PREDICTION OF MATERNAL HEALTH USING CLASSIFICATION METHOD WITH HARDWARE

A PROJECT REPORT

Submitted by

MANJU S [211419104161]

RAVEENA M [211419104219]

SHANMUGA PRIYAA B [211419104247]

in partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

PANIMALAR ENGINEERING COLLEGE

(An Autonomous Institution, Affiliated to Anna University, Chennai)

APRIL 2023

PANIMALAR ENGINEERING COLLEGE

(An Autonomous Institution, Affiliated to Anna University, Chennai)

BONAFIDE CERTIFICATE

Certified that this project report **"PREDICTION OF MATERNAL HEALTH USING CLASSIFICATION METHOD"** is the bonafide work of **"MANJU S (211419104161) RAVEENA M (211419104219) SHANMUGA PRIYAA B (211419104247) "** who carried out the project under my supervision.

SIGNATURE	**SIGNATURE**
Dr.L.JABASHEELA,M.E.,Ph.D., **HEAD OF THE DEPARTMENT**	**Dr.K.VALARMATHI,M.E.,Ph.D.,** **SUPERVISOR** **PROFESSOR**
DEPARTMENT OF CSE, PANIMALAR ENGINEERING COLLEGE, NASARATHPETTAI, POONAMALLEE, CHENNAI-600 123.	DEPARTMENT OF CSE, PANIMALAR ENGINEERING COLLEGE, NASARATHPETTAI, POONAMALLEE, CHENNAI-600 123.

Certified that the above candidates were examined in the End Semester Project Viva-Voce Examination held on 11.04.2023.

INTERNAL EXAMINER **EXTERNAL EXAMINER**

DECLARATION

We MANJU S (211419104161), RAVEENA M (211419104219), SHANMUGA PRIYAA B (211419104247) hereby declare that this project report titled "**PREDICTION OF MATERNAL HEALTH USING CLASSIFICATION METHOD WITH HARDWARE** " , under the guidance of **Dr.K VALARMATHI**, M.E Ph.D. is the original work done by us and we have not plagiarized or submitted to any other degree in any university by us.

MANJU S

RAVEENA M

SHANMUGA PRIYAA B

ACKNOWLEDGEMENT

We would like to express our deep gratitude to our respected Secretary and Correspondent **Dr.P.CHINNADURAI, M.A., Ph.D**. for his kind words and enthusiastic motivation, which inspired us a lot in completing this project.

We express our sincere thanks to our beloved Directors **Tmt. C.VIJAYARAJESWARI, Dr.C.SAKTHIKUMAR,M.E.,Ph.D** and **Dr.SARANYASREE SAKTHIKUMAR B.E.,M.B.A.,Ph.D.,** for providing us with the necessary facilities to undertake this project.

We also express our gratitude to our Principal **Dr.K.MANI, M.E., Ph.D.** who facilitated us in completing the project.

We thank the Head of the CSE Department, **Dr.L.JABASHEELA, M.E.,Ph.D.,** for the support extended throughout the project.

We would like to thank my Project Guide, **Dr.K.VALARMATHI, M.E.,Ph.D.,** and all the faculty members of the Department of CSE for their advice and encouragement for the successful completion of the project.

MANJU S

RAVEENA M

SHANMUGA PRIYAA B

ABSTRACT

The maternal health data has been collected and different machine learning algorithms are used for the prediction of maternal health. The Trained Machine learning model takes the parameters regarding the maternal women as input and then predicts the risk level of maternal health.The Healthcare industry produces a gigantic amount of data related to child immunization, maternal health, family planning, clinical data, health surveys, diagnosis, etc. As the process of data collection in the health sector increases, the usage of data mining and machine learning techniques for analysing and decision-making also increases. There is one major health issue in the health sector that is maternal health that needs to be worried about. In this research paper, maternal health data has been collected and different machine learning algorithms are used for the prediction of maternal health. Various performance measures including Accuracy, Precision, Recall, and F-measure, have also been used for calculating the performance.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATION

ABBREVIATION	EXPANSION
ML	Machine Learning
CNN	Convolutional Neural Network
ER	Entity Relationship
MLP	Multi-Layer Perceptron
UML	Unified Modeling Language
FP	False Positives
FN	False Negatives
TP	True Positives
TN	True Negatives

CHAPTER 1

INTRODUCTION

1.1 OVERVIEW

Ladies are the mainstays of any family and society however maternal passing during pregnancy and labour is an extraordinary misfortune to a child, family, society, and country. Most maternal passouts happen because of preventable causes including draining after birth, dangerous early terminations, blocked work, hypertension, discharge, and hypertensive problems, cracked uterus, hepatitis, weakness, and so on. Maternal mortality alludes to ladies' passing per 100,000 live births because of childbearing or in something like 42 days of the end of pregnancy and labour. One of the significant medical problems that should be stressed universally. Worldwide 6.4 million ladies get pregnant consistently and on a regular schedule, and roughly 800 ladies bite the dust from causes connected with pregnancy and labour. The non-industrial nations like India have the most elevated maternal death rate with 25% of the world's maternal passouts accounted in 2015 as it were. As per the most recent report of the World Wellbeing Association (WHO), India's maternal death rate (MMR) in 2016 was 130 passings for every 100,000 live births which comes after the death pace of Bhutan with 148 passings yet the death pace of India got declined from the figure of 215 passings that was accounted for in the year 2010.

1.2 PROBLEM DEFINITION

Around the world, in excess of 200 million ladies become pregnant every year. Although most pregnancies end with a live child to a solid mother, at times, the occasion is a period of agony, enduring, and even demise. Truth be told, an expected 585,000 ladies kick the bucket every year, and another 20 million ladies foster persistent, incapacitating diseases because of pregnancy-related difficulties. Of these maternal passing's, an expected almost 100 occur in emerging nations.All through the world, varying conduct and natural elements increase the endangerment of a lady creating perilous entanglements. In many societies, labor is encircled by customs, a significant number of which are useful; others might be destructive to the lady and kid. Maternal mortality is impacted by many interrelated factors including economic well-being and position of ladies, monetary assets, what's more, the foundation of the nation, and the openness and accessibility of abilities, materials, and offices for suitable family arranging and obstetric consideration.

CHAPTER 2

LITERATURE SURVEY

2.1 LITERATURE SURVEY

The paper [1] "Maternal Risk Level Prediction Using Ensemble Model" work has created a system for accurately monitoring and forecasting a pregnant woman's risk level. Pregnant women's health information and risk factors will be examined by this method to determine the risk intensity level. By 2030, the United Nations wants to lower mother and infant deaths and improve maternal health, but the rate is not declining as quickly as it should. This study evaluated the risk level based on risk factors in pregnancy using the relevant analytical tools and machine learning algorithms.

The paper [2] "Prediction Model for Mortality Analysis of Pregnant Women Affected With COVID-19" aims to develop a predictive model to estimate the possibility of death for a COVID-diagnosed mother based on documented symptoms: dyspnea, cough, rhinorrhea, arthralgia, and the diagnosis of pneumonia. The machine learning models that have been used in our study are support vector machines, decision trees, random forest, gradient boosting, and artificial neural network. The models have provided impressive results and can accurately predict the mortality of pregnant mothers with a given input.

In the paper [9] "Machine learning for maternal health: Predicting delivery location in a community health worker program in Zanzibar" We use program data from 38,787 women enrolled in Safer Deliveries, a community health worker program in Zanzibar, to build a generalizable prediction model that accurately predicts whether a newly enrolled pregnant woman will deliver in a health facility Our models correctly predicted the delivery location for 68%–77% of the women in the test set, with slightly higher accuracy when predicting facility delivery versus home delivery.

The paper [12] “Wearable electrocardiogram signal monitoring and analysis based on convolutional neural network,” proposes a transfer learning approach for Arrhythmia Detection and Classification in Cross ECG Databases. This approach relies on a deep convolutional neural network (CNN) pretrained on an auxiliary domain (called imageNet) with very large labeled images coupled with an additional network composed of fully connected layers.

The paper [3] "Analysis and Prediction of Gestational Diabetes Mellitus by the Ensemble Learning Method" is to use a new integrated LightGBM-Xgboost-GB ensemble learning method for maternal and child health care providers to analyze the risk factors for GDM. This method uses a group of the most important influencing factors selected through learning algorithms to identify GDM risk factors accurately and to predict the possibility of gestational diabetes precisely. In response to the research problem, the proposed model improves the existing methodology. In the paper [4] "A Semi-Supervised Machine Learning Approach in Predicting High-Risk Pregnancies in the Philippines" the study first compared multiple supervised machine learning algorithms to analyze and accurately predict high-risk pregnancies. Through hyperparameter tuning, supervised learning algorithms such as Decision Tree, Random Forest, Support Vector Machine, K-Nearest Neighbors, Naïve Bayes, and Multilayer Perceptron were evaluated by using 10-fold cross validation to obtain the best parameters with the best scores.

The paper [5] "An Efficient Small for Gestational Age Prognosis System Using Stacked Generalization Scheme (SGS)" proposes to use two steps for the classification of an SGA. In the first step, a cleaned feature vector is established to remove noisiness from the SGA dataset using the proposed Cleaned Feature Vector In the second step, SGS is used for the classification task. Stacking enhances the SGA classification performance with minimized generalization errors. While using stacking, the classifiers are constructed and tested using ten-fold cross-validation to diagnose an infant as an SGA or non-SGA.

In the paper [6] "Ensemble learning-based feature engineering to analyze maternal health during pregnancy and health risk prediction" the proposed approach combines DT and BiLTCN for feature engineering on two rationales. First, DT and BiLTCN individually perform well for the task at hand. Owing to their performance, these models are combined in the proposed approach. In addition, we experimented with other combinations, however, the performance of those combinations was inferior.

