A project report on

CHILD MORTALITY PREDICTION USING MACHINE LEARNING

Submitted in partial fulfillment for the award of the degree of

M.Tech (Software Engineering)

by

HARISH R (18MIS0384)

VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

SCHOOL OF INFORMATION TECHNOLOGY & ENGINEERING

April, 2023

DECLARATION

I here by declare that the thesis entitled "CHILD MORTALITY PREDICTION USING MACHINE LEARNING" submitted by me, for the award of the degree of M.Tech (Software Engineering) is a record of bonafide work carried out by me under the supervision of Professor. Thandeeswaran R.

I further declare that the work reported in the thesis has not been submitted and will not be submitted, either in part or in full, for the award of any other degree or diploma in this institute or any other institute or university.

Place: Vellore **Signature of the Candidate**

Date: 04.04.2023 **Harish R**

CERTIFICATE

This is to certify that the thesis entitled "CHILD MORTALITY PREDICTION USING MACHINE LEARNING" submitted by HARISH R (18MIS0384), School of Information Technology & Engineering, Vellore Institute of Technology, Vellore for the award of the degree M.Tech (Software Engineering) is a record of bonafide work carried out by him/her under my supervision.

The contents of this report have not been submitted and will not be submitted either in part or in full, for the award of any other degree or diploma in this institute or any other institute or university. The Project report fulfils the requirements and regulations of VELLORE INSTITUTE OF TECHNOLOGY, VELLORE and in my opinion meets the necessary standards for submission.

Signature of the Guide **Signature of the HoD**

Internal Examiner **External Examiner**

ABSTRACT

Children's Mortality alludes to mortality of children younger than 5. The kid death rate, in addition under-five death rate, alludes to the probability of biting the mud among birth and exactly 5 years recent. The mortality of kids in addition happens in embryo. The purpose is to analysis AI based mostly strategies for grouping of mortality vertebrate upbeat characterization brings concerning best truth. Low birth weight and low gestational age are associated with an increased risk of mortality. Preterm birth also increases the risks of several complications, which can increase the risk of death, or cause long-term morbidities with both individual and societal impacts. In this work, we use machine learning for prediction of neonatal mortality as well as neonatal morbidities of bronchopulmonary dysplasia, necrotizing enterocolitis and retinopathy of prematurity, among very low birth weight infants. This paper proposes a machine learning-based approach for predicting child mortality and compares various machine learning methods against the provided dataset.

ACKNOWLEDGEMENT

It is my pleasure to express with deep sense of gratitude to Prof. Thandeeswaran R, Professor Grade 1, School of Information Technology and Engineering, Vellore Institute of Technology, for his constant guidance, continual encouragement, understanding; more than all, he taught me patience in my endeavor. My association with him is not confined to academics only, but it is a great opportunity on my part of work with an intellectual and expert in the field of IT.

I would like to express my gratitude to DR.G.VISWANATHAN, Chancellor VELLORE INSTITUTE OF TECHNOLOGY, VELLORE, MR.SANKAR VISWANATHAN, DR.SEKAR VISWANATHAN, MR.G V SELVAM, Vice - Presidents VELLORE INSTITUTE OF TECHNOLOGY, VELLORE, DR.RAMBABU KODALI, Vice - Chancellor, DR.S.NARAYANAN, Pro - Vice Chancellor and Dr.S.Sumathy, Dean, School of Information Technology & Engineering (SITE), for providing with an environment to work in and for his inspiration during the tenure of the course.

In jubilant mood I express ingeniously my whole-hearted thanks to Dr.Shantharajah S.P, Program chair & HoD/Professor, all teaching staff and members working as limbs of our university for their not-self-centered enthusiasm coupled with timely encouragements showered on me with zeal, which prompted the acquirement of the requisite knowledge to finalize my course study successfully. I would like to thank my parents for their support.

It is indeed a pleasure to thank my friends who persuaded and encouraged me to take up and complete this task. At last but not least, I express my gratitude and appreciation to all those who have helped me directly or indirectly toward the successful completion of this project.

Place: Vellore **Name of the student**

Date: 04.04.2023 **Harish R (18MIS0384)**

TABLE OF CONTENTS

LIST OF FIGURES..vii

LIST OF TABLES..viii

LIST OF ABBREVATION..ix

CHAPTER 1

INTRODUCTION

1.1 BACKGROUND..1
1.2 MOTIVATION..1
1.3 PROJECT STATEMENT..1
1.4 OBJECTIVES..2
1.5 SCOPE OF THE PROJECT..2

CHAPTER 2

LITERATURE SURVEY

2.1 SUMMARY OF THE EXISTING WORKS..3
2.2 CHALLENGES PRESENT IN EXISTING SYSTEM..11

CHAPTER 3

REQUIREMENTS

3.1 HARDWARE REQUIREMENTS..12
3.2 SOFTWARE REQUIREMENTS..12
3.3 GANTT CHART..13

CHAPTER 4

ANALYSIS & DESIGN

4.1 PROPOSED METHODOLOGY..14
4.2 SYSTEM ARCHITECTURE..14
4.3 MODULE DESCRIPTION..15
4.4 ALGORITHMS USED..16

CHAPTER 5

IMPLEMENTATION & TESTING

5.1 SAMPLE CODE..20
5.2 SAMPLE OUTPUT...31
5.3 TEST PLAN & DATA VERFICATION...38

CHAPTER 6

RESULTS

6.1 RESEARCH FINDINGS...39
6.2 RESULT ANALYSIS & EVALUATION METRICS...39

CONCLUSION AND FUTURE WORK..41

REFERENCES...42

LIST OF FIGURES

3.4 GANTT CHART...13
4.2 SYSTEM ARCHITECTURE...14
4.3 MODULE DIAGRAM...16
4.4.1 XGBOOST...17
4.4.2 RANDOM FOREST..19

LIST OF TABLES

2.1 SUMMARY OF THE EXISTING WORK...3
5.3 TEST PLAN AND DATA VERIFICATION...38
6.2 RESULT ANALYSIS AND EVALUATION MATRICS....................................39

LIST OF ABBREVATION

S.No	ABBREVATION	EXPANSION
01	XG Boost	Extreme Gradient Boosting
02	RF	Random Forest

Chapter 1

Introduction

1.1 BACKGROUND

Child mortality, also known as under-5 mortality, refers to the death of children under the age of five. It is a major public health concern in many parts of the world, particularly in low- and middle-income countries. According to the World Health Organization, in 2019, an estimated 5.2 million children under the age of five died, with more than half of these deaths occurring in sub-Saharan Africa and Southern Asia. The leading causes of under-5 mortality are pneumonia, diarrhoea, and malaria, as well as complications during childbirth.

Overall, the use of machine learning in predicting child mortality has the potential to improve healthcare outcomes and save lives, particularly in resource-limited settings where access to healthcare may be limited.

1.2 MOTIVATION

The high rate of child mortality in many parts of the world is a major public health concern. Despite significant progress in reducing under-5 mortality rates in recent decades, an estimated 5.2 million children under the age of five still die each year, with more than half of these deaths occurring in sub-Saharan Africa and Southern Asia.

The motivation behind the project is to leverage the power of machine learning to address a critical public health issue and improve the health outcomes of vulnerable populations, especially in low- and middle-income countries.

1.3 PROJECT STATEMENT

In preceding research focused on predicting neonatal and perinatal mortality using ML approaches, researchers reduced the risk of bias and improve predictive accuracy. Collected data may contain missing values which may lead to inconsistencies. To get better results, the data should be pre-processed to improve the efficiency of the algorithm. Outliers should be removed and mutable conversions should also be performed.

1.4 OBJECTIVES

In this project proposed a new and more efficient algorithm that produces solutions which are very close to the optimal ones. Our contribution is efficient not only for the bursting of behavior-based compositions but also for model-based compositions of services.

The ML models have potentials to perform better than the traditional statistical models because their ability to deal with non-linear complex data, multiple interactions between determinants and handle multiple factors and chain of events simultaneously. Also, ML is a prediction method that importantly determine not only who are a high risk to be died but also when the women and infants are at a higher risk.

1.5 SCOPE OF THE PROJECT

The relevant data that are needed to find the mortality rate are identified and the data has been pre-processed and trained to predict the mortality rate using machine learning algorithm. Different algorithms performed and accuracy has been compared to get the best model to predict the child mortality rate.

