother age at first delivery, pregnancy duration, physically normal, literacy, occupation, grade completed, humidity, TMPMAX, TMPMIN, Precipitation and Ageatbirth were identified as factors influencing the under-five mortality. As shown in Figure 5-33, it is evident that antenatal care, child gender, wealth index and total number of alive children are the top 4 most significant factors influencing under-five mortality whereas TMPMAX, TMPMIN, Precipitation and Ageatbirth features are less significant factors influencing under-five mortality. While climate features still have some impact, their influence is considerably less compared to the most important factors (health, socio-demographic factors) influencing under-five mortality in eastern Hararghe Ethiopia. The reason why we used XGBoost feature importance score was because of its higher accuracy and better performance than other models used for the prediction of under-five mortality.

The experimental results shown in table 5-28 provide an answer to the RQ2 [Which machine learning algorithm is effective, and appropriate for predicting under-five mortality?] ,the XGBoost model performs better than other models used for predicting under-five mortality, with an overall testing accuracy of 97.9%, and an AUC-ROC score of 99%. And 98%, 98%, and 98% for the precision, recall, and F1-score, respectively. Hence, the XGBoost model is selected for developing a model used for prediction of under-five mortality.

As once we identified the best performing model, which is the XGBoost model, from the result shown in table 5-28, the researchers conducted an experiment to provide an answer to the RQ3 [To what extent can the proposed model predict the under-five mortality?] The model was trained with different parameters to evaluate to what extent the XGBoost model can predict under-five mortality. The maximum performance we achieved with the XGBoost model was an overall testing accuracy score of 97.9% and the best parameters used are listed below.

Best Parameters	**Parameter Value**
max_depth	7
learning_rate	0.1
colsample_bytree	0.9
gamma	0.1
n_estimators	300
subsample	0.8

Table 5- 30 Best Parameters used

5.12. Comparison of related works

In this section, we have compared our proposed study with the previous studies on under-five mortality prediction. As discussed in section 2. There are the researches conducted on prediction of under-five mortality but all of the studies were limited to utilizing the socio-demographic survey data to determine the significant factors that influence the prediction of under-five mortality, but our study integrate data of health, socio-demographic, and climate data to develop predictive model for under-five mortality. When we compare our result with others researchers on prediction of under-five mortality areas, our model outperform the studies conducted so far on prediction of under-five mortality. The summary of the comparison with the previous studies is shown in the following Table 5-29.

No.	Author	Title	ML/DL models used	Accuracy
1.	Saroj, Yadav, Rajneesh, & Chilyabanyama, (2022)	Machine Learning Algorithms for understanding the determinants of under-five Mortality(2022)	Decision Tree, Random Forest, Naïve Bayes, K-Nearest Neighbor (KNN), Logistic Regression, Support Vector Machine (SVM), Neural Network and Ridge Classifier	95.96
2.	Fikrewold, Samuel, Lloyd, & Corey, 2020	Machine learning approach for predicting under-five mortality determinants in Ethiopia.	Random Forest, KNN, and Logistic Regression	67.2%
3	Adegbosin, B, & J, 2019	Predicting Under-five mortality across 21 Low and Middle countries.	Deep Neural Networks, Convolutional Neural Networks, Logistic Regression, and Hybrid CNN-DNN	not mentioned in Accuracy
4	Carlos, et al., 2023	Understanding the social determinants of child mortality in Latin America	Random Forest	not mentioned in Accuracy

		over the last two decades: a machine learning approach		
5	Solomon, Angela, Oluwafemi, Christabel, & Ignace, 2023	Trend Analysis and Determinants of under-5 Mortality in Nigeria: A Machine Learning Approach	Deep Neural Network (DNN)	74%
6	Our study	Machine Learning-Based Prediction of Under-Five Mortality Using Health, Socio-Demographic, and Climate Data in Eastern Hararghe, Ethiopia.	XGBoost	97.9%
			RF	97.2

Table 5- 31 Comparison of related works.

CHAPTER 6
6. CONCLUSION AND RECOMMENDATION

This chapter presents the conclusion and recommendation of the proposed model for under-five mortality prediction on health, socio-demographic, and climate data. Section 6.1 presents the conclusion based on the research findings. Section 6.2 discusses the contribution of this research work, and Section 6.3 discusses recommendations for future researchers who are interested in continuing in the same or related research area.