In the paper [11] "IoT-Based Smart Health System for Ambulatory Maternal and Fetal Monitoring" the data generated by IoT devices are transferred to the emergency subsystem, which analyzes and determines if there is any significant maternal or fetal distress. If any emergency situation is detected, the medical staff is promptly informed. After this processing phase, data is sent to the cloud solution,where all features are calculated and submitted to the proposed prediction subsystem based on a 1-D CNN. The paper [15] "Maternal health and health-related behaviors and their associations with child health: Evidence from an Australian birth cohort" is a study using Wave 1 (2004) and Wave 7 (2016) data from the Longitudinal Survey of Australian Children (LSAC). We measured mothers' general health, presence of a medical condition during pregnancy and mental health during pregnancy or in the year after childbirth.Our results showed that poor general health of the mother in the year after childbirth was associated with higher odds of poor health in infants and adolescents in all three dimensions: poor general health The paper [6] "Ensemble learning-based feature engineering to analyze maternal health during pregnancy and health risk prediction" aims to develop an artificial neural network-based system for predicting maternal health risks using health data records. In the paper [10] "Machine learning to predict pregnancy outcomes: a systematic review, synthesizing framework and future research agenda " The primary aim of this review study is to explore current research and development perspectives that utilizes the ML techniques to predict the optimal mode of childbirth and to detect various complications during childbirth. A total of 26 articles (published between 2000 and 2020) from an initial set of 241 articles were selected and reviewed following a Systematic

Literature Review (SLR) approach.

In this paper [7] "Predicting perinatal mortality based on maternal health status and health insurance service using homogeneous ensemble machine learning methods" Several experiments were conducted using different feature selection methods such as filter (Mutual information and Chi-square), wrapper (step forward and step backward) and domain experts. And 17 features that were selected by step backward feature selection methods and features that were recommended by domain experts were used to develop the classification model. The paper [14] "Perinatal health predictors using artificial intelligence: A review" Machine learning, a commonly used artificial intelligence method, has been used to predict preterm birth, birthweight, preeclampsia, mortality, hypertensive disorders, and postpartum depression. Real-time electronic health recording and predictive modeling using artificial intelligence have found early success in fetal monitoring and monitoring of women with gestational diabetes especially in low-resource settings. Artificial intelligence-based methodologies have the potential to improve prenatal diagnosis of birth defects and outcomes in assisted reproductive technology too. The paper [13] "Predictive behavior of maternal health inputs and child mortality in West Bengal – An analysis based on NFHS-3" makes use of the 2005-06 National Family Health Survey (NFHS-3), 2005, the third in the series of these national surveys. It covers all the states of India and the respondents were men of age 15–54 years and women of 15–49 years old.

The paper [8] "A Catalog of Machine Learning Algorithms for Healthcare Risk Predictions" proposes a data analysis mechanism experimented in diverse healthcare scenarios, towards constructing a catalog of the most efficient ML algorithms to be used depending on the healthcare scenario's requirements and datasets, for efficiently predicting the onset of a disease.

2.2 COMPARISON TABLE BASED ON LITERATURE SURVEY

YEAR	TITLE	AUTHOR	DESCRIPTION	ADVANTAGES	DISADVANTAGES
2023	"Maternal Risk Level Prediction Using Ensemble Model".	Nirmala , Rekha S Kambli	The work has created a system for accurately monitoring and forecasting a pregnant woman's risk level. Pregnant women's health information and risk factors will be examined by this method to determine the risk intensity level.	To identify significant risk components for evaluating, categorizing, and predicting risk intensity, a different strategy has been taken. The dataset's maximum accuracy is provided by the ensemble random forest model.	It doesn't make use of hybrid models for prediction in which the accuracy of the result may be more accurate.
2022	"Prediction Model for Mortality Analysis of Pregnant Women Affected With COVID-19"	Quazi Adibur Rahman Adib Sidratul Tanzila Tasmi,Shahriar Islam Bhuiyan, Mohsin Sarker Raihan,etal	The machine learning models that have been used in our study are support vector machines, decision trees, random forest, gradient boosting, and artificial neural network. The model has provided impressive results and can accurately predict the mortality of pregnant mothers with a given input.	The model can be utilized by health workers globally to list down emergency patients, which can ultimately reduce the death rate of COVID-19 diagnosed pregnant mothers.	The health condition will be uncertain for each and every individual based on the age and examination time and region of the patients in these data were unknown

YEAR	TITLE	AUTHOR	DESCRIPTION	ADVANTAGES	DISADVANTAGES
2022	"Analysis and Prediction of Gestational Diabetes Mellitus by the Ensemble Learning Method".	Xiaojia Wang,Yurong Wang, Shanshan Zhang, Lushi Yao, Sheng Xu	The prediction model studies divide the population into low-risk, medium-risk and high-risk groups according to the probability of disease to provide a basis and direction for personalized diagnosis and treatment and comprehensive intervention in clinical practice	The new integrated model has a good effect in predicting gestational diabetes mellitus. It is possible to improve the accuracy of diagnosis of the disease to improve maternal and infant outcomes as early as possible.	The examination time and region of the patients in these data were unknown, and the analysis results may be different by time and region. It is difficult to explain the inherent complexity of the variable interactions.
2022	"Predicting perinatal mortality based on maternal health status and health insurance service using homogeneous ensemble machine learning methods”	DawitS. Bogale, Tesfamariam. AbuhayBelaynh E. Dejene	The data were pre-processed to get quality data that are suitable for the homogenous ensemble machine-learning algorithms to develop a model that predicts perinatal mortality. We have applied and wrapped feature selection methods. After selecting all the relevant features, we developed a predictive model using cat boost, random forest, and gradient boosting algorithms and evaluated the model.	Risk factors of perinatal mortality were identified using feature importance analysis and relevant rules were extracted using the best performing model.	The limitation of this study is we were not able to measure the effect of the model in the real-world environment.

YEAR	TITLE	AUTHOR	DESCRIPTION	ADVANTAGES	DISADVANTAGES
2022	"A Catalogue of Machine Learning Algorithms for Healthcare Risk Predictions"	Argyro Mavrogiorgou,Athanasios Kiourtis, Spyridon Kleftakis, Konstantinos Mavrogiorgos, Nikolaos Zafeiropoulos and DimosthenisKyriazis	The proposed mechanism utilizes seven widely used and well-established ML algorithms, namely the BNB, KNN, DT, RF, LR, NN, and SGD, to train models to perform predictions across diverse healthcare anomalies' scenarios	This study is useful in assisting the selection of classification algorithms for future applications that exploit relevant health-related data.	Failed to test the current ML algorithms in distributed environments and datasets by utilizing state-of-the-art techniques such as Federated Learning (FL).
2021	"Predictive behaviour of maternal health inputs and child mortality in West Bengal – An analysis based on NFHS-3"	SaswatiChaudhuri, Biswajit Mandal	The joint estimation technique which is a Full Information Likelihood Method (FILM). This takes care of the problem of unobserved heterogeneity which is likely to be present in this kind of research.We have estimated three binary probit equations with the three outcome variables viz. PC, HD, and CM.	This study would help policymakers to identify areas to stress upon in order to pave the way for the formation of good quality human capital in the long run. Another distinguishing feature of this paper is the use of joint estimation techniques that solves unobserved heterogeneity problems.	The analysis becomes less relevant if we fail to identify the factors responsible for the mothers and the households to be exposed to the outcome of child mortality. We find that hospital delivery translates to lower child mortality.

TABLE 2.2 Comparison Table Based on Literature Survey

CHAPTER 3

SYSTEM ANALYSIS

3.1 EXISTING SYSTEM

Assessing the health of both the fetus and mother is vital in preventing and identifying possible complications in pregnancy. This article focuses on a device that can be used effectively by the mother herself with minimal supervision and provide a reasonable estimation of fetal and maternal health while being safe, comfortable, and easy to use. The device proposed uses a belt with a single accelerometer over the mother's uterus to record the required information. The device is expected to monitor both the mother and the fetus constantly over a long period and provide medical professionals with useful information. We have proposed the device to be used to monitor the respiratory patterns of the mother and fetal movement. It can be noted that by using a peak detecting algorithm on the data, the respiratory rate of the subject can be obtained.

APPROACH

One approach was to use convolutional neural networks (CNNs), where a scalogram was generated by taking the wavelet transform of the signal and was then used to identify foetal movement. The other approach was to use a recurrent neural network (RNN), where gated recurrent unit (GRU) cells were used so that the signal can be analysed over long-term dependencies to detect foetal movement

DRAWBACKS

- Deep learning is implemented.
- Two architectures are compared.
- Uses sensors which is a complex process.

3.2 PROPOSED SYSTEM

The main aim of the proposed method is to predict whether the gestation period of maternal health is risky or not. This can be predicted by a machine learning method. More algorithms will be compared .Reduces the error made by the professional .The main aim of the proposed method is to predict whether the gestation period of maternal health is risky or not. This can be predicted by a machine learning method. The process starts from collecting the data. After collecting the dataset, it is pre-processed for removing the unwanted data from the dataset. Machine learning method is now used and mostly used in all the departments where it reduces the mistake. Many algorithms are used and the best one is used for predicting maternal health.

MACHINE LEARNING

Machine learning (ML) is a field of inquiry devoted to understanding and building methods that 'learn', that is, methods that leverage data to improve performance on some set of tasks. It is seen as a part of artificial intelligence. Machine learning algorithms build a model based on sample data, known as training data, in order to make predictions or decisions without being explicitly programmed to do so.

APPROACH

The prediction is done by comparing the results of four algorithms namely Multi-Layer Perceptron, Logistic Regression , Random Forest ,Voting Classifier and the model with higher accuracy is found. Since the results of four algorithms are compared, the accuracy of the result is higher.

ADVANTAGES

- More algorithms will be compared.
- It is not a complex process.
- Reduces the errors made by the professionals.

3.3 FEASIBILITY STUDY

DATA WRANGLING

In this section of the report will load in the data, check for cleanliness, and then trim and clean the given dataset for analysis. Make sure that the document steps carefully and justify cleaning decisions.

DATA COLLECTION

The data set collected for predicting given data is split into Training set and Test set. Generally, 7:3 ratios are applied to split the Training set and Test set. The Data Model which was created using Machine Learning Classification Techniques are applied on the Training set and based on the test result accuracy, Test set prediction is done.

PREPROCESSING

The data which was collected might contain missing values that may lead to inconsistency. To gain better results data need to be preprocessed so as to improve the efficiency of the algorithm. The outliers have to be removed and also variable conversion needs to be done.

BUILDING THE CLASSIFICATION MODEL

The prediction of maternal health, a high accuracy prediction model is effective because of the following reasons: It provides better results in Classification problems.