Chapter 2

Literature Survey

2.1 SUMMARY OF THE EXISTING WORKS

TITLE	AUTHOR	DESCRIPTION	YEAR
Child Mortality Prediction using Machine Learning	Samireddy Adithya; Tudimaladinna Sanjay Rezinald; A. Viji Amutha Mary; J. Refonaa; S. L. Jany Shabu; P. Jeyanthi	Children under the age of five are considered to be mortal in this context. The death rate for children under the age of five, or the under-five death rate, refers to the likelihood of dying between the ages of birth and the age of five. The death of a fetus is just as common as the death of a kid. The goal is to study AI-based strategies for determining the mortality fetal well-being arrangement that provides the best precision. It will be necessary to examine the entire dataset using the SMLT regulated AI strategy in order to identify the few data points that are similar to variable identification, univariate investigation, bivariate investigation, and multivariate investigation, as well as missing worth medicines and dissect information approval, cleaning/getting ready, and information perception. Using the results of this study, a complete approach has been developed to sensitivity analysis for model parameters that affect fetal health categorization. This paper proposes a machine learning-based approach for	2022

		predicting child mortality and compares various machine learning methods against the provided dataset.	
Implementation of a Predictive Model for Skilled Child Delivery Service use in Afghanistan	Nasratullah Nasrat; Mohammad Dawood Babakerkhell; Gawhar Shah Gawhari; Abdul Rahim Ahmadi	Institutional delivery during childbirth is essential to reduce both maternal and child mortality. Nevertheless, in Afghanistan, which has a high rank of maternal and child mortality around the globe, the number of childbirth attended by skilled birth attendants (SBAs) in health facilities remains extremely low. Therefore, the ability to predict the skilled child delivery service use is helpful and an excellent preventive measure. This study aims to develop a web-based skilled child delivery service use predictive model using data mining classification algorithms and identify the most suitable classifier among the four well-known machine learning algorithms. These are Random Forest (RF), Support Vector Machine (SVM), Artificial Neural Network (ANN), and PART rule induction. Waikato Environment for Knowledge Analysis (WEKA) version 3.8.4 was used to develop optimal models. The dataset used is the Afghanistan Demographic and Health Survey (AfDHS). The classification in this study comprises two categories, 'Skilled delivery' and 'No skilled delivery'. Preparation of the dataset is carefully done to ensure well-balanced samples in each category. The validation of	2021

		the predictive models is assessed by means of Accuracy, Precision, Recall, and area under Receiver Operating Characteristics (ROC) curve. The study found that Random Forest is the best classifier with an accuracy, precision, recall, and the area under ROC of 84.23%, 84.40%, 84.20% and 91.70% respectively. Subsequent to developing an optimal predictive model, we relied on this model to develop a web-based mobile application system for skilled child delivery service use prediction. Thus, the result can help decide targeted interventions for pregnant women to ensure skilled assistance at child delivery.	
Machine Learning Methods for Neonatal Mortality and Morbidity Classification	Joel Jaskari; Janne Myllärinen; Markus Leskinen; Ali Bahrami Rad; Jaakko Hollmén; Sture Andersson; Simo Särkkä	Preterm birth is the leading cause of mortality in children under the age of five. In particular, low birth weight and low gestational age are associated with an increased risk of mortality. Preterm birth also increases the risks of several complications, which can increase the risk of death, or cause long-term morbidities with both individual and societal impacts. In this work, we use machine learning for prediction of neonatal mortality as well as neonatal morbidities of bronchopulmonary dysplasia, necrotizing enterocolitis, and retinopathy of prematurity, among very low birth weight infants. Our predictors include time series data and clinical variables collected at the	2020

		neonatal intensive care unit of Children's Hospital, Helsinki University Hospital. We examine 9 different classifiers and present our main results in AUROC, similar to our previous studies, and in F1-score, which we propose for classifier selection in this study. We also investigate how the predictive performance of the classifiers evolves as the length of time series is increased, and examine the relative importance of different features using the random forest classifier, which we found to generally perform the best in all tasks. Our systematic study also involves different data pre-processing methods which can be used to improve classifier sensitivities. Our best classifier AUROC is 0.922 in the prediction of mortality, 0.899 in the prediction of bronchopulmonary dysplasia, 0.806 in the prediction of necrotizing enterocolitis, and 0.846 in the prediction of retinopathy of prematurity. Our best classifier F1-score is 0.493 in the prediction of mortality, 0.704 in the prediction of bronchopulmonary dysplasia, 0.215 in the prediction of necrotizing enterocolitis, and 0.368 in the prediction of retinopathy of prematurity.	
Early Prediction of Sepsis from Clinical Data Using Artificial Intelligence	R Murat Demirer; Oya Demirer	Sepsis is a major cause of death in the world. World Health Organization estimates 30 million people developing sepsis and 6 million people die	2019

		from sepsis each year; an estimated 4.2 million newborns and children are affected. The mortality rate is highest in septic shock in poor and developing countries. Early prediction of sepsis is critical for improving sepsis outcomes. The late prediction of sepsis in non-sepsis patients is a challenging problem. The aim of this study is to develop an artificial intelligence-based early warning and therapeutic decision support system which reduces sepsis-associated hospital mortality. We propose two compatible Boolean switchable Partially Observable Markov Decision Processes (POMDP) under a general risk-sensitive optimization criterion with finite time horizon. It is based on Spectral analysis of unevenly sampled (missing) observations with Demographics, Vital Signs, and Laboratory values for the patient. The policy is a common mixture of sepsis and non-sepsis beliefs on own utility functions which favors to achieve Pareto Optimality from this high dimensional belief space.	
Modification of MELD score by including Serum Albumin to improve prediction of mortality outcome of cirrhotic patient based on Thai	Montri Duangkrut; Yaowadee Temtanapat; Piyawat Komolmit	Nowadays, the Model for End-stage Liver Disease (MELD) has become a popular model and replaced the Child-Pugh score for the assessment of the mortality opportunity of patients with cirrhosis in 3-month period. The model predicts the severity of the disease based on 3	2014

cirrhotic patients		biochemical parameters: serum creatinine, serum total bilirubin, and INR. However, in the past, the first model like Child-Pugh score signified the importance of Serum Albumin, a protein producing in a liver. It is, thus, expected that the Serum Albumin has an effect on patients' mortality prediction. In this research, our main focus is to refine and evaluate the effect of Serum Albumin to mortality of Thai cirrhotic patients if included into the MELD model. We use the data collection from 158 Thai cirrhotic patients with different degrees of severity. They were treated at the Liver Unit and Clinic, King Chulalongkorn Memorial Hospital, The Thai Red Cross Society. The collected data were divided into the periods of 3 months, 6 months, 1 year and 2 years respectively. The Kaplan-Meier statistic was used to analyze the survival opportunity of each period. Also, the Cox-Regression was utilized to evaluate the relationship and the statistical significance of the substance in each period in order to find the connection between the Serum Albumin and mortality opportunities. Results of the study show that of all the data from 158 patients, with the Serum Albumin level between 1.0 and 3.5 g/dL, when tested by Pearson's Chi-squared[2], Log Rank Test and Wilcoxon rank-sum (Mann-Whitney)[3]	

		has the statistical significance at the 1% level of confidence (p <; 0.001). Moreover, the correlation of the results using Cox Regression demonstrated also that Serum Albumin influenced the mortality opportunity at the hazard ratio of 5.14 (95%CI:2.971-8.920) with level of confidence p-value <; 0.0001. Thus, we believe that the Serum Albumin affected the mortality prediction model. We also propose two refined MELD models, ThaiMELD-Albumin and ThaiMELD-CTP[5]. For the efficiency assessment of the models, we compare our models to others using the ROC. We found that ThaiMELD-Albumin had 0.85 (95% CI: 0.68-1.00) and it is better than MELD, MELD-Albumin and 5vMELD, while ThaiMELD-CTP is just better than MELD. Consequently, ThaiMELD-Albumin is better for prediction of the mortality opportunity for Thai patients than the MELD, MELD-Albumin or 5vMELD. While ThaiMELD-CTP which just added a scale value to MELD could give a better assessment than MELD itself. Therefore, our model could benefit to Thai patients for the assessment of mortality opportunity as well as symptoms' severity. It could, perhaps, be further used for the consideration of liver transplantation in Thailand.	
A Knowledge	Vikraman	The Neonatal Intensive Care	2011

Management Based Approach for Mortality Prediction in the Neonatal Intensive Care Unit	Baskaran; Irene Bajan; Bharat Shah; Franklyn I. Prescod; Andrew James	Unit (NICU) is one of the most information sensitive environments where efforts are made continuously to deliver the optimal health-care for fragile, critically ill patients. NICU health care providers employ cutting edge clinical processes, technologies, and latest tools and techniques to provide care for critically ill new born infants. Research has shown that predicting an infant's mortality is important in making critical care decisions. Contemporary, 21st century neonatal intensive care involves the active participation of parents. Although, there are neonatal scores that can be used to measure severity of illness, they are complex and are difficult to comprehend for a novice. Hence, there is a need to combine all the care related information to obtain an indication of the newborn infant's state of health. A new score for Neonatal Mortality Prediction (NMPS) is proposed in this paper. This NMPS would provide an easy to understood numeric value that can be comprehended by both neonatal health care providers and parents. The NMPS would employ factors captured from the antenatal, perinatal and neonatal periods for estimating a consolidated score.	

2.2 CHALLENGES PRESENT IN EXISTING SYSTEM

We present a novel data-driven modelling approach for post-ICU mortality prediction based on the temporal dynamics of patient condition during ICU stay. Specifically, we develop a new switching state-space model that coverts and fuses various patient variables into a SAPS II-based sequence to represent patient condition. We further model the relationship of in-ICU patient condition and Post-ICU patient survival using a logistic regression model.

Chapter 3

Requirements

3.1 HARDWARE REQUIREMENTS

The hardware requirements may serve as the basis for a contract for the implementation of the system and should therefore be a complete and consistent specification of the whole system. They are used by software engineers as the starting point for the system design. It shows what the system does and not how it should be implemented.

PROCESSOR	:	Intel I5
RAM	:	4GB
HARD DISK	:	50 GB

3.2 SOFTWARE REQUIREMENTS

The software requirements document is the specification of the system. It should include both a definition and a specification of requirements. It is a set of what the system should do rather than how it should do it. The software requirements provide a basis for creating the software requirements specification. It is useful in estimating cost, planning team activities, performing tasks and tracking the team's and tracking the team's progress throughout the development activity.