6.1. Conclusion

Health is one of the important elements of international development, and the world has made significant investments to mitigate morbidity and mortality rates, with a focus on vulnerable populations like the poor, women, and children. Under-five mortality is the probability of the children dying before reaching five years and it is the most commonly used indicator to measure the health status of children. Despite the significant improvement in the reduction of the under-five mortality, Ethiopia is still home to high deaths of children younger than five years. Understanding of the risk factors that influence under-five mortality provides insights into accelerating under-five mortality reduction, and it helps to meet the SDG 3.2 target, which is to reduce under-five mortality rate to at least 25 deaths per 1,000 live births.

In this study, we used machine learning algorithms to predict under-five mortality based on health, socio-demographic, and climate data in eastern Hararghe, Ethiopia. The research utilized design science research methodology and techniques. In this research work, we have used two data sources: one is health, socio-demographic data from Hararghe Health Demographic Surveillance System, and climate data from the National Meteorology Agency. We have integrated these two data to create one comprehensive dataset used for model training. Preprocessing techniques like data cleaning, handling missing values, handling categorical variables, dataset balancing, feature scaling, and feature selection were applied to the dataset. Then the researchers split the dataset into 80% for training and 20% for testing. The researchers used performance evaluation metrics like precision, recall, F1-score, accuracy, and AUC-ROC score to evaluate the performance of the models in predicting under-five mortality. In this research work, we have carried out two experiments, one using data balancing techniques and the other using an imbalanced dataset to develop predictive models

for under-five mortality. And eight machine learning algorithms (NB, SVM, KNN, DT, RF, XGBoost, TabNet, and CNN) were used in this research work. Based on the results of the experiments, using data balancing techniques achieved higher accuracy and better performance in prediction of under-five mortality. XGBoost model outperforms all models with testing accuracy score of 97.9%. And the model achieved the precision score of 98%, recall score of 98%, F1-score of 98%, and AUC-ROC score of 99%. Followed by RF having testing accuracy score of (97.2%), DT (96.1%), CNN (93.7%), TabNet (92.6%), KNN (91.0%), SVM (89.1%), and NB (55.6%). Thus, the XGBoost model provides better predictive power than other models used in predicting under-five mortality in eastern Hararghe, Ethiopia. Using the feature importance of the XGBoost model, antenatal care, child gender, wealth index, total number of alive children, preceding child alive, physical healthy ,birth place, weight of baby, site, gravidity, attendant at birth, mother age at first delivery, pregnancy duration, physically normal, literacy, occupation, grade completed and humidity were identified as determinant factors influencing the under-five mortality. While climate features still have some impact, their influence is considerably weaker compared to the most important factors (health, socio-demographic factors like antenatal care, child gender, and wealth index) influencing under-five mortality in eastern Hararghe Ethiopia. This study emphasizes the use of machine learning algorithms to identify and understand significant risk factors that are determinant to under-five mortality, with the goal of informing policy decisions. In conclusion, this research provides insights into accelerating the reduction of under-five mortality rate, which helps the country to meet the SDG3.2 target of reducing under-five mortality rate to less than 25 out of 1,000 live births by 2030. Also, our findings reveal that priority should be given to the most important identified factors that influence under-five mortality, including antenatal care, wealth index, and birthplace.

6.2. Contribution of the Thesis

The main contribution of this study are discussed as follows.

- Proposed the machine learning model for the prediction of under-five mortality on health, socio-demographic, and climate data in eastern Hararghe, Ethiopia, with significantly high accuracy, precision, recall, F1-score, and AUC-ROC score.
- Prepared a comprehensive dataset by integrating health, socio-demographic, and climate data that can be used by other researchers in this research area.

- A prototype based on the model saved is developed.
- Identified the important factors that influence the under-five mortality using the model with high performance to enhance the acceleration in reduction of under-five mortality.