- It is strong in preprocessing outliers, irrelevant variables, and a mix of continuous, categorical and the continuous variables.
- It produces out-of-bag estimate problems which has proven to be unbiased in many tests and it is relatively easy to tune with.

CONSTRUCTION OF PREDICTIVE MODEL

Machine learning needs data gathering and has a lot of past data. Data gathering has sufficient historical data and raw data. Before data pre-processing, raw data can’t be used directly. It’s used to pre-process then, what kind of algorithm with model. Training and testing this model working and predicting correctly with minimum errors. Tuned model involved by tuned time to time with improving the accuracy.

3.4 HARDWARE REQUIREMENTS

- Processor : Intel i3
- Hard disk : minimum 10 GB
- RAM : minimum 4 GB

3.5 SOFTWARE REQUIREMENTS

- Operating System : Windows 10 or later
- Tool : Anaconda with Jupyter Notebook

CHAPTER 4

SYSTEM DESIGN

4.1 ER DIAGRAM

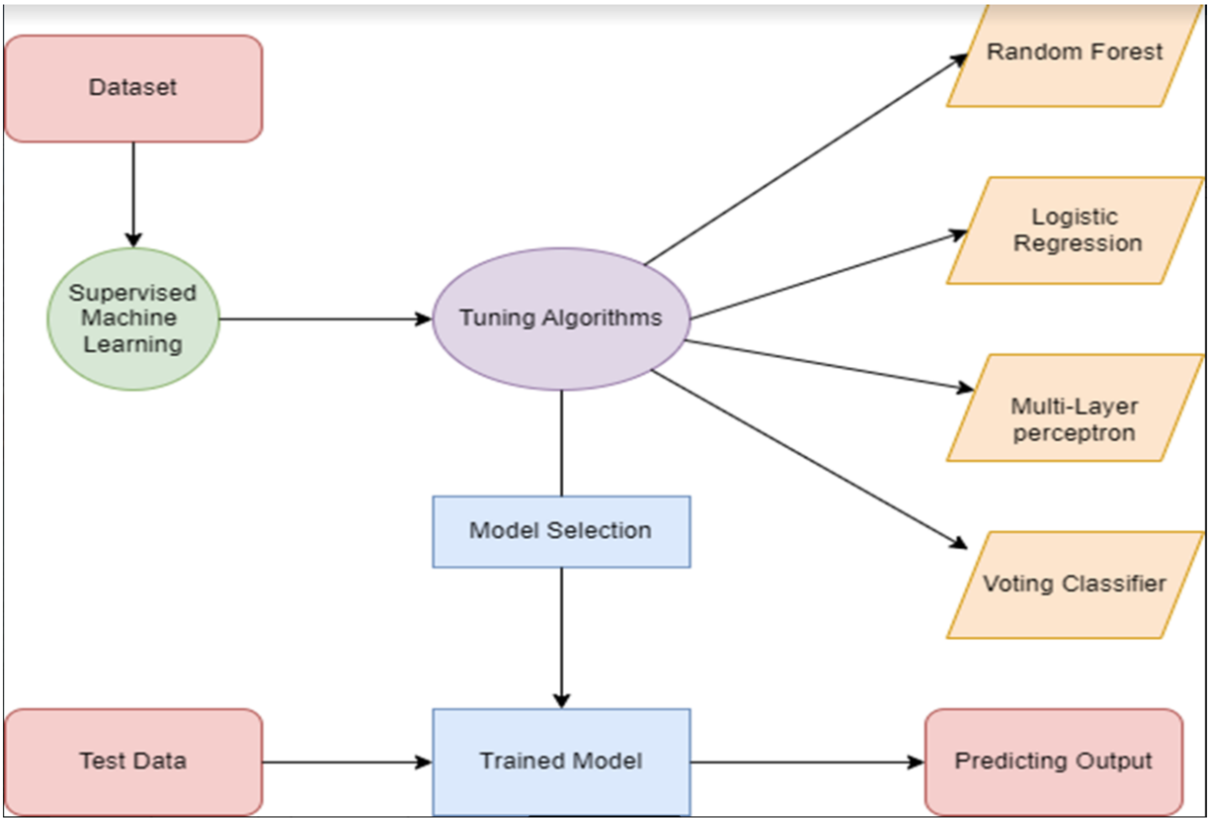


Fig 4.1.1 Entity Relationship Diagram For Maternal Health Prediction System

An entity relationship diagram (ERD), also known as an entity relationship model, is a graphical representation of an information system that depicts the relationships among people, objects, places, concepts or events within that system. An ERD is a data modeling technique that can help define business processes and be used as the foundation for a relational database. Entity relationship diagrams provide a visual starting point for database design that can also be used to help determine information system requirements throughout an organization. After a relational database is rolled out, an ERD can still serve as a referral point, should any debugging or business process re-engineering be needed later.

4.2 TABLE NORMALIZATION

CLASS	Precision	Recall	F1-score	Support
HIGH RISK	0.97	0.90	0.94	82
LOW RISK	0.85	0.82	0.83	122
MID RISK	0.75	0.82	0.78	101
ACCURACY	-	-	0.84	305
MACRO AVERAGE	0.86	0.85	0.85	305
WEIGHTED AVERAGE	0.85	0.84	0.84	305

Table 4.2 Evaluation of the model-Random Forest

The model of Random Forest achieved a training accuracy of 92.38% and the testing accuracy of 84.26%.The Precision scores of High risk ,Low risk and Mid risk respectively are 97%,90% and 94%. The Recall score of High risk,Low risk and Mid risk respectively are 90%, 82% and 82%. The F1 score of High risk,Low risk and Mid risk respectively are 94% ,83% and 78%. The Support of High risk,Low risk and Mid risk respectively are 82, 122 and 101.

These results indicate that the classification model performed well in predicting the Maternal health. The high precision and recall scores suggest that the model had a low number of false positives and false negatives Overall, the results demonstrate the effectiveness of using classification models for predicting the Maternal health.