Operating system : Windows 7 (Service Pack 1), 8, 8.1 and 10

Front End : Django Framework

Coding Language : Python

Backend : Python

Software Tool : Anaconda Jupyter Notebook

3.3 GANTT CHART

Activity	Description of the Activity	Guide Remarks
01	Review 0 (Title selection)	Ok
02	Review 1 (Architecture)	Datasets should be clear
03	Review 2 (Implementation)	Guided for the implementation

Chapter 4

Analysis & Design

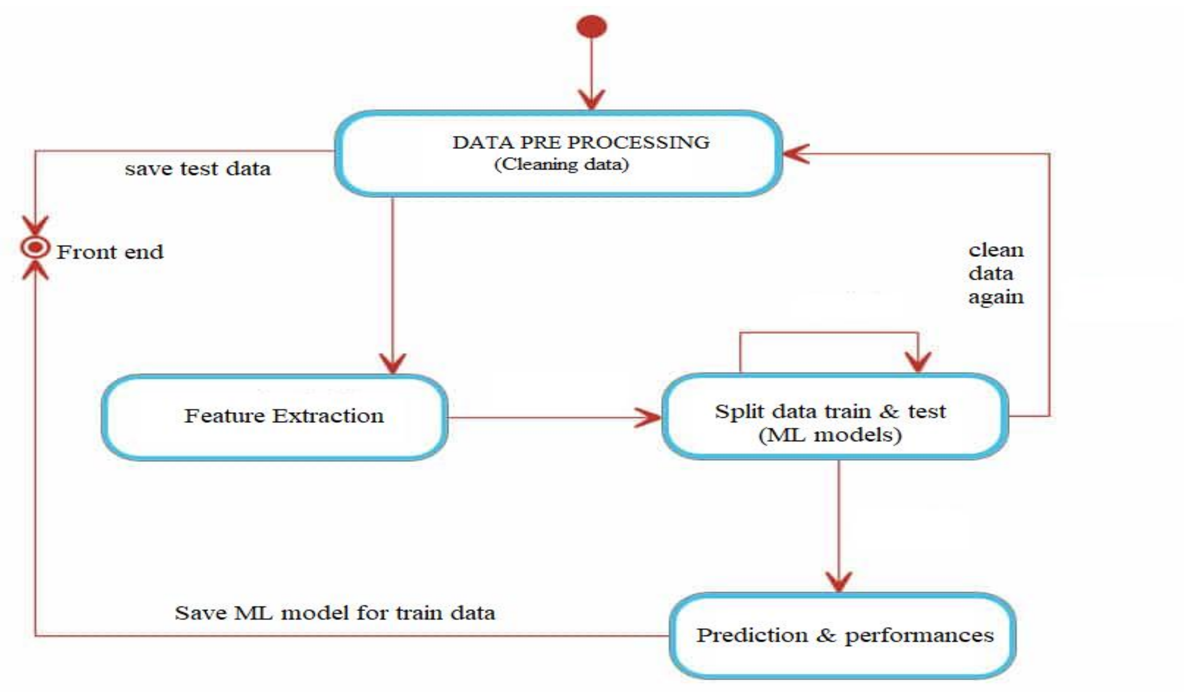
4.1 PROPOSED METHODOLOGY

To get better results, the data should be pre-processed to improve the efficiency of the algorithm. Outliers should be removed and mutable conversions should also be performed. The data set collected to predict the given data is divided into training set and test set. The data model created using machine learning algorithms is applied to the training set, and based on the accuracy of the test results, the prediction of the test set is made. The model can classify mortality. Extreme Gradient Boosting Classifier machine learning algorithms can be compared and the best algorithm can be used for classification.

4.1.1 ADVANTAGES OF PROPOSED SYSTEM

- Accuracy of prediction is high
- Time for doing analysis is very less

4.2 SYSTEM ARCHITECTURE



4.3 MODULE DESCRIPTION

- Data collection
- Data pre processing
- Data splitting
- Training and testing

4.3.1 DATA COLLECTION:

It's time for a data analyst to pick up the baton and lead the way to machine learning implementation. The job of a data analyst is to find ways and sources of collecting relevant and comprehensive data, interpreting it, and analyzing results with the help of statistical techniques.

The type of data depends on what you want to predict. There is no exact answer to the question "How much data is needed?" because each machine learning problem is unique. In turn, the number of attributes data scientists will use when building a predictive model depends on the attributes' predictive value.

4.3.2 DATA PRE PROCESSING:

We loaded the data set as pandas data frame to process the data set and load it in the machine learning model. In this experiment we dropped the null values.

4.3.2.1 LABEL ENCODING:

Label Encoding refers to converting the labels into a numeric form so as to convert them into the machine-readable form. Machine learning algorithms can then decide in a better way how those labels must be operated. It is an important pre-processing step for the structured dataset in supervised learning.

4.3.3 DATA SPLITTING:

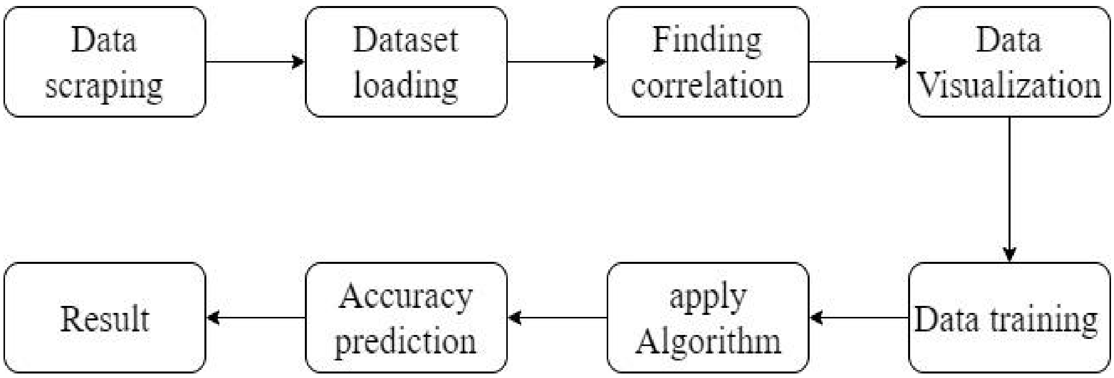
For each experiment, we split the entire dataset into 70% training set and 30% test set. We used the training set for resampling, hyper parameter tuning, and training the model and we used test set to test the performance of the trained model. While splitting the data, we specified a random seed (any random number), which ensured the same data split every time the program executed.

4.3.4 TRAINING AND TESTING:

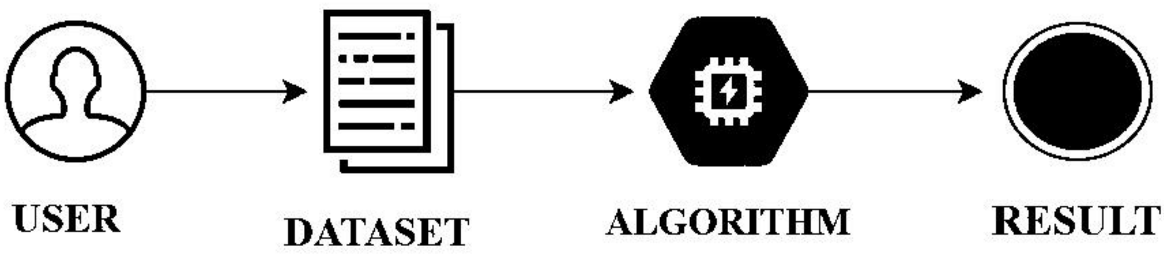
Algorithms learn from data. They find relationships, develop understanding, make decisions, and evaluate their confidence from the training data they're given. And the better the training data is, the better the model performs.

In fact, the quality and quantity of your training data has as much to do with the success of your data project as the algorithms themselves.

4.3.5 MODULE DIAGRAM



FRONT END MODULE DIAGRAMS:



4.4 ALGORITHM USED

- XGBoost
- Random Forest

4.4.1 XGBOOST:

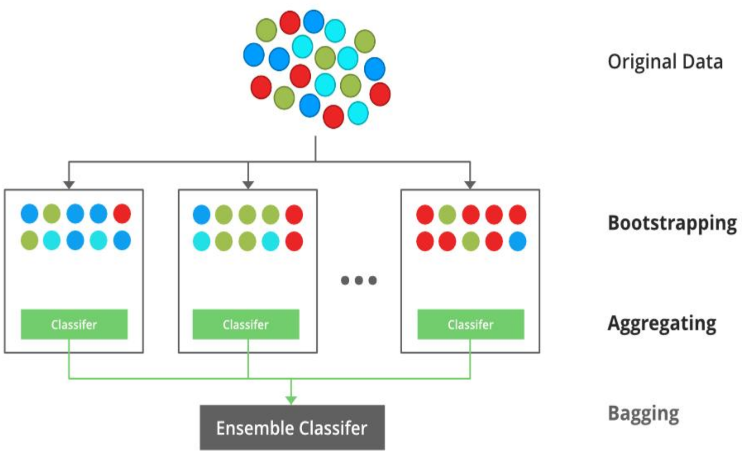
XGBoost stands for Extreme Gradient Boosting, which was proposed by the researchers at the University of Washington. It is a library written in C++ which optimizes the training for Gradient Boosting.

A tree can be "learned" by splitting the source set into subsets based on an attribute value test. This process is repeated on each derived subset in a recursive manner called recursive partitioning. The recursion is completed when the subset at a node all has the same value of the target variable, or when splitting no longer adds value to the predictions.

4.4.1.1 BAGGING:

A Bagging classifier is an ensemble meta-estimator that fits base classifiers each on random subsets of the original dataset and then aggregate their individual predictions (either by voting or by averaging) to form a final prediction. Such a meta-estimator can typically be used as a way to reduce the variance of a black-box estimator (e.g., a decision tree), by introducing randomization into its construction procedure and then making an ensemble out of it. Each base classifier is trained in parallel with a training set which is generated by randomly drawing, with replacement, N examples (or data) from the original training dataset, where N is the size of the original training set. The training set for each of the base classifiers is independent of each other. Many of the original data may be repeated in the resulting training set while others may be left out.