6.3. Recommendation

This research aimed to develop a predictive model of under-five mortality on health, socio-demographic, and climate data in eastern Hararghe, Ethiopia, using supervised machine learning algorithms. And the climate data utilized in this research were at the site level, not the household level data. In our finding, the climate factors are less significant compared to health and socio-demographic factors in influencing under-five mortality in the context of eastern Hararghe, Ethiopia, so there is a need to see the climate effects in different regions of the country. Thus, the following suggested recommendations shall be considered and addressed in the future work.

- Increase the size of the data by accessing the health, socio-demographic data from 10+ HDSS sites in Ethiopia and integrate with the household level climate data, aiming to improve the accuracy of the prediction.
- Integrate Extreme Gradient Boosting model with other state-of-the-art to improve the training time and prediction time.

REFERENCES

Adegbosin, A. E., B. S., & J. S. (2019). Predicting Under-five mortality across 21 Low and Middle countries. . . doi:https://doi.org/10.1101/19007583

Boateng, E. Y., J. O., & D. A. (2020). Basic Tenets of Classification Algorithms K-Nearest-Neighbor, Support Vector Machine, Random Forest and Neural Network: A Review. *Journal of Data Analysis and Information Processing*.

Saroj, R. K., Yadav, P. K., R. S., & Chilyabanyama, O. N. (2022). Machine Learning Algorithms for understanding the determinantsof under-five Mortality. *BioData Mining*. doi:https://doi.org/10.1186/s13040-022-00308-8

A. A., & A. O. (2014). Classification of Phishing Email Using Random Forest Machine Learning Technique.

A. S., B. O., & H. K. (2014). Knowledge and perceptions about the health impact of climate change among health sciences students in Ethiopia: a cross-sectional study .

Alshaher, H. (2021). Studying the Effects of Feature Scaling in Machine Learning.

Alzubaidi, L., J. Z., A. J., A. A.-D., Y. D., O. A.-S., . . . L. F. (2021). Review of deep learning: concepts, CNN architectures, challenges, applications, future directions. *Journal of Big Data, 8*. Retrieved from https://link.springer.com/article/10.1186/s40537-021-00444-8

Arık, S. O., & T. P. (2021). TabNet: Attentive Interpretable Tabular Learning.

Aryo, A. D., & S. S. (2021). Performance Comparison of Grid Search and Random Search Methods for Hyperparameter Tuning in Extreme Gradient Boosting Algorithm to Predict Chronic Kidney Failure. *International Journal of Intelligent Engineering and Systems, Vol.14*. doi:10.22266/ijies2021.1231.19

B. S., H. B., W. D., A. K., K. B., & J. S. (2016). Review of climate change and health in Ethiopia.

Badillo, S., B. B., F. B., Davydov, I. I., L. H., T. K.-T., . . . Zhang, J. D. (2020). An Introduction to Machine Learning.

Bharadiya, J. P. (2023). Convolutional Neural Networks for Image Classification. *International Journal of Innovative Science and Research Technology, 8*. Retrieved from https://www.researchgate.net/profile/Jasmin-Bharadiya-4/publication/370944952_Convolutional_Neural_Networks_for_Image_Classification/links/646b960b7b575d49292a0be3/Convolutional-Neural-Networks-for-Image-

Classification.pdf?__cf_chl_tk=1KS2tRM61sa0zbEwZBfk

Bisandu, D. B. (2016). Design science research methodology in Computer Science and Information Systems.

Brocke, J. v., A. H., & A. M. (2020). Introduction to Design Science Research.

Buse, K., & S. H. (2015). Health in the sustainable development goals:development goals:ready for a paradigm shift? *Globalization and Health*.

C.-W. C., Y.-H. T., F.-R. C., & W.-C. L. (2020). Ensemble feature selection in medical datasets: Combining filter, wrapper, and embedded feature selection results.

D. H., C. A., M. N., Ebi, K. L., P. F., & T. A. (2021). Climate change and child health: a scoping review and an expanded conceptual framework.

D. P., & D. N. (2022). Climate Change, Fossil-Fuel Pollution, and Children's Health. *The NEW ENGLAND JOURNAL of MEDICINE*, VOL. 386 NO. 24. doi:DOI: 10.1056/NEJMra2117706

D. R.-M., M. R., T. R.-M., & K. C. (2020). Introduction to Anaconda and Python:Installation and setup.

D.Moyer, J., & S. H. (2020). Are we on the right path to achieve the sustainable development goals?