4.3 DATA FLOW DIAGRAM

LEVEL 0

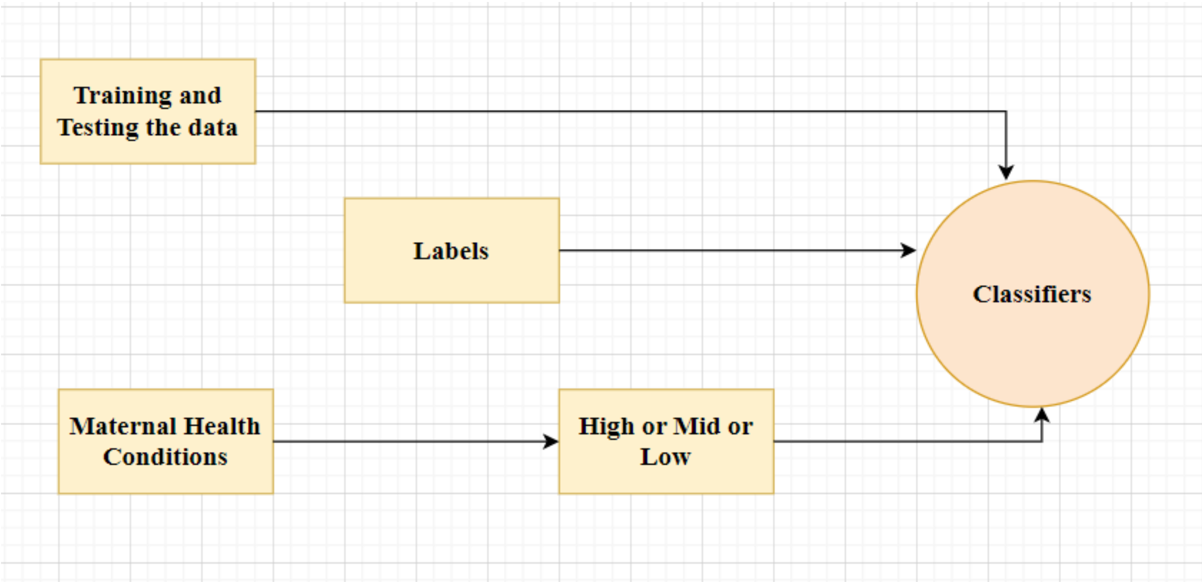


Fig 4.3.1 Level 0 Data Flow Diagram For Maternal Health Prediction System

LEVEL 1

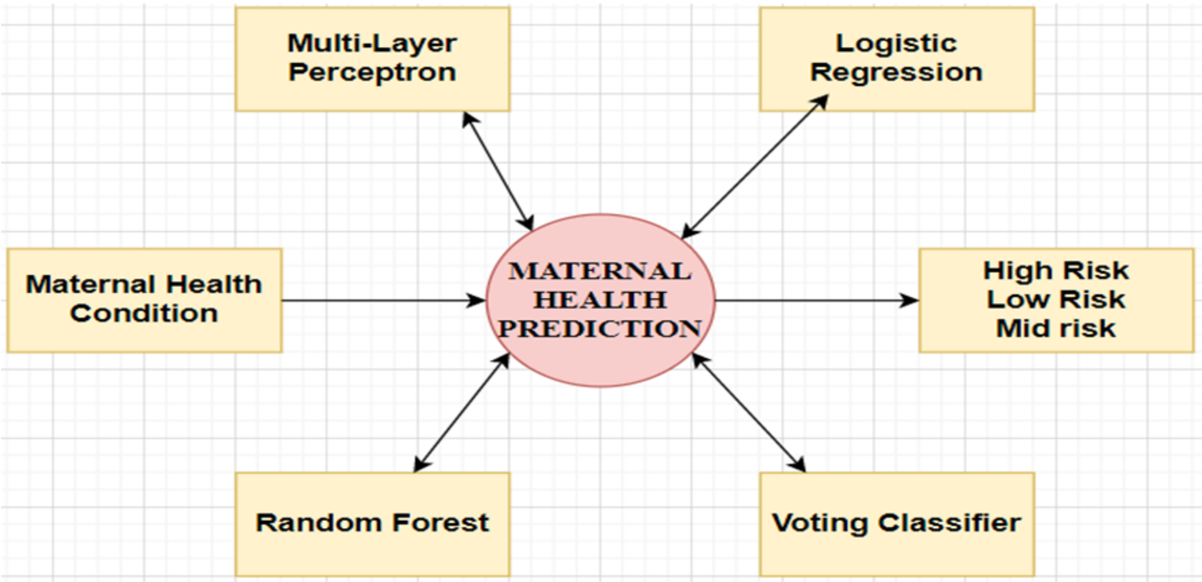


Fig 4.3.2 Level 1 Data Flow Diagram For Maternal Health Prediction System

LEVEL 2

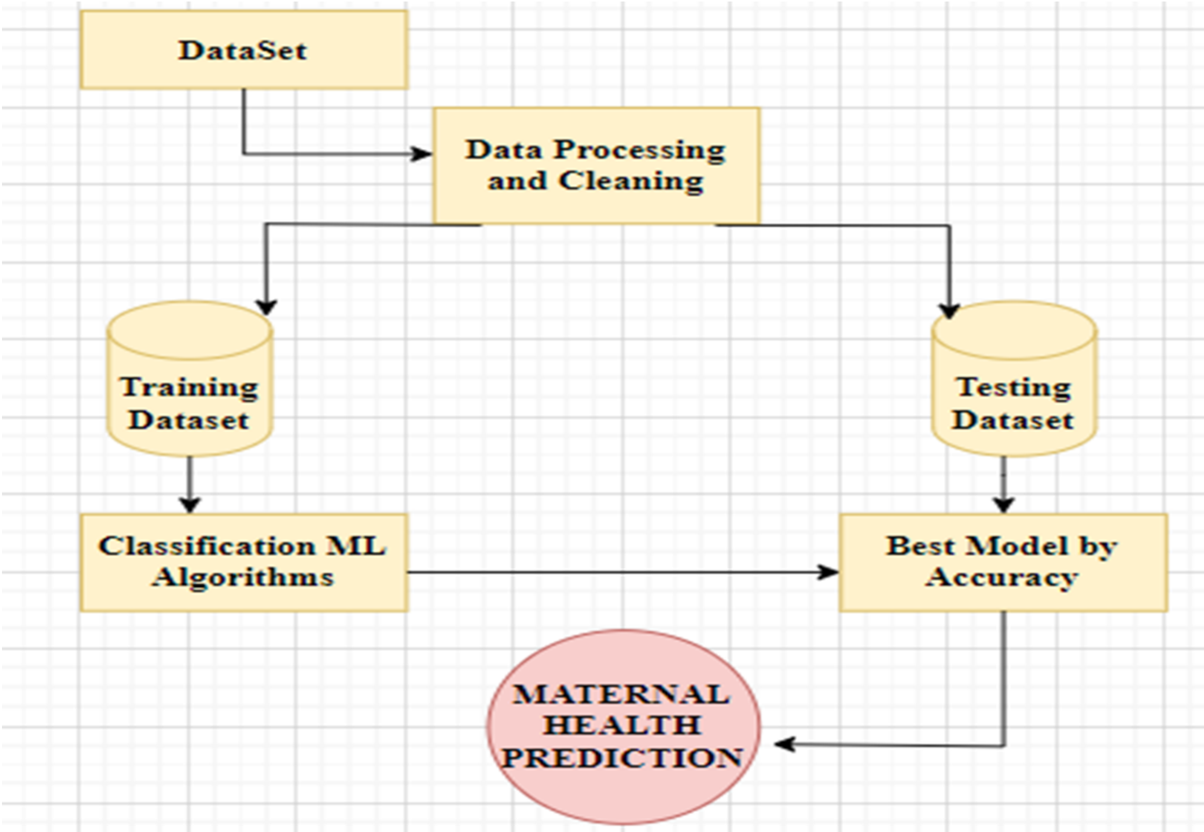


Fig 4.3.3 Level 2 Data Flow Diagram For Maternal Health Prediction System

A data flow diagram (DFD) is a graphical or visual representation using a standardized set of symbols and notations to describe a business operations through data movement. They are often elements of a formal methodology such as Structured Systems Analysis and Design Method (SSADM).

A data-flow diagram is a way of representing a flow of data through a process or a system. The DFD also provides information about the outputs and inputs of each entity and the process itself. A data-flow diagram has no control flow — there are no decision rules and no loops

4.4 UML DIAGRAMS

4.4.1 USE CASE DIAGRAM

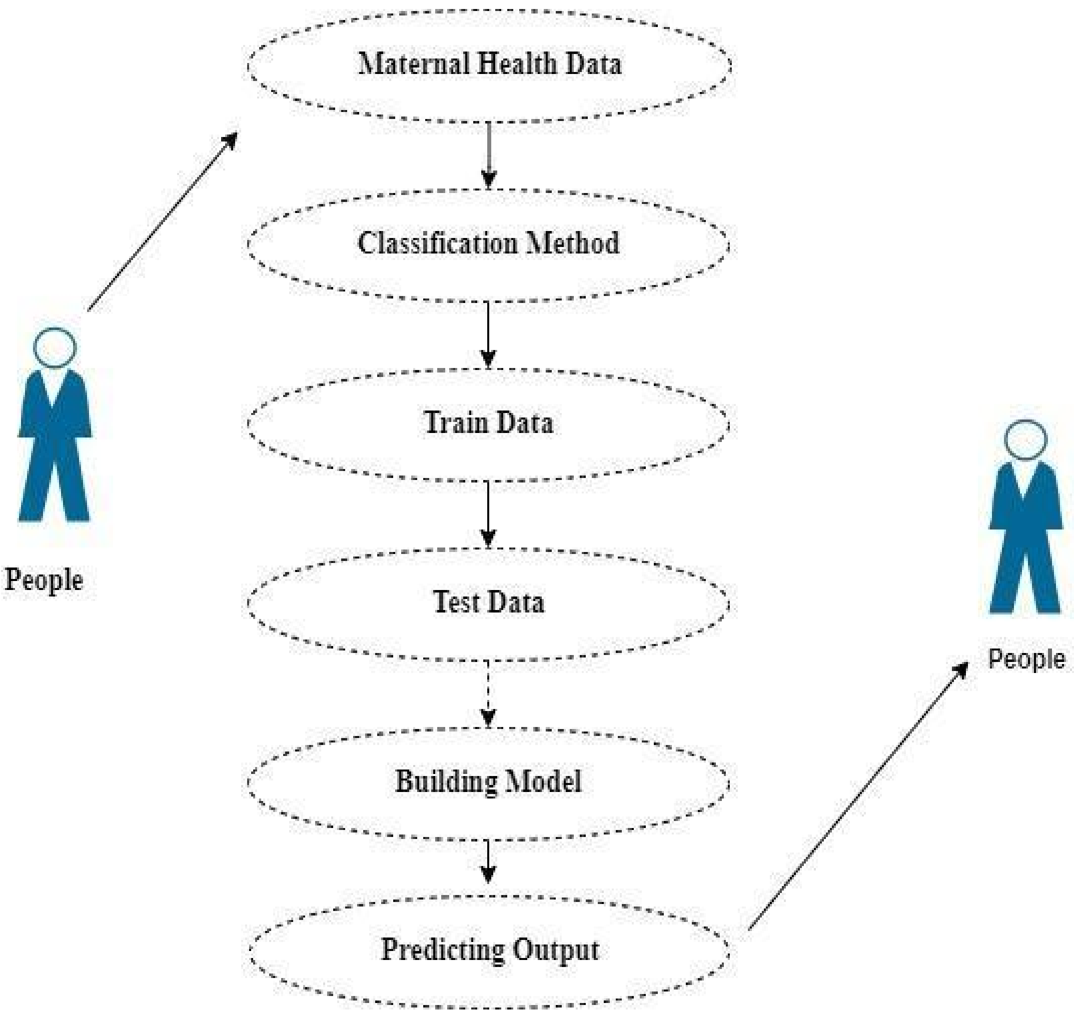


Fig 4.4.1 Use Case Diagram For Maternal Health Prediction System

Use case diagrams are considered for high level requirement analysis of a system. So when the requirements of a system are analysed the functionalities are captured in use cases. So, it can say that use cases are nothing but the system functionalities written in an organized manner.

4.4.2 CLASS DIAGRAM

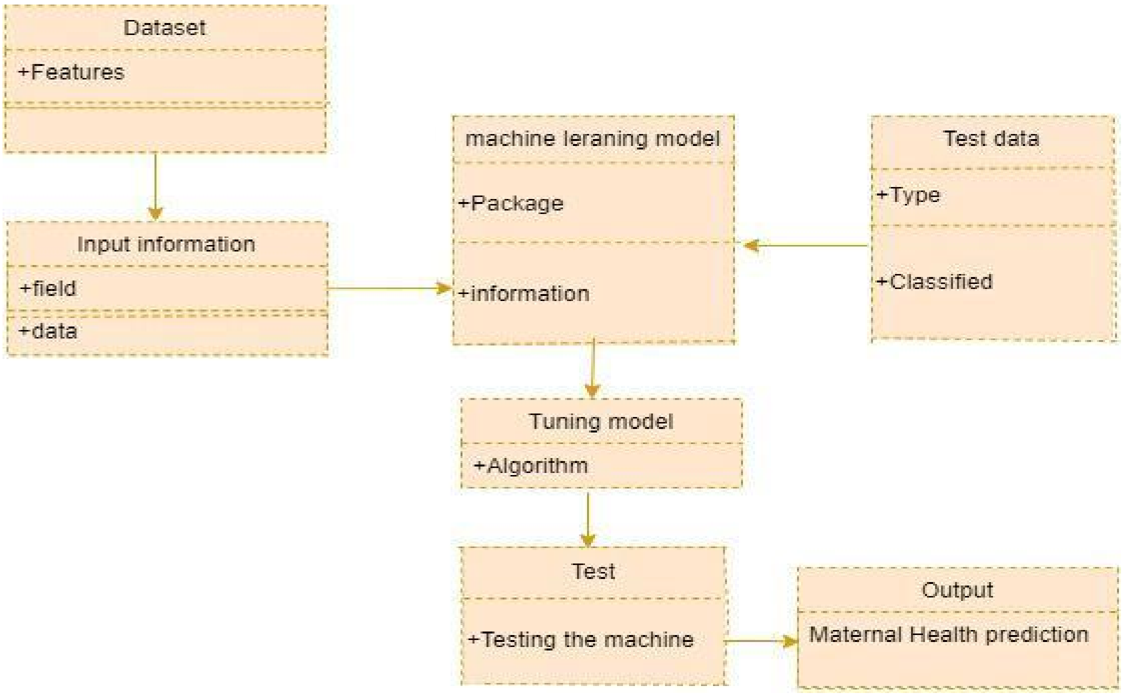


Fig 4.4.2 Class Diagram For Maternal Health Prediction System

Class diagram is basically a graphical representation of the static view of the system and represents different aspects of the application. So a collection of class diagrams represent the whole system. The name of the class diagram should be meaningful to describe the aspect of the system. Each element and their relationships should be identified in advance Responsibility (attributes and methods) of each class should be clearly identified for each class minimum number of properties should be specified and because unnecessary properties will make the diagram complicated. Use notes whenever required to describe some aspect of the diagram and at the end of the drawing it should be understandable to the developer/coder. Finally, before making the final version, the diagram should be drawn on plain paper and reworked as many times as possible to make it correct.