Bagging reduces overfitting (variance) by averaging or voting, however, this leads to an increase in bias, which is compensated by the reduction in variance though.



4.4.1.2 BOOSTING:

Boosting is an ensemble modelling, technique that attempts to build a strong classifier from the number of weak classifiers. It is done by building a model by using weak models in series. Firstly, a model is built from the training data. Then the second model is built which tries to correct the errors present in the first model. This procedure is continued and models are added until either the complete training data set is predicted correctly or the maximum number of models are added.

4.4.1.3 GRADIENT BOOSTING:

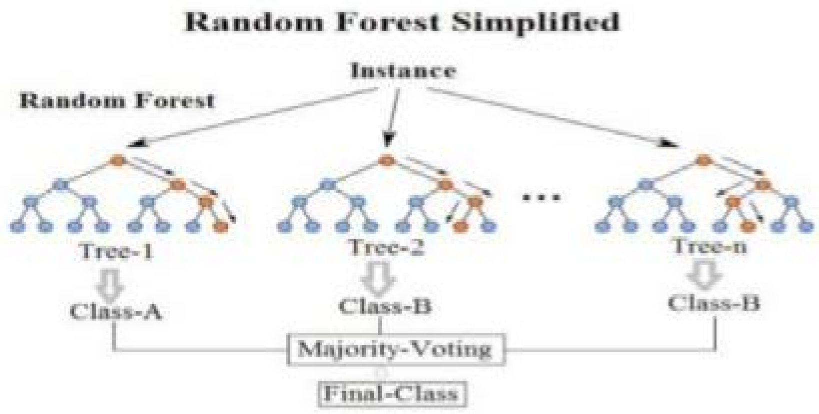
Gradient Boosting is a popular boosting algorithm. In gradient boosting, each predictor corrects its predecessor's error. In contrast to Adaboost, the weights of the training instances are not tweaked, instead, each predictor is trained using the residual errors of predecessor as labels.

There is a technique called the Gradient Boosted Trees whose base learner is CART (Classification and Regression Trees).

4.4.2 RANDOM FOREST:

Random Forest algorithm is a machine learning based algorithm that combines multiple decision trees together for obtaining efficient outcome. Decision trees are created by random forest algorithm based on data samples and selects the best solution by means of voting.

Random Forest algorithms are used for classification as well as regression. It creates a tree for the data and makes prediction based on that. Random Forest algorithm can be used on large datasets and can produce the same result even when large sets record values are missing. The generated samples from the decision tree can be saved so that it can be used on other data. In random forest there are two stages, firstly create a random forest then make a prediction using a random forest classifier created in the first stage.



The random forest is a supervised learning algorithm that randomly creates and merges multiple decision trees into one “forest.” The goal is not to rely on a single learning model, but rather a collection of decision models to improve accuracy. The primary difference between this approach and the standard decision tree algorithm is that the root nodes feature splitting nodes are generated randomly.

Chapter 5

Implementation & Testing

5.1 SAMPLE CODE

BACKEND:

```
import pandas as pd
import numpy as np
import pickle
import matplotlib.pyplot as plt
import geopandas as gpd

%matplotlib inline

df = pd.read_csv(r'C:\Users\PAVITHRA\Music\ITML35-Child
mortality\DATASET\ChildMOrtalytRate.csv')

df

df.columns

df.drop('Unnamed: 0', axis =1,inplace=True)

df

df['Gender'].value_counts()

# changing value name
df['Gender'].replace({'Total': 'Others'},inplace=True)

df['Gender'].value_counts()

df['Country'].value_counts()

df.info()

df = df.dropna()

df.info()
```

```
df['Gender'].unique()

plt.pie(df['Gender'].value_counts(),labels=df['Gender'].unique(),startangle = 90,
shadow = True, autopct='%1.2f%%')
plt.legend()
plt.title('Genders Chart')
plt.show()

from sklearn.preprocessing import LabelEncoder

le = LabelEncoder()

df['Country']=le.fit_transform(df['Country'])
df['Gender']=le.fit_transform(df['Gender'])

df

import seaborn as sns
sns.distplot(df['Mortality Rate'])

X = df.drop('Mortality Rate', axis =1)

X

y = df['Mortality Rate']

y

from sklearn.model_selection import train_test_split
X_train,X_test,y_train,y_test = train_test_split(X,y,test_size=0.30,random_state=40)

X_test

y_test

y_train

X_test.to_csv('test.csv',index=False)

#random forest regressor
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_squared_error
regressor=RandomForestRegressor(criterion='mse',n_estimators=180)
```

```
#fit the model
regressor.fit(X_train,y_train)

#create the predict model
y_pred1=regressor.predict(X_test)

from sklearn import metrics

# Model Evaluation
print('R^2:', metrics.r2_score(y_test, y_pred1))
print('Adjusted R^2:',1 - (1-metrics.r2_score(y_test,
y_pred1))*(len(y_test)-1)/(len(y_test)-X_train.shape[1]-1))
print('MAE:',metrics.mean_absolute_error(y_test, y_pred1))
print('MSE:',metrics.mean_squared_error(y_test, y_pred1))
print('RMSE:',np.sqrt(metrics.mean_squared_error(y_test, y_pred1)))

import numpy as np
rf_compare = pd.DataFrame({'Real Values':y_test, 'Predicted Values': y_pred1})
rf_compare.head(10)

import xgboost as xgb
from xgboost import XGBRegressor
xgb = XGBRegressor(n_estimators=180,max_depth=100)


#fit the model
xgb.fit(X_train,y_train)

#create the predict model
y_pred2=xgb.predict(X_test)

# Model Evaluation
print('R^2:', metrics.r2_score(y_test, y_pred2))
print('Adjusted R^2:',1 - (1-metrics.r2_score(y_test,
y_pred2))*(len(y_test)-1)/(len(y_test)-X_train.shape[1]-1))
print('MAE:',metrics.mean_absolute_error(y_test, y_pred2))
print('MSE:',metrics.mean_squared_error(y_test, y_pred2))
print('RMSE:',np.sqrt(metrics.mean_squared_error(y_test, y_pred2)))

xg_compare = pd.DataFrame({'Real Values':y_test, 'Predicted Values': y_pred2})
xg_compare.head(10)
```