Danquah, R. A. (2020). Handling Imbalanced Data: A Case Study For Binary Class Problems.

Dheresa, M., Roba, H. S., G. D., M. A., Tura, A. K., T. A., . . . T. D. (2022). Uncertainties in the path to 2030: Increasing trends of under-five mortality in the aftermath of Millennium Development Goalin Eastern Ethiopia. *Journal of Global Health*.

Dodo Zaenal Abidin, S. N. ((2020)). RSSI Data Preparation for Machine Learning .

E. A., K. G., M. K., A. H., A. M.-P., A. B., . . . N. S. (2021). Climate Change and Child Health Inequality: A Review of Reviews.

Ethiopian Public Health Institute, Federal Ministry of Health, & Icf. (2021). Ethiopia Mini Demographic and Health Survey 2019. Addis Ababa, Ethiopia.

F. H., S. H., L. P., & C. S. (2020). Machine learning approach for predicting under-five mortality determinants in Ethiopia.

Fernández, H. A., García, L. S., Galar, M, & Prati. (2018). Learning from Imbalanced Data Sets. Springer, Gewerbestrasse.

Feyisa Abebe Gelashe, G. M. (2020). perodsad.

FILIOU, A. (2023). A C omparative Analysis Of The Tabnet And Xgboost Algorithms For Breast Cancer Classification.

G. C., & F. S. (2014). A survey on feature selection methods.

G. G., H. W., D. B., Y. B., & K. G. (2003). KNN Model-Based Approach in Classification.

Gebretsadik, S., & E. G. (2016). Determinants of Under-Five Mortality in High Mortality Regions of Ethiopia: An Analysis of the 2011 Ethiopia Demographic and Health Survey Data. *International Journal of Population Research* .

Gurucharan, M. (2024). *Basic CNN Architecture: Explaining 5 Layers of Convolutional Neural Network*. Retrieved from upGrad.

H. R., & S R, S. C. (2019). Analysis of Decision Tree and K-Nearest Neighbor Algorithm in the Classification of Breast Cancer.

H. S., M. A.-F., & S. K. (2018). Predicting Potential Banking Customer Churn using Apache Spark ML and MLlib Packages: A Comparative Study. *(IJACSA) International Journal of Advanced Computer Science and Applications,*, Vol. 9, No. 11.

Hanna, R., & P. O. (2016). Implications of Climate Change for Children in Developing Countries.

Health, K. (2024). *Health and Well-being in Korea: Balancing Tradition and Modernity for a Holistic Lifestyle*. Retrieved from Tech Pro: https://techpro67.tistory.com/m/2

Helldén, D., C. A., M. N., K. L., P. F., & T. A. (2021). Climate change and child health: a scoping review and an expanded conceptual framework.

Hevner, A. R., S. R., March , a. T., & J. P. (2004). Design Science in Information systems Research.

Khosravi, B., A. D., F. N., J. P., H. M., C. C., & R. E. (2023). Demystifying Statistics and Machine Learning in Analysis of Structured Tabular Data. *The Journal of Arthroplasty, 38*(10). doi:https://doi.org/10.1016/j.arth.2023.08.045

Lee, S., & C. L. (2020). Revisiting spatial dropout for regularizing convolutional neural networks. *Multimedia Tools and Applications , 79*, pages 34195–34207. Retrieved from https://link.springer.com/article/10.1007/s11042-020-09054-7

Li, Z., F. L., W. Y., S. P., & J. Z. (2021). A Survey of Convolutional Neural Networks: Analysis, Applications, and Prospects. *IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS*.

doi:https://doi.org/10.1109/TNNLS.2021.3084827.

M. B., A. G., & A. C. (2022). A comparative analysis of K-Nearest Neighbor, Genetic, Support Vector Machine, Decision Tree, and Long Short Term Memory algorithms in machine learning.

M. V., M. G., & M. R. (2020). Performance Analysis and Comparison of Machine and Deep Learning Algorithms for IoT Data Classification.

Matsuo, Y., Y. L., M. S., D. P., D. S., M. S., . . . J. M. (2022). Deep learning, reinforcement learning, and world models.