4.4.3 ACTIVITY DIAGRAM

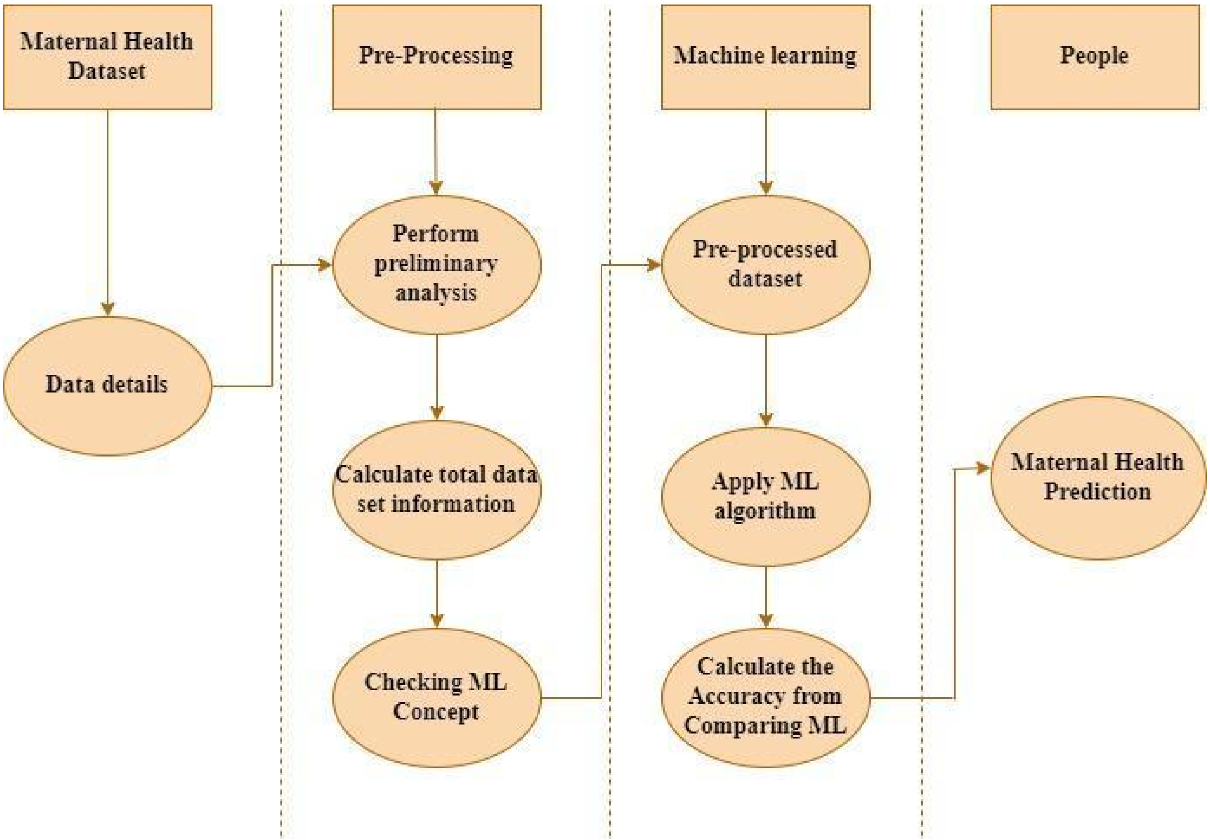


Fig 4.4.3 Activity Diagram For Maternal Health Prediction System

Activity is a particular operation of the system. Activity diagrams are not only used for visualizing the dynamic nature of a system but they are also used to construct the executable system by using forward and reverse engineering techniques. The only missing thing in the activity diagram is the message part. It does not show any message flow from one activity to another. Activity diagram is sometimes considered as the flow chart. Although the diagram looks like a flow chart, it is not. It shows different flows like parallel, branched, concurrent and single.

4.4.4 SEQUENCE DIAGRAM

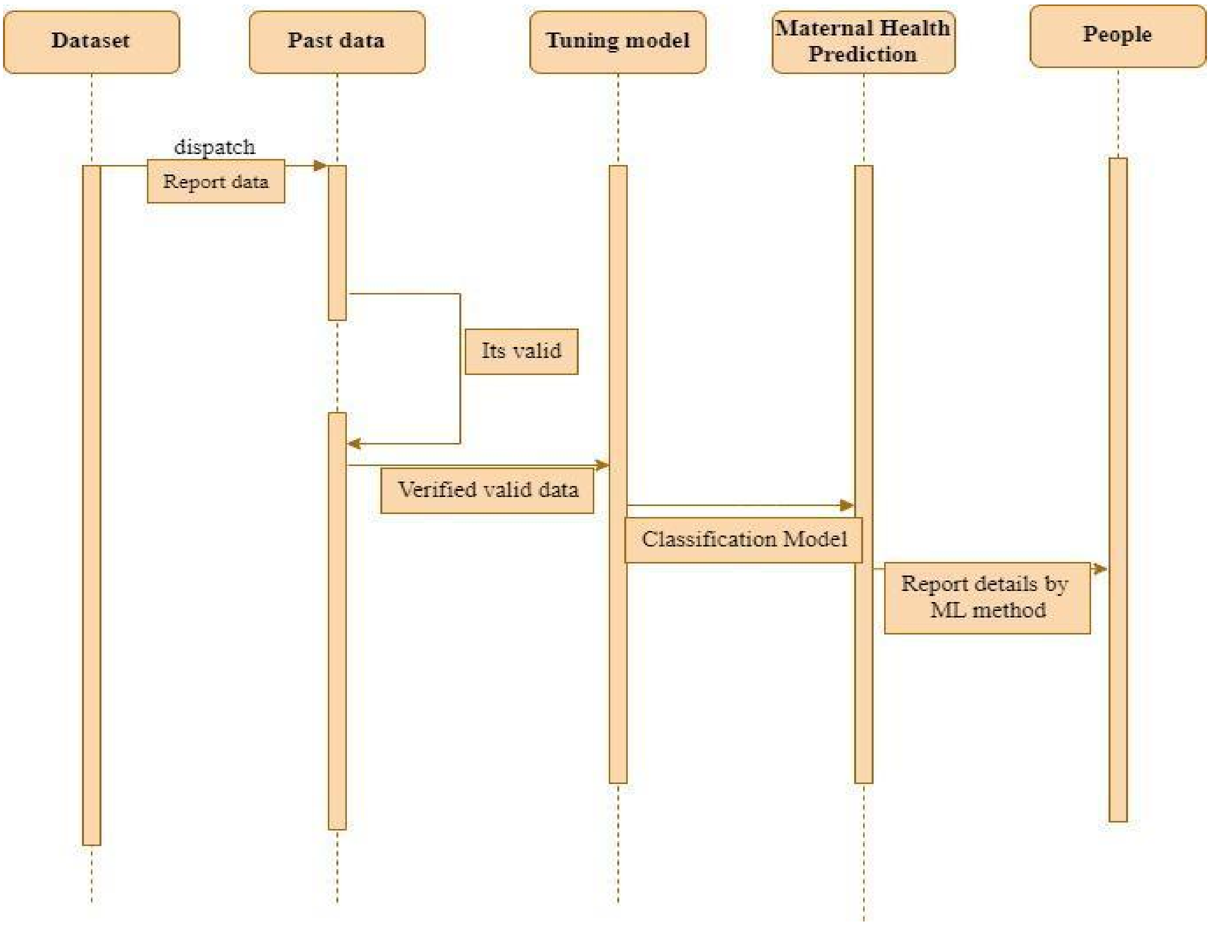


Fig 4.4.4 Sequence Diagram For Maternal Health Prediction System

Sequence diagrams model the flow of logic within your system in a visual manner, enabling you both to document and validate your logic, and are commonly used for both analysis and design purposes. Sequence diagrams are the most popular UML artifact for dynamic modelling, which focuses on identifying the behaviour within your system. Other dynamic modelling techniques include activity diagramming, communication diagramming, timing diagramming, and interaction overview diagramming. Sequence diagrams, along with class diagrams and physical data models are in my opinion the most important design-level models for modern business application development.