FRONTEND:

```
{% load static %}

<!DOCTYPE html>
<!--
Template Name: Shiphile
Author: <a href="https://www.os-templates.com/">OS Templates</a>
Author URI: https://www.os-templates.com/
Copyright: OS-Templates.com
Licence: Free to use under our free template licence terms
Licence URI: https://www.os-templates.com/template-terms
-->
<html lang="">
<!-- To declare your language - read more here: https://www.w3.org/International/questions/qa-html-language-declarations -->
<head>
<title>Children mortality prediction </title>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0, maximum-scale=1.0, user-scalable=no">
<link href="{% static 'layout/styles/layout.css' %}" rel="stylesheet" type="text/css" media="all">
</head>
<body id="top">
<!--
############################################################################################
#################################### -->
<!--
############################################################################################
#################################### -->
<!--
############################################################################################
#################################### -->
<!-- Top Background Image Wrapper -->
<div class="bgded overlay" style="background-image:url('{% static 'images/demo/backgrounds/1.jpg' %}');">
  <!--
############################################################################################
#################################### -->

  <!--
############################################################################################
#################################### -->
```

```
   <!--
########################################################################
############################ -->

   <!--
########################################################################
############################ -->
   <div id="pageintro" class="hoc clear">
      <!--
########################################################################
############################ -->
      <article>
         <h3 class="heading"> Children mortality prediction </h3>
         <p> Enter the details
         </p>



      <footer>
         <form action='input' method="POST" enctype="multipart/form-data" class="login100-form validate-form">
            {% csrf_token %}




                  <ul class="nospace inline pushright">
                     <li>
                       <div class="wrap-input100 validate-input" data-validate = "Valid email is required: ex@abc.xyz">
               <input class="input100" type="text" name="name" style = "color: black;">
               <span class="focus-input100"></span>
               <span class="label-input100">Name</span>
            </div>
            <input class="input100" type="int" name="password" style = "color: black;">
            <span class="focus-input100"></span>
            <span class="label-input100">Password</span>

                     <footer><button type="submit" class="btn" href="#"> LOGIN </button></footer>
                  </ul>
                  </form>
      <!--
########################################################################
```

```
########################### -->
  </div>
  <!--
#####################################################################
########################### -->
</div>
<!-- End Top Background Image Wrapper -->
<!--
#####################################################################
########################### -->
<!--
#####################################################################
########################### -->
<!--
#####################################################################
########################### -->

<!--
#####################################################################
########################### -->
<!--
#####################################################################
########################### -->

<!--
#####################################################################
########################### -->

<!--
#####################################################################
########################### -->

<!--
#####################################################################
########################### -->
<!--
#####################################################################
########################### -->
<a id="backtotop" href="#top"><i class="fas fa-chevron-up"></i></a>
<!-- JAVASCRIPTS -->
<script src="{% static 'layout/scripts/jquery.min.js' %}"></script>
<script src="{% static 'layout/scripts/jquery.backtotop.js' %}"></script>
<script src="{% static 'layout/scripts/jquery.mobilemenu.js' %}"></script>
</body>
```

```
</html>

{% load static %}

<!DOCTYPE html>
<!--
Template Name: Shiphile
Author: <a href="https://www.os-templates.com/">OS Templates</a>
Author URI: https://www.os-templates.com/
Copyright: OS-Templates.com
Licence: Free to use under our free template licence terms
Licence URI: https://www.os-templates.com/template-terms
-->
<html lang="">
<!-- To declare your language - read more here:
https://www.w3.org/International/questions/qa-html-language-declarations -->
<head>
<title> Child mortality prediction</title>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0,
maximum-scale=1.0, user-scalable=no">
<link href="{% static 'layout/styles/layout.css' %}" rel="stylesheet" type="text/css"
media="all">
</head>
<body id="top">
<!--
#####################################################################
########################### -->

<!--
#####################################################################
########################### -->
<!-- Top Background Image Wrapper -->
<div class="bgded overlay" style="background-image:url('{% static
'images/demo/backgrounds/2.jpg' %}');">
  <!--
#####################################################################
########################### -->



  <!--
#####################################################################
########################### -->
```

```
<div id="pageintro" class="hoc clear">
     <!--
##########################################################################
############################ -->
     <article>
        <h3 class="heading">Child mortality </h3>
        <p> </p>

     </article>
     <!--
##########################################################################
############################ -->
     <form action='output' method="POST" enctype="multipart/form-data"
class="login100-form validate-form">
        {% csrf_token %}



              <ul class="nospace inline pushright">
                 <li>
                  <div class="wrap-input100 validate-input" data-validate = "Valid
email is required: ex@abc.xyz">
           <input class="input100" type="row" name="row" style = "color: black;">
           <span class="focus-input100"></span>
           <span class="label-input100">ID</span>
        </div>




                 </select>
                 <span class="label-input100">Algorithm</span>

                 <div class="wrap-input100 validate-input" data-validate="Password is
required">
                            <select class="input100" name="algo" style = "color:
black;">

                            <option value='xgb'>XgBOOST</option>
                            <option value='rf'>Random Forest</option>

                            </select>

                    </div>
```

```
                <footer><button type="submit" class="btn"
href="#">Predict</button></footer>
              </ul>
              </form>


    </div>
    <!--
################################################################################
############################### -->
</div>
<!-- End Top Background Image Wrapper -->

<!--
################################################################################
############################### -->
<a id="backtotop" href="#top"><i class="fas fa-chevron-up"></i></a>
<!-- JAVASCRIPTS -->
<script src="{% static 'layout/scripts/jquery.min.js' %}"></script>
<script src="{% static 'layout/scripts/jquery.backtotop.js' %}"></script>
<script src="{% static 'layout/scripts/jquery.mobilemenu.js' %}"></script>
</body>
</html>

{% load static %}

<!DOCTYPE html>
<!--
Template Name: Shiphile
Author: <a href="https://www.os-templates.com/">OS Templates</a>
Author URI: https://www.os-templates.com/
Copyright: OS-Templates.com
Licence: Free to use under our free template licence terms
Licence URI: https://www.os-templates.com/template-terms
-->
<html lang="">
<!-- To declare your language - read more here:
https://www.w3.org/International/questions/qa-html-language-declarations -->
<head>
<title>Child mortality prediction </title>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0,
maximum-scale=1.0, user-scalable=no">
```

```
<link href="{% static 'layout/styles/layout.css' %}" rel="stylesheet" type="text/css"
media="all">
</head>
<body id="top">
<!--
#####################################################################
########################### -->
<!--
#####################################################################
########################### -->
<!--
#####################################################################
########################### -->
<!-- Top Background Image Wrapper -->
<div class="bgded overlay" style="background-image:url('{% static
'images/demo/backgrounds/3.jpg' %}');">
  <!--
#####################################################################
########################### -->

  <!--
#####################################################################
########################### -->
  <!--
#####################################################################
########################### -->

  <!--
#####################################################################
########################### -->
  <div id="pageintro" class="hoc clear">
    <!--
#####################################################################
########################### -->
    <article>
      <h3 class="heading">   Child Mortality rate </h3>

      <footer><a class="btn" href="#">{{out}}</a></footer>
    </article>
    <!--
#####################################################################
########################### -->
  </div>
  <!--
```

```
############################################################################
############################# -->
</div>
<!-- End Top Background Image Wrapper -->
<!--
############################################################################
############################# -->
<!--
############################################################################
############################# -->
<!--
############################################################################
############################# -->

<!--
############################################################################
############################# -->
<!--
############################################################################
############################# -->



<!--
############################################################################
############################# -->

<!--
############################################################################
############################# -->

<!--
############################################################################
############################# -->
<!--
############################################################################
############################# -->
<a id="backtotop" href="#top"><i class="fas fa-chevron-up"></i></a>
<!-- JAVASCRIPTS -->
<script src="{% static 'layout/scripts/jquery.min.js' %}"></script>
<script src="{% static 'layout/scripts/jquery.backtotop.js' %}"></script>
<script src="{% static 'layout/scripts/jquery.mobilemenu.js' %}"></script>
</body>
</html>
```

5.2 SAMPLE OUTPUT

In [1]:
```
import pandas as pd
import numpy as np
import pickle
import matplotlib.pyplot as plt
import geopandas as gpd


%matplotlib inline
```

In [2]:
```
df = pd.read_csv(r'C:\Users\PAVITHRA\Music\ITML35-Child mortality\DATASET\ChildMOrtalytRate.csv')
```

In [3]:
```
df
```

Out[3]:

	Unnamed: 0	Country	Year	Gender	Child Mortality(1 to 4)	Total Population	Mortality Rate
0	0	Afghanistan	1967	Female	26012.0	5080.813	5.119653
1	1	Afghanistan	1968	Female	26192.0	5202.606	5.034400
2	2	Afghanistan	1969	Female	26335.0	5333.936	4.937255
3	3	Afghanistan	1970	Female	26562.0	5476.630	4.850063
4	4	Afghanistan	1971	Female	26671.0	5630.099	4.737217
...	...	...	...	...	...	...	...
30935	30935	Zimbabwe	2015	Total	9031.0	13814.642	0.653727
30936	30936	Zimbabwe	2016	Total	8566.0	14030.338	0.610534
30937	30937	Zimbabwe	2017	Total	8318.0	14236.599	0.584269
30938	30938	Zimbabwe	2018	Total	7692.0	14438.812	0.532731
30939	30939	Zimbabwe	2019	Total	7397.0	14645.473	0.505071

30940 rows × 7 columns

In [4]:
```
df.columns
```

Out[4]:
```
Index(['Unnamed: 0', 'Country', 'Year', 'Gender', 'Child Mortality(1 to 4)',
       'Total Population', 'Mortality Rate'],
      dtype='object')
```

In [5]:
```
df.drop('Unnamed: 0', axis =1,inplace=True)
```

In [6]:
```
df
```

Out[6]:

	Country	Year	Gender	Child Mortality(1 to 4)	Total Population	Mortality Rate
0	Afghanistan	1967	Female	26012.0	5080.813	5.119653
1	Afghanistan	1968	Female	26192.0	5202.606	5.034400
2	Afghanistan	1969	Female	26335.0	5333.936	4.937255
3	Afghanistan	1970	Female	26562.0	5476.630	4.850063
4	Afghanistan	1971	Female	26671.0	5630.099	4.737217
...	...	...	...	...	...	...
30935	Zimbabwe	2015	Total	9031.0	13814.642	0.653727
30936	Zimbabwe	2016	Total	8566.0	14030.338	0.610534
30937	Zimbabwe	2017	Total	8318.0	14236.599	0.584269
30938	Zimbabwe	2018	Total	7692.0	14438.812	0.532731
30939	Zimbabwe	2019	Total	7397.0	14645.473	0.505071

30940 rows × 6 columns

In [7]:
```
df['Gender'].value_counts()
```

Out[7]:
```
Female    10362
Male      10362
Total     10216
Name: Gender, dtype: int64
```

In [8]:
```
# changing value name
df['Gender'].replace({'Total': 'Others'},inplace=True)
```

In [9]:
```
df['Gender'].value_counts()
```

Out[9]:
```
Female    10362
Male      10362
Others    10216
Name: Gender, dtype: int64
```

In [10]:
```
df['Country'].value_counts()
```

Out[10]:
```
Poland          195
Switzerland     195
Canada          195
Seychelles      195
Senegal         195
               ...
Timor-Leste      90
San Marino       90
Nauru            90
Andorra          90
Somalia          66
Name: Country, Length: 194, dtype: int64
```

In [11]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 30940 entries, 0 to 30939
Data columns (total 6 columns):
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   Country               30940 non-null  object
 1   Year                  30940 non-null  int64
 2   Gender                30940 non-null  object
 3   Child Mortality(1 to 4)  30940 non-null  float64
 4   Total Population      30064 non-null  float64
 5   Mortality Rate        30064 non-null  float64
dtypes: float64(3), int64(1), object(2)
memory usage: 1.4+ MB
```

In [12]:
```
df = df.dropna()
```

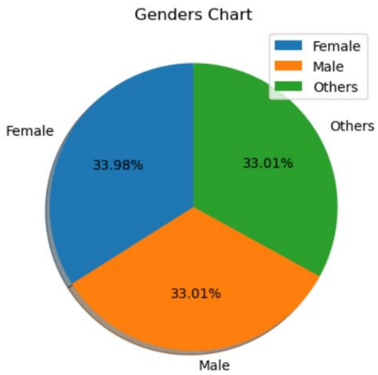
In [13]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 30064 entries, 0 to 30939
Data columns (total 6 columns):
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   Country               30064 non-null  object
 1   Year                  30064 non-null  int64
 2   Gender                30064 non-null  object
 3   Child Mortality(1 to 4)  30064 non-null  float64
 4   Total Population      30064 non-null  float64
 5   Mortality Rate        30064 non-null  float64
dtypes: float64(3), int64(1), object(2)
memory usage: 1.6+ MB
```