McDonnell, K., F. M., B. S., L. M., & G. C. (2023). Deep learning in insurance: Accuracy and model interpretability using TabNet.

Mienye, I. D., & Y. S. (2022). A Survey of Ensemble Learning: Concepts, Algorithms, Applications, and Prospects.

Morera, A., J. M., J. A., J. L., & S. d.-M. (2021). Performance of statistical and machine learning-based methods for predicting biogeographical patterns of fungal productivity in forest ecosystems. *Morera et al. Forest Ecosystems*. doi:https://doi.org/10.1186/s40663-021-00297-w

N. A., Y. L., B. B., H. K., D. Z., N. B., . . . M. D. (2016). Neonatal mortality and causes of death in Kersa Health and Demographic Surveillance System (Kersa HDSS), Ethiopia, 2008–2013.

P. D., & C. A. (2021). A comprehensive survey on feature selection in the various fields of machine learning.

P. M., & Yadav, A. S. (2020). Improving the Classification Accuracy using Recursive Feature Elimination with Cross-Validation.

P. S., & N. K. (2023). An Overview on Data Cleaning on Real World Data.

PEFFERS, K., T. T., ROTHENBERGE, M. A., & S. C. (2007). A Design Science Research Methodology for Information Systems Research.

Pereira, L. M., A. S., & L. V. (2023). A Comparative Study on Recent Automatic Data Fusion Methods. *MDPI*. doi:https://doi.org/10.3390/computers13010013

Poian, V. D., B. T., L. C., B. M., J. M., J. C., & S. H. (2023). Exploratory data analysis (EDA) machine learning approaches for ocean world analog mass spectrometry. *Frontiers in Astronomy and Space Sciences, 10 -2023*.

doi:https://doi.org/10.3389/fspas.2023.1134141

S, R., N. U., S. R., & S. B. (2016). Experimenting XGBoost Algorithm for Prediction and Classifi cation of Different Datasets.

S. B., & K. L. (2021). Resampling imbalanced data for network intrusion detection datasets. *Journal of big data*.

S. J., W. E., M. D., & L. K. (2020). The effects of climate change on human health in Africa, a dermatologic perspective: a report from the International Society of Dermatology Climate Change Committee.

S. N., A. C., O. O., C. J., & I. H.-K. (2023). Trend Analysis and Determinants of under5 Mortality in Nigeria: A Machine Learning Approach.

S. O., & T. P. (2021). TabNet: Attentive Interpretable Tabular Learning.

Sarker, I. H. (2021). Deep Learning: A Comprehensive Overview on Techniques, Taxonomy, Applications and Research Directions.

Sharma, N., R. S., & N. J. (2021). Machine Learning and Deep Learning Applications-A Vision.

Simane, B., H. B., W. D., A. K., K. B., & J. S. (2016). Review of Climate Change and Health in Ethiopia: Status and Gap Analysis .

Swana, E. F., W. D., & P. B. (2022). Tomek Link and SMOTE Approaches for Machine Fault Classification with an Imbalanced Dataset.

T. E., T. M., D. M., T. S., B. M., & O. T. (2021). A survey on missing data in machine learning.

Tamboli, N. (2023, July 14). Effective Strategies for Handling Missing Values in Data Analysis (Updated 2023).

Tura, A. K. (2021). bhbh.

UNICEF. (2024, 03). Retrieved from unicef: https://data.unicef.org/topic/child-survival/under-five-mortality/

UNICEF. (2024). Under-five mortality.

United Nations Inter-Agency Group for Child Mortal. (2020). Levels & Trends in Child Mortality.

W. W. (2024). *World Health Organization, History, Organization, & Definition of Health*. Retrieved from Wikipedia: https://en.wikipedia.org/wiki/World_Health_Organization

W.-C. L., & C.-F. T. (2019). Missing value imputation: a review and analysis of the literature (2006–2017).

Waititu1, H. W., Koskei1 , o. A., & Onyango, N. O. (2020). Determinants of Under Five Child Mortality from KDHS Data: A Balanced Random Survival Forests (BRSF) Technique. *International Journal of Statistics and Applications 2020*.