CHAPTER 5

SYSTEM ARCHITECTURE

ARCHITECTURE OVERVIEW

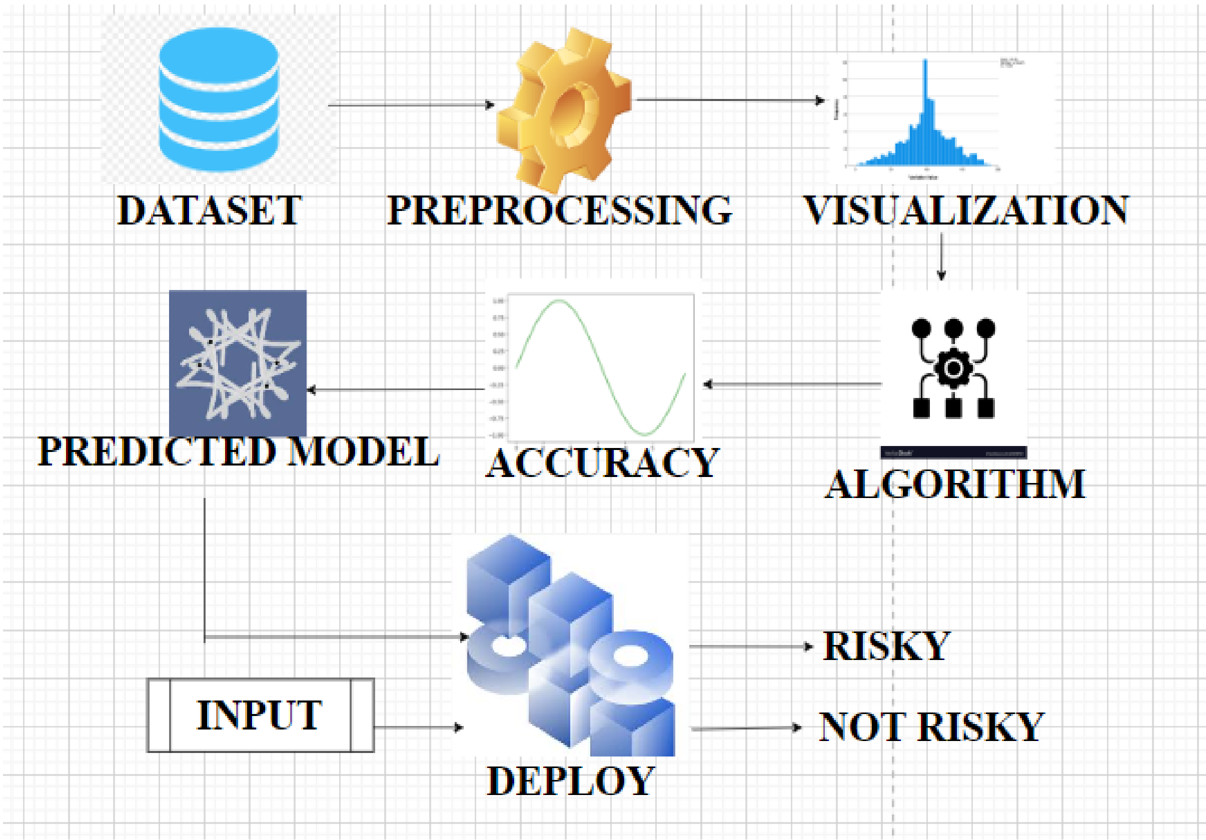


Fig 5.1 Architecture Diagram For Maternal Health Prediction System

First the datasets are collected from the Kaggle. These data are then preprocessed which means removing duplicated data. Then those data are visualized in the pictorial form namely histogram,boxplot,scatter plot,pie chart.Then using four different algorithm model namely Multi-Layer Perceptron,Logistic Regression ,Random Forest,Voting Classifier the risk level is predicted and the model which gives the highest accuracy is taken for the deployment stage.The inputs Age,Blood Sugar,Systolic Value, Diastolic Value, Heart Rate and Body Temperature of the Maternal are given and the model predicts the level of risk.

DEVICE ARCHITECTURE

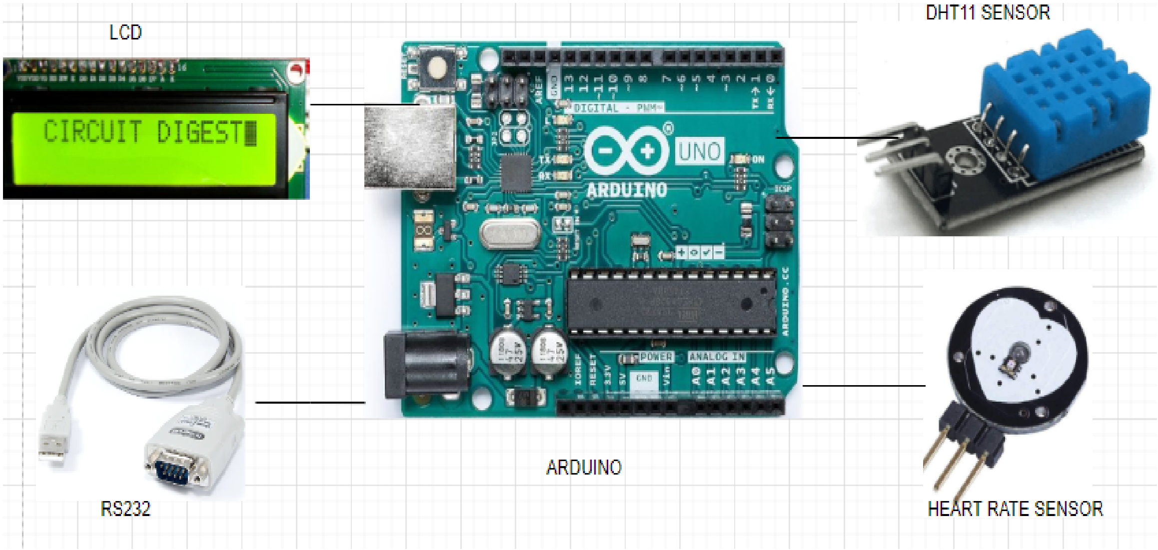


Fig 5.2 Component Configuration of the Maternal Health Prediction Model

The device implemented consists of an Arduino,LCD Display,DHT11 Temperature Sensor,Heart rate Sensor and RS232 Cable.The ATMega 365 is used to store the entire program which is used for prediction.The analog and digital pins are used.The analog pins can work as digital pins. Heart rate sensor measures the heart rate and the temperature sensor measures the body temperature of the pregnant women.RS232 Cable is used to transfer the data to the software.The LCD Display is used to display the values of Temperature and Heart rate of the pregnant women. A pulse sensor, like any other optical heart-rate sensor, works by shining a green light (~ 550 nm) on the finger and measuring the amount of reflected light with a photosensor. This optical pulse detection technique is known as photoplethysmogram. The DHT11 is a basic, ultra low-cost digital temperature and humidity sensor. It uses a capacitive humidity sensor and a thermistor to measure the body temperature and spits out a digital signal on the data pin (no analog input pins needed).

5.1 MODULE DESIGN SPECIFICATION

MODULE – 1

DATA PRE-PROCESSING

Validation techniques in machine learning are used to get the error rate of the Machine Learning (ML) model, which can be considered as close to the true error rate of the dataset. If the data volume is large enough to be representative of the population, you may not need the validation techniques. However, in real-world scenarios, to work with samples of data that may not be a true representative of the population of a given dataset. To find the missing value, duplicate value and description of data type whether it is float variable or integer. The sample of data used to provide an unbiased evaluation of a model fit on the training dataset while tuning model hyper parameters.

The evaluation becomes more biased as skill on the validation dataset is incorporated into the model configuration. The validation set is used to evaluate a given model, but this is for frequent evaluation. Machine learning engineers use this data to fine-tune the model hyper parameters. Data collection, data analysis, and the process of addressing data content, quality, and structure can add up to a time-consuming to-do list. During the process of data identification, it helps to understand your data and its properties; this knowledge will help you choose which algorithm to use to build your model.

A number of different data cleaning tasks using Python's Pandas library and specifically, it focus on probably the biggest data cleaning task, missing values and it able to more quickly clean data. It wants to spend less time cleaning data, and more time exploring and modelling.

Some of these sources are just simple random mistakes. Other times, there can be a deeper reason why data is missing. It's important to understand these different types of missing data from a statistics point of view. The type of missing data will influence how to deal with filling in the missing values and to detect missing values, and do some basic imputation and detailed statistical approach for dealing with missing data. Before joining code, it's important to understand the sources of missing data. Here are some typical reasons why data is missing:

- User forgot to fill in a field.
- There was a programming error.
- Users chose not to fill out a field tied to their beliefs about how the results would be used or interpreted.

Variable identification with Uni-variate, Bi-variate and Multivariate analysis:

- import libraries for access and functional purpose and read the given dataset
- General Properties of analysing the given dataset
- Display the given dataset in the form of data frame
- show columns
- shape of the data frame
- To describe the data frame
- Checking data type and information about dataset
- Checking for duplicate data
- Checking Missing values of data frame
- Checking unique values of data frame
- Checking count values of data frame
- Rename and drop the given data frame
- To specify the type of values
- To create extra columns

DATA VALIDATION/CLEANING/PREPARING PROCESS

Importing the library packages with loading given dataset. To analyse the variable identification by data shape, data type and evaluating the missing values, duplicate values. A validation dataset is a sample of data held back from training your model that is used to give an estimate of model skill while tuning models and procedures that you can use to make the best use of validation and test datasets when evaluating your models. Data cleaning / preparing by renaming the given dataset and dropping the column etc. to analyse the univariate, bi-variate and multivariate process. The steps and techniques for data cleaning will vary from dataset to dataset. The primary goal of data cleaning is to detect and remove errors and anomalies to increase the value of data in analytics and decision making.

	Age	SystolicBP	DiastolicBP	BS	BodyTemp	HeartRate	RiskLevel
0	25	130	80	15.0	98.0	86	high risk
1	35	140	90	13.0	98.0	70	high risk
2	29	90	70	8.0	100.0	80	high risk
3	30	140	85	7.0	98.0	70	high risk
4	35	120	60	6.1	98.0	76	low risk

Fig 5.1.1 Input and Output Classification

MODULE DIAGRAM

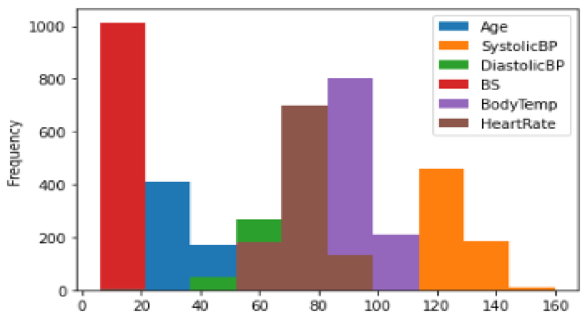
Fig 5.1.2 Module Diagram for Data Processing

MODULE – 2

DATA VISUALIZATION

Data visualization is an important skill in applied statistics and machine learning. Statistics does indeed focus on quantitative descriptions and estimations of data. Data visualization provides an important suite of tools for gaining a qualitative understanding. This can be helpful when exploring and getting to know a dataset and can help with identifying patterns, corrupt data, outliers, and much more. With a little domain knowledge, data visualizations can be used to express and demonstrate key relationships in plots and charts that are more visceral and stakeholders than measures of association or significance. Data visualization and exploratory data analysis are whole fields themselves and it will recommend a deeper dive into some of the books mentioned at the end.