In [14]:
```
df['Gender'].unique()
```

Out[14]:
```
array(['Female', 'Male', 'Others'], dtype=object)
```

In [15]:
```
plt.pie(df['Gender'].value_counts(),labels=df['Gender'].unique(),startangle = 90, shadow = True, autopct='%1.2f%%')
plt.legend()
plt.title('Genders Chart')
plt.show()
```



In [16]:
```
from sklearn.preprocessing import LabelEncoder
```

In [17]:
```
le = LabelEncoder()
```

In [18]:
```
df['Country']=le.fit_transform(df['Country'])
df['Gender']=le.fit_transform(df['Gender'])
```

```
C:\Users\PAVITHRA\AppData\Local\Temp\ipykernel_21968\1679067845.py:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-ve
rsus-a-copy
  df['Country']=le.fit_transform(df['Country'])
C:\Users\PAVITHRA\AppData\Local\Temp\ipykernel_21968\1679067845.py:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-ve
rsus-a-copy
  df['Gender']=le.fit_transform(df['Gender'])
```

```
In [19]: df
```

Out[19]:

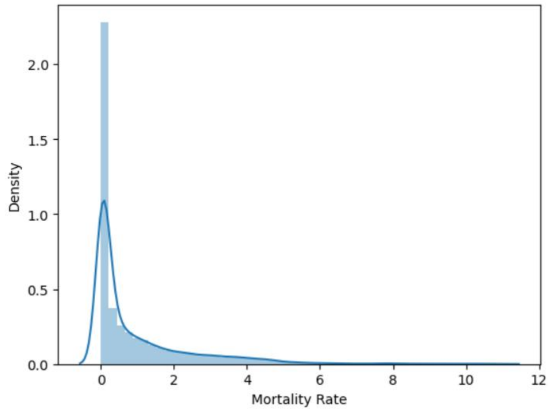
	Country	Year	Gender	Child Mortality(1 to 4)	Total Population	Mortality Rate
0	0	1967	0	26012.0	5080.813	5.119653
1	0	1968	0	26192.0	5202.606	5.034400
2	0	1969	0	26335.0	5333.936	4.937255
3	0	1970	0	26562.0	5476.630	4.850063
4	0	1971	0	26671.0	5630.099	4.737217
...	...	...	...	...	...	...
30935	193	2015	2	9031.0	13814.642	0.653727
30936	193	2016	2	8566.0	14030.338	0.610534
30937	193	2017	2	8318.0	14236.599	0.584269
30938	193	2018	2	7692.0	14438.812	0.532731
30939	193	2019	2	7397.0	14645.473	0.505071

30064 rows × 6 columns

```
In [20]: import seaborn as sns
sns.distplot(df['Mortality Rate'])
```

```
C:\Users\PAVITHRA\anaconda3\lib\site-packages\seaborn\distributions.py:2619: FutureWarning: `distplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

Out[20]: <AxesSubplot:xlabel='Mortality Rate', ylabel='Density'>




```
In [21]: X = df.drop('Mortality Rate', axis =1)
```

```
In [22]: X
```

Out[22]:

	Country	Year	Gender	Child Mortality(1 to 4)	Total Population
0	0	1967	0	26012.0	5080.813
1	0	1968	0	26192.0	5202.606
2	0	1969	0	26335.0	5333.936
3	0	1970	0	26562.0	5476.630
4	0	1971	0	26671.0	5630.099
...	...	...	...	...	...
30935	193	2015	2	9031.0	13814.642
30936	193	2016	2	8566.0	14030.338
30937	193	2017	2	8318.0	14236.599
30938	193	2018	2	7692.0	14438.812
30939	193	2019	2	7397.0	14645.473

30064 rows × 5 columns

```
In [23]: y = df['Mortality Rate']
```

```
In [24]: y
```

```
Out[24]: 0        5.119653
         1        5.034400
         2        4.937255
         3        4.850063
         4        4.737217
                    ...
         30935    0.653727
         30936    0.610534
         30937    0.584269
         30938    0.532731
         30939    0.505071
         Name: Mortality Rate, Length: 30064, dtype: float64
```

```
In [25]: from sklearn.model_selection import train_test_split
         X_train,X_test,y_train,y_test = train_test_split(X,y,test_size=0.30,random_state=40)
```

```
In [26]: X_test
```

Out[26]:

	Country	Year	Gender	Child Mortality(1 to 4)	Total Population
15188	93	1990	1	4072.0	2129.857
17322	106	1990	2	7.0	362.017
17728	109	1968	0	420.0	398.757
23539	146	2013	0	2.0	90.095
339	2	1967	1	29522.0	6661.827
...	...	...	...	...	...
17074	105	1989	1	22023.0	4099.150
25681	160	1988	1	11592.0	3525.266
8899	56	2008	1	971.0	1529.322
18798	116	1990	2	46789.0	12987.292
17331	106	1999	2	5.0	390.626

9020 rows × 5 columns

```
In [27]: y_test
```

```
Out[27]: 15188    1.911865
         17322    0.019336
         17728    1.053273
         23539    0.022199
         339      4.431517
                    ...
         17074    5.372577
         25681    3.288263
         8899     0.634922
         18798    3.602676
         17331    0.012800
         Name: Mortality Rate, Length: 9020, dtype: float64
```

```
In [28]: y_train
```

```
Out[28]: 1381     0.039121
         1288     0.007814
         9365     3.946414
         17790    0.968997
         29732    0.073802
                    ...
         24788    1.173230
         28516    3.637423
         14799    0.403736
         14853    0.324126
         11888    0.402256
         Name: Mortality Rate, Length: 21044, dtype: float64
```

```
In [29]: X_test.to_csv('test.csv',index=False)
```

```
In [30]: #random forest regressor
         from sklearn.ensemble import RandomForestRegressor
         from sklearn.metrics import mean_squared_error
         regressor=RandomForestRegressor(criterion='mse',n_estimators=180)

         #fit the model
         regressor.fit(X_train,y_train)


         #create the predict model
         y_pred1=regressor.predict(X_test)
```

```
C:\Users\PAVITHRA\anaconda3\lib\site-packages\sklearn\ensemble\_forest.py:396: FutureWarning: Criterion 'mse' was deprecated in
v1.0 and will be removed in version 1.2. Use `criterion='squared_error'` which is equivalent.
  warn(
```

In [31]:
```
from sklearn import metrics

# Model Evaluation
print('R^2:', metrics.r2_score(y_test, y_pred1))
print('Adjusted R^2:',1 - (1-metrics.r2_score(y_test, y_pred1))*(len(y_test)-1)/(len(y_test)-X_train.shape[1]-1))
print('MAE:',metrics.mean_absolute_error(y_test, y_pred1))
print('MSE:',metrics.mean_squared_error(y_test, y_pred1))
print('RMSE:',np.sqrt(metrics.mean_squared_error(y_test, y_pred1)))
```

```
R^2: 0.9993231889336186
Adjusted R^2: 0.9993228135114607
MAE: 0.01428187393289763
MSE: 0.0014948873852951936
RMSE: 0.03866377355219216
```

In [32]:
```
import numpy as np
rf_compare = pd.DataFrame({'Real Values':y_test, 'Predicted Values': y_pred1})
rf_compare.head(10)
```

Out[32]:

	Real Values	Predicted Values
15188	1.911865	1.918357
17322	0.019336	0.019883
17728	1.053273	1.039737
23539	0.022199	0.022208
339	4.431517	4.410968
14603	1.523055	1.502932
21945	0.152330	0.156243
20935	2.914837	2.918765
18792	3.768008	3.749586
1740	0.452978	0.445744

In [33]:
```
import xgboost as xgb
from xgboost import XGBRegressor
xgb = XGBRegressor(n_estimators=180,max_depth=100)


#fit the model
xgb.fit(X_train,y_train)



#create the predict model
y_pred2=xgb.predict(X_test)
```

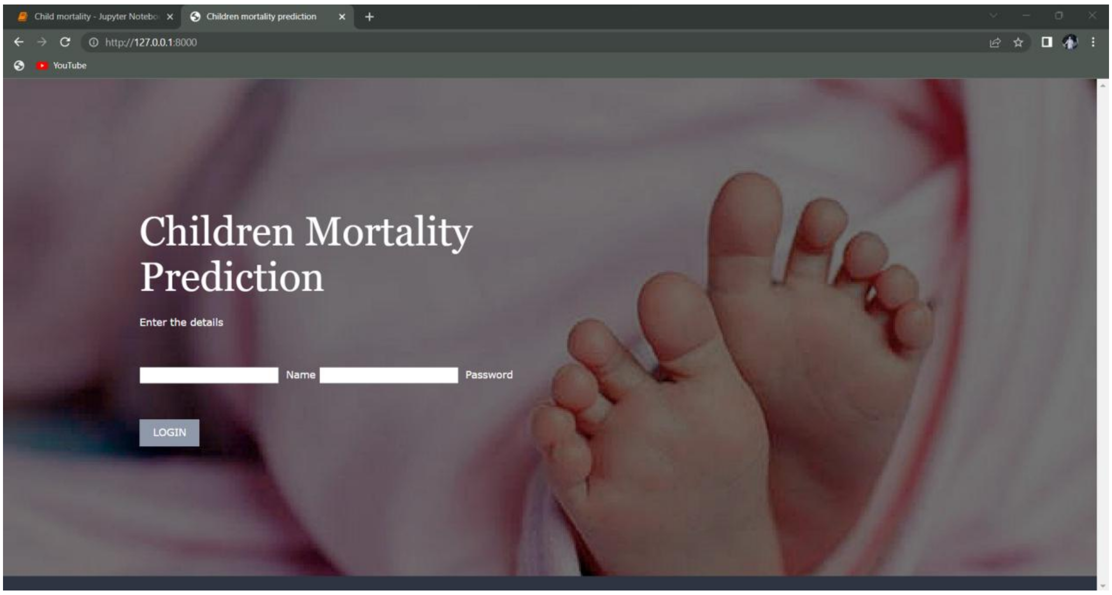
In [34]:
```