WHO. (2020). improving survival and well-being.

WHO. (2023). Climate change.

X. D., A. Y., J. Z., D. X., W. Y., Z. S., . . . X. C. (2022). Bagging–XGBoost algorithm based extreme weather identification and short-term load forecasting model.

Y. Z., W. S., X. L., & L. L. (2018). Chi-square Statistics Based Feature Selection Method in Text Classification.

Zhang, Y., J. L., & W. S. (2022). A Review of Ensemble Learning Algorithms Used in Remote Sensing Applications.

APPENDICES

Appendix A: sample code for importing necessary library

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.preprocessing import LabelEncoder
from sklearn.impute import KNNImputer
from imblearn.over_sampling import SMOTENC,RandomOverSampler,KMeansSMOTE
from sklearn.model_selection import train_test_split
from sklearn.svm import SVC
from sklearn.neighbors import KNeighborsClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import
accuracy_score,confusion_matrix,classification_report,roc_curve,auc,RocCurveDisplay
from sklearn.model_selection import cross_val_score
from sklearn.utils import resample
pd.set_option('display.max_columns', None)
import pickle
%matplotlib inline
import pymysql
from sqlalchemy import create_engine
import warnings
warnings.filterwarnings('ignore')
%load_ext sql
```

Appendix B: sample code data balancing

```
def data_balance(X,Y):
    #SMOTE DATA BALANCING
    smt=SMOTE(sampling_strategy='auto',random_state=42)
    x_smoteresampled,y_smoteresampled=smt.fit_resample(X,Y)

    #ADASYN
    adasyn=ADASYN(random_state=42)
    x_adasynresampled,y_adasynresampled=adasyn.fit_resample(X,Y)

    #TOMEK LINK
    tom=TomekLinks()
    x_tomresampled,y_tomresampled=tom.fit_resample(X,Y)
    #SMOTE+TOMEkLINKS
    tomlink=TomekLinks()

x_smotetomelinkresampled,y_smotetomelinkresampled=tomlink.fit_resample(x_smoteresampled,y_s
```

```
moteresampled)
return adasynresampled,y_adasynresampled,x_smotetomelinkresampled,y_smotetomelinkresampled
```

Appendix C: Sample code to plot feature importance

```
def feature_imp(df,model):
    fi = pd.DataFrame()
    fi["feature"] = df.columns
    fi["importance"] = model.feature_importances_
    fi.sort_values(by="importance", ascending=False).plot('feature', 'importance',
     'barh',        figsize=(12,7), legend=False)
    plt.show()
```

Appendix D: Sample code for model training

```
def xgboost_f(X_train,X_test,y_train,y_test):
     param_grid = {
    'max_depth': [3, 5, 7],
    'learning_rate': [0.01, 0.05, 0.1,0.3],
    'n_estimators': [100, 200, 300],
    'subsample': [0.3,0.6, 0.8, 0.9],
    'colsample_bytree': [0.1,0.3, 0.9],
    'gamma': [0.1, 0.2,0.3,0.6,0.9],
    }
    grid_search = GridSearchCV(XGBClassifier(),param_grid, cv=5)
    grid_search.fit(X_train, y_train)
    xgb_bp = grid_search.best_params_
    xgb_model=xgb.XGBClassifier(n_estimators=xgb_bp["n_estimators"],

                        max_depth=xgb_bp["max_depth"],
                        learning_rate=xgb_bp["learning_rate"],
                        )
    xgb_model.fit(X_train, y_train)
    y_pred = xgb_model.predict(X_test)

    y_pred_proba = xgb_model.predict_proba(X_test)[:, 1]

    # Calculate ROC curve and AUROC
    fpr, tpr, thresholds = roc_curve(y_test, y_pred_proba)
    roc_auc = auc(fpr, tpr)
    cm=confusion_matrix(y_test,y_pred)

    # Evaluate the model
    cl_report = classification_report(y_test,y_pred])
    accuracy = accuracy_score(y_test, y_pred)
    print(cl_report,"\n")
    print('Accuracy on test set:'.format(accuracy))
  feature_imp(X_train,xgb_model)
```

Appendix E: sample code for plotting the AUC-ROC curve