Sometimes data does not make sense until it can be looked at in a visual form, such as with charts and plots. Being able to quickly visualize data samples and others is an important skill both in applied statistics and in applied machine learning. It will discover the many types of plots that you will need to know when visualizing data in Python and how to use them to better understand your own data.




```
]: plt.figure(figsize=(8,4))
   plt.hist(df['BS'])
   plt.title('Average Blood Sugar Levels')
   plt.xlabel('Blood Sugar Level')
   plt.ylabel('Total Patients')
   plt.show()
```

Fig 5.1.3 Data Visualization

5.2 ALGORITHM IMPLEMENTATION

It is important to compare the performance of multiple different machine learning algorithms consistently and to create a test harness to compare multiple different machine learning algorithms in Python with scikit-learn. It can use this test harness as a template on your own machine learning problems and add more and different algorithms to compare. Each model will have different performance characteristics. Using resampling methods like cross validation, you can get an estimate for how accurate each model may be on unseen data. It needs to be able to use these estimates to choose one or two best models from the suite of models that you have created. When having a new dataset, it is a good idea to visualize the data using different techniques in order to look at the data from different perspectives. The same idea applies to model selection. You should use a number of different ways of looking at the estimated accuracy of your machine learning algorithms in order to choose the one or two to finalize. A way to do this is to use different visualization methods to show the average accuracy, variance and other properties of the distribution of model accuracies.In the next section you will discover exactly how you can do that in Python with scikit-learn. The key to a fair comparison of machine learning algorithms is ensuring that each algorithm is evaluated in the same way on the same data and it can achieve this by forcing each algorithm to be evaluated on a consistent test harness.

PERFORMANCE METRICS TO CALCULATE

FALSE POSITIVES (FP): A person who will pay is predicted as a defaulter. When the actual class is no and the predicted class is yes. E.g. if the actual class says this passenger did not survive but the predicted class tells you that this passenger will survive.

FALSE NEGATIVES (FN): A person who default predicted as payer. When the actual class is yes but predicted class is no. E.g. if the actual class value indicates that this passenger survived and the predicted class tells you that passenger will die.

TRUE POSITIVES (TP): A person who will not pay predicted as defaulter. These are the correctly predicted positive values which means that the value of the actual class is yes and the value of predicted class is also yes. E.g. if the actual class value indicates that this passenger survived and the predicted class tells you the same thing.

TRUE NEGATIVES (TN): A person who default predicted as payer. These are the correctly predicted negative values which means that the value of actual class is no and value of predicted class is also no. E.g. if actual class says this passenger did not survive and predicted class tells you the same thing.

True Positive Rate(TPR) = TP / (TP + FN)

False Positive rate(FPR) = FP / (FP + TN)

ACCURACY: The Proportion of the total number of predictions that is correct otherwise overall how often the model predicts correctly defaulters and non-defaulters.

ACCURACY CALCULATION

Accuracy = (TP + TN) / (TP + TN + FP + FN)

Accuracy is the most intuitive performance measure and it is simply a ratio of correctly predicted observation to the total observations. One may think that, if we have high accuracy then our model is best.

PRECISION: The proportion of positive predictions that are actually correct.

Precision = TP / (TP + FP)

RECALL: The proportion of positive observed values correctly predicted. (The proportion of actual defaulters that the model will correctly predict)

Recall = TP / (TP + FN)

Recall(Sensitivity) - Recall is the ratio of correctly predicted positive observations to the all observations in actual class - yes.

F1 Score is the weighted average of Precision and Recall. Therefore, this score takes both false positives and false negatives into account. Intuitively it is not as easy to understand as accuracy, but F1 is usually more useful than accuracy, especially if you have an uneven class distribution. Accuracy works best if false positives and false negatives have similar cost. If the cost of false positives and false negatives are very different, it’s better to look at both Precision and Recall.

GENERAL FORMULA

F- Measure = 2TP / (2 TP + FP + FN)

F1-Score Formula:

F1 Score = 2*(Recall * Precision) / (Recall + Precision)

The below 4 different algorithms are compared:

- MLP(MULTI-LAYER PERCEPTRON)
- RANDOM FOREST CLASSIFIER
- VOTING CLASSIFIER
- LOGISTIC REGRESSION

MLP(MULTI-LAYER PERCEPTRON)

Multilayer Perceptrons are feedforward artificial neural networks that generate outputs from a set of inputs. In a Multilayer Perceptron, multiple layers of input nodes are connected as a directed graph between the input and output layers. The Multilayer Perceptron is a deep learning method that uses backpropagation to train the network. Though Perceptron are widely recognized as algorithms, they were originally designed for image recognition. It gets its name from performing the human-like function of perceiving, seeing, and identifying image .Multilayer Perceptron are essentially feed-forward neural networks with three types of layers The input layer receives the input signal for processing. The output layer performs tasks such as classification and prediction.

```
# Finding Accuracy
accuracy=accuracy_score(y_test,predicted)
print('Accuracy of MLPClassifier',accuracy*100)

Accuracy of MLPClassifier 58.032786885245905
```

```
# Finding Classification Report
cr=classification_report(y_test,predicted)
print('Classification Report\n\n',cr)

Classification Report

              precision    recall  f1-score   support

           0       0.91      0.73      0.81        82
           1       0.56      0.44      0.49       122
           2       0.44      0.62      0.52       101

    accuracy                           0.58       305
   macro avg       0.64      0.60      0.61       305
weighted avg       0.61      0.58      0.59       305
```

Fig 5.2.1 Accuracy and Classification Report of MLP

MODULE DIAGRAM

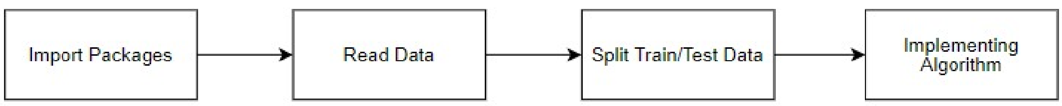


Fig 5.2.2 Module Diagram for MLP

RANDOM FOREST CLASSIFIER

Random Forest is a popular machine learning algorithm that belongs to the supervised learning technique. It can be used for both Classification and Regression problems in ML. It is based on the concept of ensemble learning, which is a process of combining multiple classifiers to solve a complex problem and to improve the performance of the model. Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset.

```
]: # Finding Accuracy
   accuracy=accuracy_score(y_test,predicted)
   print('Accuracy of Random Forest Classifier',accuracy*100)

   Accuracy of Random Forest Classifier 83.60655737704919
```

```
]: # Finding Classification Report
   cr=classification_report(y_test,predicted)
   print('Classification Report\n\n',cr)

   Classification Report

                 precision    recall  f1-score   support

              0       0.97      0.89      0.93        82
              1       0.83      0.83      0.83       122
              2       0.74      0.80      0.77       101

       accuracy                           0.84       305
      macro avg       0.85      0.84      0.84       305
   weighted avg       0.84      0.84      0.84       305
```

Fig 5.2.3 Accuracy and Classification Report of Random Forest

MODULE DIAGRAM

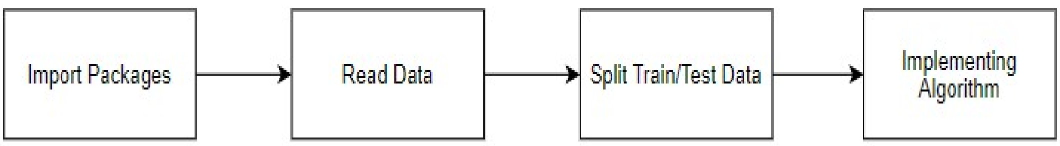


Fig 5.2.4 Module Diagram for Random Forest

VOTING CLASSIFIER

A Voting Classifier is a machine learning model that trains on an ensemble of numerous models and predicts an output (class) based on their highest probability of chosen class as the output. It simply aggregates the findings of each classifier passed into Voting Classifier and predicts the output class based on the highest majority of voting. The idea is instead of creating separate dedicated models and finding the accuracy for each of them, we create a single model which trains by these models and predicts output based on their combined majority of voting for each output class.

```
# Finding Accuracy
accuracy=accuracy_score(y_test,predicted)
print('Accuracy of Voting Classifier is: ',accuracy*100)

Accuracy of Voting Classifier is:  79.67213114754098

# Finding Classification Report
cr=classification_report(y_test,predicted)
print('Classification Report\n--------------------\n',cr)

Classification Report
--------------------
              precision    recall  f1-score   support

           0       0.97      0.93      0.95        82
           1       0.85      0.87      0.86       122
           2       0.80      0.81      0.80       101

    accuracy                           0.87       305
   macro avg       0.88      0.87      0.87       305
weighted avg       0.87      0.87      0.87       305
```

Fig 5.2.5 Accuracy and Classification Report of Voting Classifier

MODULE DIAGRAM

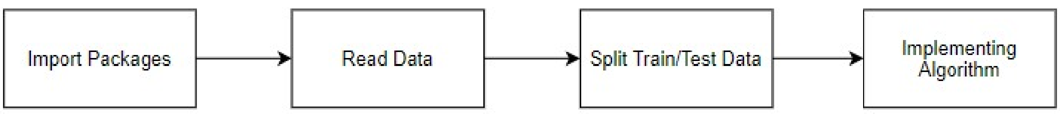


Fig 5.2.6 Module Diagram for Voting Classifier

LOGISTIC REGRESSION

Logistic regression is an example of supervised learning. It is used to calculate or predict the probability of a binary (yes/no) event occurring. An example of logistic regression could be applying machine learning to determine if a person is likely to be infected with COVID-19 or not. Since we have two possible outcomes to this question - yes they are infected, or no they are not infected - this is called binary classification. In this imaginary example, the probability of a person being infected with COVID-19 could be based on the viral load and the symptoms and the presence of antibodies, etc. Viral load, symptoms, and antibodies would be our factors , which would influence our outcome .

```
# Finding Accuracy
accuracy=accuracy_score(y_test,predicted)
print('Accuracy of LogisticRegression',accuracy*100)
```

```
Accuracy of LogisticRegression 60.32786885245902
```

```
# Finding Classification Report
cr=classification_report(y_test,predicted)
print('Classification Report\n\n',cr)
```

```
Classification Report

              precision    recall  f1-score   support

           0       0.84      0.70      0.76        82
           1       0.57      0.78      0.66       122
           2       0.46      0.32      0.38       101

    accuracy                           0.60       305
   macro avg       0.62      0.60      0.60       305
weighted avg       0.61      0.60      0.59       305
```