# Model Evaluation
print('R^2:', metrics.r2_score(y_test, y_pred2))
print('Adjusted R^2:',1 - (1-metrics.r2_score(y_test, y_pred2))*(len(y_test)-1)/(len(y_test)-X_train.shape[1]-1))
print('MAE:',metrics.mean_absolute_error(y_test, y_pred2))
print('MSE:',metrics.mean_squared_error(y_test, y_pred2))
print('RMSE:',np.sqrt(metrics.mean_squared_error(y_test, y_pred2)))
```

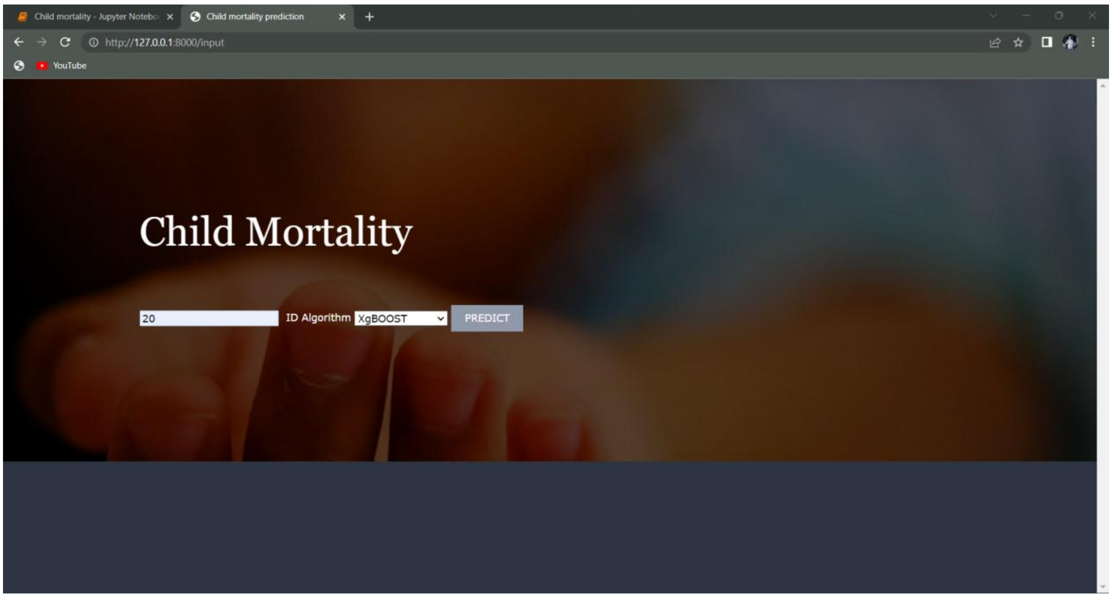
```
R^2: 0.9984652175753922
Adjusted R^2: 0.9984643662427848
MAE: 0.02359065297237996
MSE: 0.0033899074641105436
RMSE: 0.058222911848434235
```

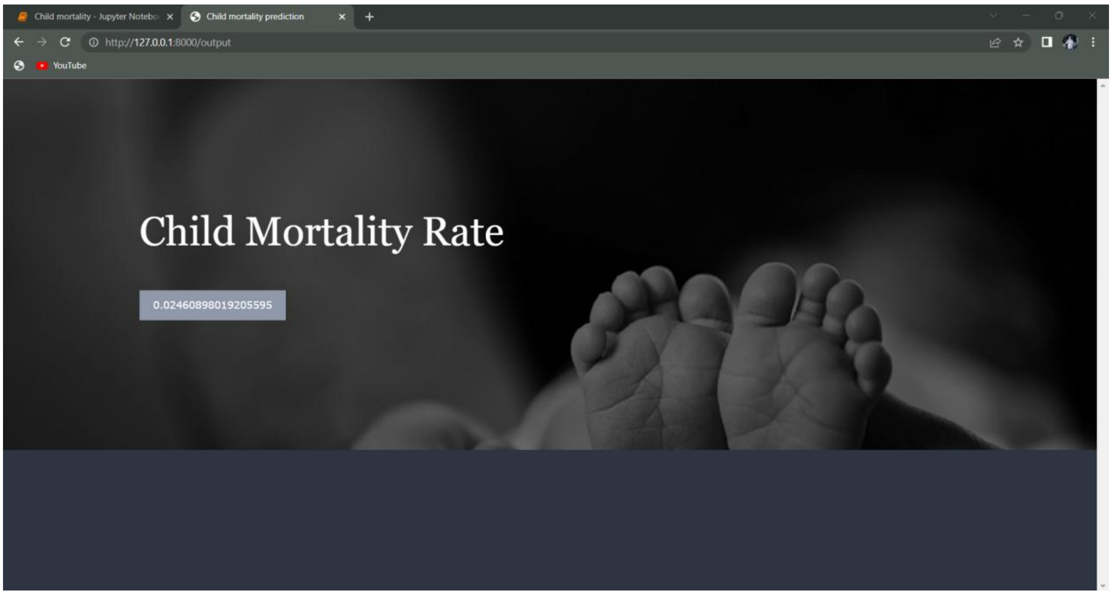
In [35]:
```
xg_compare = pd.DataFrame({'Real Values':y_test, 'Predicted Values': y_pred2})
xg_compare.head(10)
```

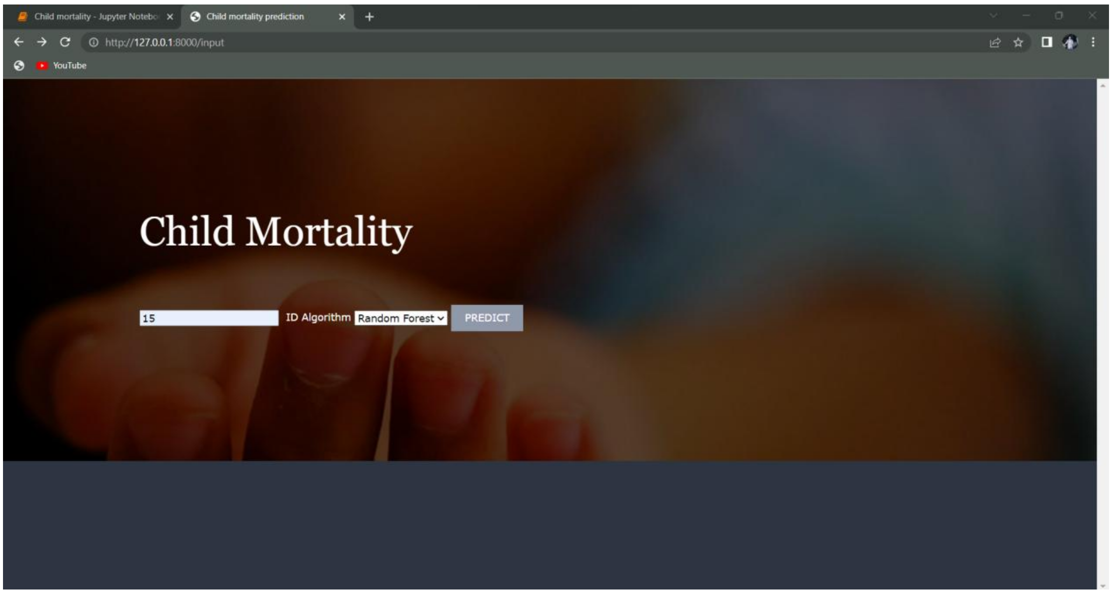
Out[35]:

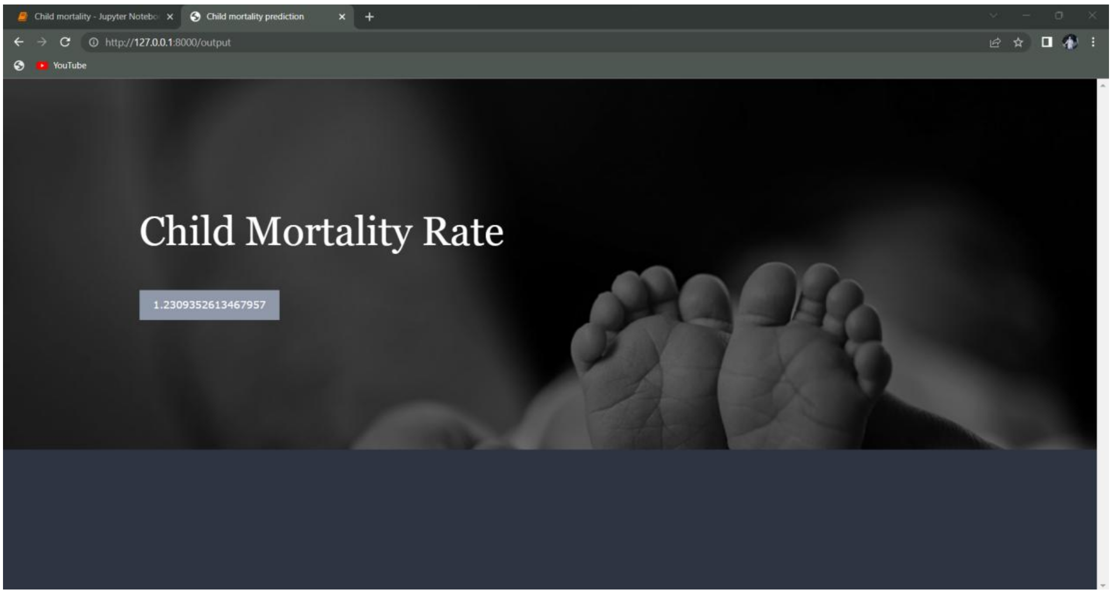
	Real Values	Predicted Values
15188	1.911865	1.927048
17322	0.019336	0.020476
17728	1.053273	1.040518
23539	0.022199	0.022738
339	4.431517	4.464691
14603	1.523055	1.461040
21945	0.152330	0.153861
20935	2.914837	2.711911
18792	3.768008	3.781269
1740	0.452978	0.397881


Child mortality - Jupyter Notebo
Children mortality prediction
http://127.0.0.1:8000
YouTube
Children Mortality Prediction
Enter the details
Name
Password
LOGIN


Child mortality - Jupyter Notebo
Child mortality prediction
http://127.0.0.1:8000/input
YouTube
Child Mortality
20
ID Algorithm
XgBOOST
PREDICT


Child mortality - Jupyter Notebo
Child mortality prediction
http://127.0.0.1:8000/output
YouTube
Child Mortality Rate
0.02460898019205595


Child mortality - Jupyter Notebo
Child mortality prediction
http://127.0.0.1:8000/input
YouTube
Child Mortality
15
ID Algorithm
Random Forest
PREDICT



5.3 TEST PLAN & DATA VERIFICATION

S.NO	**TEST CASES**	**TESTING**
01	Data Collection	Success
02	Data Pre-Processing	Success
03	Data Formatting	Success
04	Data Sampling	Success
05	Featurization	Success
06	Splitting	Success
07	Training	Success
08	Testing	Success
09	Algorithm Testing	Success

Chapter 6

Results

6.1 RESEARCH FINDINGS

To get a general understanding of the child mortality prediction, I initially conducted manual searches on Google using the term child mortality prediction. There were numerous cases and research papers on mortalities.
I then restricted my search to child mortality prediction. A few papers came from it. I picked the Random Forest and XGBoost algorithms because they produce good accuracy of 99% and 98% respectively. Despite the fact that there are alternative child mortality predictions that employ different algorithms.

6.2 RESULT ANALYSIS AND EVALUATION MATRICS

S.NO	ALGORITHM	TRAIN ACCURACY	TEST ACCURACY
01	Random Forest	99	98
02	XGBoost	99	98

6.2.1 PERFORMANCE MATRICES

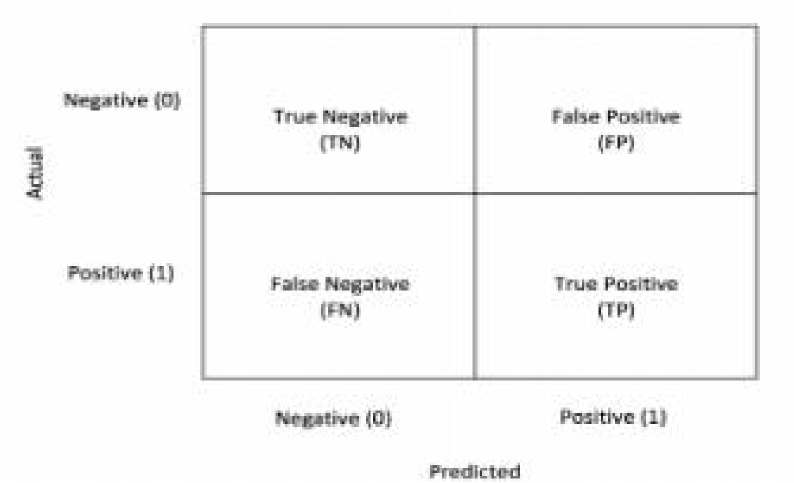
Data was divided into two portions, training data and testing data, both these portions consisting 70% and 30% data respectively. All these two algorithms were applied on same dataset using Enthought Canaopy and results were obtained.

$$\text{Accuracy} = (TP+TN) / (P + N)$$

Predicting accuracy is the main evaluation parameter that we used in this work. Accuracy can be defied using equation. Accuracy is the overall success rate of the algorithm.

6.2.2 CONFUSION MATRICES

It is the most commonly used evaluation metrics in predictive analysis mainly because it is very easy to understand and it can be used to compute other essential metrics such as accuracy, recall, precision, etc. It is an N x N matrix that describes the overall performance of a model when used on some dataset, where N is the number of class labels in the classification problem.



All predicted true positive and true negative divided by all positive and negative. True Positive (TP), True Negative (TN), False Negative (FN) and False Positive (FP) predicted by all algorithms are presented in table.

True positive (TP) indicates that the positive class is predicted as a positive class, and the number of sample positive classes was actually predicted by the model.

False negative indicates (FN) that the positive class is predicted as a negative class, and the number of negative classes in the sample was actually predicted by the model.

False positive (FP) indicates that the negative class is predicted as a positive class, and the number of positive classes of samples was actually predicted by the model.