```
nb_model = pickle.load(open(r"C:\Users\fgelashe\Feyisathesis\model_saved\nbpredict.sav", 'rb'))
svm_model = pickle.load(open(r"C:\Users\fgelashe\Feyisathesis\model_saved\svmpredict.sav", 'rb'))
knn_model = pickle.load(open(r"C:\Users\fgelashe\Feyisathesis\model_saved\knnpredict.sav", 'rb'))
dt_model = pickle.load(open(r"C:\Users\fgelashe\Feyisathesis\model_saved\dtpredict.sav", 'rb'))
rf_model = pickle.load(open(r"C:\Users\fgelashe\Feyisathesis\model_saved\rfpredict.sav", 'rb'))
xgboost_model = pickle.load(open(r"C:\Users\fgelashe\Feyisathesis\model_saved\xgboostpredict.sav", 'rb'))
```

```
y_pred_proba1 = xgboost_model.predict_proba(X_test)[:, 1]
y_pred_proba2 = rf_model.predict_proba(X_test)[:, 1]
y_pred_proba3 = dt_model.predict_proba(X_test)[:, 1]
y_pred_proba4 = knn_model.predict_proba(X_test)[:, 1]
y_pred_proba5 = svm_model.predict_proba(X_test)[:, 1]
y_pred_proba6 = nb_model.predict_proba(X_test)[:, 1]


fpr1, tpr1, thresholds1 = roc_curve(y_test, y_pred_proba1)
fpr2, tpr2, thresholds2 = roc_curve(y_test, y_pred_proba2)
fpr3, tpr3, thresholds3 = roc_curve(y_test, y_pred_proba3)
fpr4, tpr4, thresholds4 = roc_curve(y_test, y_pred_proba4)
fpr5, tpr5, thresholds5 = roc_curve(y_test, y_pred_proba5)
fpr6, tpr6, thresholds6 = roc_curve(y_test, y_pred_proba6)

auc1 = roc_auc_score(y_test, y_pred_proba1)
auc2 = roc_auc_score(y_test, y_pred_proba2)
auc3 = roc_auc_score(y_test, y_pred_proba3)
auc4 = roc_auc_score(y_test, y_pred_proba4)
auc5 = roc_auc_score(y_test, y_pred_proba5)
auc6 = roc_auc_score(y_test, y_pred_proba6)
#roc_auc = auc(fpr, tpr)
```

```
plt.figure(figsize=(8, 6))
plt.plot(fpr1, tpr1, color='green', label='XGBOOST (AUC={:.2f})'.format(auc1))
plt.plot(fpr2, tpr2, color='blue', label='Random Forest (AUC={:.2f})'.format(auc2))
plt.plot(fpr3, tpr3, color='black', label='Decision Tree (AUC={:.2f})'.format(auc3))
plt.plot(fpr4, tpr4, color='gray', label='KNN (AUC={:.2f})'.format(auc4))
plt.plot(fpr5, tpr5, color='black', label='SVM (AUC={:.2f})'.format(auc5))
```

```
plt.plot(fpr6, tpr6, color='red', label='NB (AUC={:.2f})'.format(auc6))
plt.plot([0, 1], [0, 1], 'k--')  # Diagonal line (random guessing)
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('U5M ROC Curve Comparison of Different Model')
plt.legend(loc='lower right')
plt.show()
```

APPROVAL SHEET HARAMAYAUNIVERSITY
SCHOOL OF POST GRADUATE STUDIES

Machine Learning-Based Prediction of Under-Five Mortality Using Health, Socio-Demographic, and Climate Data in Eastern Hararghe, Ethiopia.

SUBMITTED BY:

FEYISA ABEBE	_____	_____
Name of Student	Signatures	Date

Approved by:

1. Abebe Belay Adege (PhD)		_____
Name of Major Advisor	Signature	Date
2. Mr. Tadesse Kebede(MSc)	_____	_____
Name of Co-Advisor	Signature	Date
3. _____	_____	_____
Name of Chairman, DGC	Signature	Date
4. _____	_____	_____
Name of Dean, SGS	Signature	Date
5. _____	_____	_____
Name of Chairman, CGS	Signature	Date