True negative (TN) indicates that the negative class is predicted as a negative class, and the number of sample negative classes was actually predicted by the model.

Chapter 7

Conclusion & Future Work

In developing a predictive model, ML approaches are strong and can be used to classify certain secret knowledge that could not be detected by conventional statistical methods The analytical method started from information improvement and process, missing worth, wildcat analysis and eventually model building and analysis. The best accuracy on public check set is higher accuracy score is are going to be determine. This application will facilitate to seek out the Prediction of children's Mortality. Based to this, ML techniques can improve the accuracy of the algorithm and use training data for the training model and use unseen test data to make predictions. ML approaches have high output accuracy compared to conventional statistical methods.

Future enhancement for this paper is to add more feature and data points with better sensitivity to correlation of feature and target for example, like number of companies registered and number of companies closed doors with the size of the company as well.

REFERENCES

[1] Reidpath D, Allotey P. Infant mortality rate as an indicator of population health. J Epidemiol Community Heal [Internet]. 2003;57(5):344– 346. https://doi.org/10.1136/jech.57.5.344.

[2] Blencowe H, Krasevec J, de Onis M, Black RE, An X, Stevens GA, et al. National, regional, and worldwide estimates of low birthweight in 2015, with trends from 2000: a systematic analysis. Lancet Glob Heal [Internet]. 2019;7(7):e849–60. https://doi.org/10.1016/S2214-109X(18)30565-5.

[3] Shukla V V., Eggleston B, Ambalavanan N, McClure EM, Mwenechanya M, Chomba E, et al. Predictive Modeling for Perinatal Mortality in Resource Limited Settings. JAMA Netw Open [Internet]. 2020;3(11):1–13. https:// doi.org/10.1001/jamanetworkopen.2020.26750.

[4] Ramakrishnan R, Rao S, He JR. Perinatal health predictors using artifcial intelligence: A review. Women's Heal. 2021;17:17455065211046132.

[5] Panch T, Szolovits P, Atun R. Artifcial intelligence, machine learning and health systems. J Glob Heal. 2018;8(2):1–8. http://dx.doi.org/https://doi. org/10.7189/jogh.08.020303.

[6] Sodhi P, Awasthi N, Sharma V. Introduction to Machine Learning and Its Basic Application in Python. Elsevier [Internet]. 2019;1367–186. https:// doi.org/10.2139/ssrn.332379.

[7] Aslam N, Khan IU, Aljishi RF, Alnamer ZM, Alzawad ZM, Almomen FA, et al. Explainable Computational Intelligence Model for Antepartum Fetal Monitoring to Predict the Risk of IUGR. Electron. 2022;11(4):593. https:// doi.org/10.3390/electronics11040593.

[8] Oprescu AM, Miró-Amarante G, García-Díaz L, Beltrán LM, Rey VE, RomeroTernero M. Artifcial intelligence in pregnancy: A scoping review. IEEE Access [Internet]. 2020;8:181450–84. https://doi.org/10.1109/ACCESS.2020.3028333

[9] Biracyaza E, Habimana S. Modeling the Risks Factors Associated with Infant Mortality in Rwanda from 2011 to 2015: Analysis of Rwanda Demographic and Health Survey (RDHS). J Pediatr Child Heal Care. 2019;4(1):4–7.

[10] Dusabe J. Determinants of under-5 mortality in Rwanda. University of Nairobi Masters in Arts in Economics School of Economics. 2016. http://erepository.uonbi.ac.ke/bitstream/handle/11295/99950/Dusabe%20Josiane_Determinants%20of%20Under-5%20Mortality%20in%20Rwanda.pdf?sequence=1.

[11] Adeyinka Emmanuel Adegbosin, Stantic B, Sun J. Efcacy of deep learning methods for predicting under-fve mortality in 34 low-income and middle-income countries. BMJ Open. 2019; https://doi.org/10.1136/ bmjopen-2019-034524 .

[12] Podda M, Bacciu D, Micheli A, Bellù R, Placidi G, Gagliardi L. A machine learning approach to estimating preterm infants survival: development of the Preterm Infants Survival Assessment (PISA) predictor. Sci Rep. 2018;8:13743. https://doi.org/10.1038/s41598-018-31920-6.

[13] Kabudula C, Kara R, Wandera H, Dake FAA, Dansou J, Debeko DD, et al. Evaluation of machine learning methods for predicting the risk of child mortality in South Africa. 2019; https://www.uaps2019.popconf.org

[14] Podda M, Bacciu D, Micheli A, Bellù R, Placidi G, Gagliardi L. A machine learning approach to estimating preterm infants survival: development of the Preterm Infants Survival Assessment (PISA) predictor. Sci Rep. 2018.