Fig 5.2.7 Accuracy and Classification Report of Logistic Regression

MODULE DIAGRAM

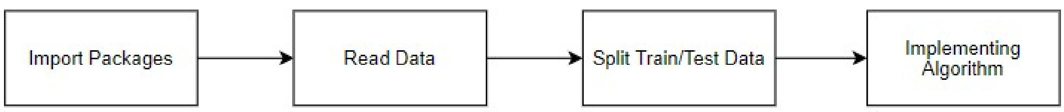


Fig 5.2.8 Module Diagram for Logistic Regression

DEPLOYMENT

FLASK(WEB FRAMEWORK)

Flask is a micro web framework written in Python. It is classified as a micro-framework because it does not require particular tools or libraries. It has no database abstraction layer, form validation, or any other components where pre-existing third-party libraries provide common functions. However, Flask supports extensions that can add application features as if they were implemented in Flask itself. Extensions exist for object-relational mappers, form validation, upload handling, various open authentication technologies and several common framework related tools. Flask was created by Armin Ronacher of Pocoo, an international group of Python enthusiasts formed in 2004. According to Ronacher, the idea was originally an April Fool's joke that was popular enough to make into a serious application. The name is a play on the earlier Bottle framework When Ronacher and Georg Brand created a bulletin board system written in Python, the Pocoo projects Werkzeug and Jinja were developed.In April 2016, the Pocoo team was disbanded and development of Flask and related libraries passed to the newly formed Pallets project.

Flask has become popular among Python enthusiasts. As of October 2020, it has second most stars on GitHub among Python web-development frameworks, only slightly behind Django, and was voted the most popular web framework in the Python Developers Survey 2018.

Flask is based on Werkzeug, Jinja2 and inspired by Sinatra Ruby framework, available under BSD licence.

FEATURES:

Flask was designed to be easy to use and extend. The idea behind Flask is to build a solid foundation for web applications of different complexity. From then on you are free to plug in any extensions you think you need. Also you are free to build your own modules. Flask is great for all kinds of projects. It's especially good for prototyping. Flask depends on two external libraries: the Jinja2 template engine and the Werkzeug WSGI toolkit.Still the question remains why use Flask as your web application framework if we have immensely powerful Django, Pyramid, and don't forget web mega-framework Turbo-gears? Those are supreme Python web frameworks BUT out-of-the-box Flask is pretty impressive too with its

- integrated support for unit testing
- RESTful request dispatching
- Uses Jinja2 Templating
- support for secure cookies
- Unicode based
- Extensive Documentation
- Google App Engine Compatibility
- Extensions available to enhance features desired

- Plus Flask gives you so much more CONTROL on the development stage of your project. It follows the principles of minimalism and let you decide how you will build your application.

- Flask has a lightweight and modular design, so it easy to transform it to the web framework you need with a few extensions without weighing it down

The configuration is even more flexible than that of Django, giving you plenty of solutions for every production need.To sum up, Flask is one of the most polished and feature-rich micro frameworks available. Still young, Flask has a thriving community, first-class extensions, and an elegant API. Flask comes with all the benefits of fast templates, strong WSGI features, thorough unit testability at the web application and library level, extensive documentation. So next time you are starting a new project where you need some good features and a vast number of extensions, definitely check out Flask.Flask is an API of Python that allows us to build up web-applications. It was developed by Armin Ronacher. Flask's framework is more explicit than Django framework and is also easier to learn because it has less base code to implement a simple web-ApplicationFlask is a micro web framework written in Python. It is classified as a micro-framework because it does not require particular tools or libraries. It has no database abstraction layer, form validation, or any other components where pre-existing third-party libraries provide common functions.Overview of Python Flask Framework Web apps are developed to generate content based on retrieved data that changes based on a user's interaction with the site. The server is responsible for querying, retrieving, and updating data. This makes web applications to be slower and more complicated to deploy than static websites for simple applications.Flask is an excellent web development framework for REST API creation. It is built on top of Python which makes it powerful to use all the python features.Flask is used for the backend, but it makes use of a templating language called Jinja2 which is used to create HTML, XML or other markup formats that are returned to the user via an HTTP request.

Django is considered to be more popular because it provides many out of box features and reduces time to build complex applications. Flask is a good start if you are getting into web development.

Advantages of Flask:

- Higher compatibility with latest technologies.
- Technical experimentation.

- Easier to use for simple cases.
- Codebase size is relatively smaller.
- High scalability for simple applications.
- Easy to build a quick prototype.
- Routing URLs is easy.
- Easy to develop and maintain applications.

Framework Flask is a web framework from the Python language. Flask provides a library and a collection of codes that can be used to build websites, without the need to do everything from scratch. But Framework flask still doesn't use the Model View Controller(MVC)method. Flask-RESTful is an extension for Flask that provides additional support for building REST APIs. You will never be disappointed with the time it takes to develop an API. Flask-Restful is a lightweight abstraction that works with the existing ORM/libraries. Flask-RESTful encourages best practices with minimal setup.

Start Using an API

- Most APIs require an API key. ...
- The easiest way to start using an API is by finding an HTTP client online, like REST-Client, Postman, or Paw.
- The next best way to pull data from an API is by building a URL from existing API documentation.
- The flask object implements a WSGI application and acts as the central object. It is passed the name of the module or package of the application.

CHAPTER 6

SYSTEM IMPLEMENTATION

6.1 Client Side Programming

MODULE 1

DATA VALIDATION AND PRE-PROCESSING TECHNIQUES

```
# DATA PRE-PROCESSING

import pandas as pd

import numpy as np

import warnings
warnings.filterwarnings('ignore')

data=pd.read_csv('Maternal_Health.csv')

data.head()

data.shape

data.size

df=data.dropna()

df

df.shape

df.size

df['RiskLevel'].unique()

df[data['RiskLevel']=='high risk']

df[data['RiskLevel']=='mid risk']

df[data['RiskLevel']=='low risk']

df.columns

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

```
a=df['Age'].unique()
a.sort()
print(a)
df['Age'].nunique()
print('Maximum Age:',df['Age'].max())
print('Minimum Age:',df['Age'].min())
print('Mean Of Age','%.2f'%df['Age'].mean())
print('Median Of Age','%.2f'%df['Age'].median())
print('Mode of Age','%.2f'%df['Age'].mode())
df.describe()
df.corr()
df.info()
pd.crosstab(df['RiskLevel'],df['Age'])
pd.crosstab(df['RiskLevel'],df['BodyTemp'])
pd.crosstab(df['RiskLevel'],df['HeartRate'])
pd.crosstab(df['RiskLevel'],df['BS'])
pd.Categorical(df['RiskLevel']).describe()
df.duplicated()
sum(df.duplicated())
```

MODULE 2

DATA VISUALIZATION

```
# Data Visualization
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
import warnings
warnings.filterwarnings('ignore')
data=pd.read_csv('Maternal_Health.csv')
df=data.dropna()
df.columns
pd.crosstab(df.RiskLevel,df.Age)
# Histogram
df.plot.hist()
plt.show()
plt.figure(figsize=(16,4))
plt.subplot(1,2,1)
plt.hist(df['BS'])
plt.title('Average Blood Sugar Levels')
plt.xlabel('Blood Sugar Level')
plt.ylabel('Total Patients')
plt.subplot(1,2,2)
plt.hist('BodyTemperature')
plt.title('Average Body Temperature Levels')
plt.xlabel('Body Temperature Level')
plt.ylabel('Temperature Range')
```

```
plt.show()
plt.figure(figsize=(9,6))
sns.scatterplot(x=df['Age'],y=df['BS'])
plt.show()
plt.boxplot(df['HeartRate'])
plt.show()
sns.boxplot(df['BS'],color='b')
sns.boxplot(df['Age'],color='g')
sns.boxplot(df['SystolicBP'])
plt.show()
sns.boxplot(df['DiastolicBP'])
plt.show()
fig,ax=plt.subplots(figsize=(10,5))
sns.stripplot(y=df['RiskLevel'],x=df['Age'],ax=ax)
plt.title('Age Wise Risk Levels')
plt.show()
def RiskLevel(df,variable):
   dataframe_pie=df[variable].value_counts()
   ax=dataframe_pie.plot.pie(figsize=(9,9),autopct='%1.2f%%',fontsize=10)
   ax.set_title(variable+' \n',fontsize=10)
   return np.round(dataframe_pie/df.shape[0]*100,2)
RiskLevel(df, 'RiskLevel')
plt.plot(df['RiskLevel'],df['Age'],color='m')
plt.xlabel('Risk Level')
plt.ylabel('Age')
plt.title('Age Wise Risk Level')
plt.show()
```

MODULE 3

```
# MLP Classifier
# import library packages
import pandas as pd
import matplotlib.pyplot as plt
import warnings
warnings.filterwarnings('ignore')
# Load Given Dataset
data=pd.read_csv('Maternal_Health.csv')
data.columns
df=data.dropna()
df.info()
from sklearn.preprocessing import LabelEncoder
label=LabelEncoder()
var=['RiskLevel']
for i in var:
df[i]=label.fit_transform(df[i]).astype(int)
df.head()
# preprocessing, split test and dataset, split response variable
X=df.drop(labels='RiskLevel',axis=1)
# Response Variable
y=df.loc[:,'RiskLevel']
# Splitting for train and test
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=
42,stratify=y)
print('Number of training Dataset:',len(X_train))
```

```
print('Number of test dataset:',len(X_test))
print('Total Number of dataset:',len(X_train)+len(X_test))
# Implementing MLP Classifier
from sklearn.neural_network import MLPClassifier
from sklearn.metrics import
confusion_matrix,classification_report,accuracy_score,plot_confusion_matrix
mlpc=MLPClassifier()
mlpc.fit(X_train,y_train)
predicted=mlpc.predict(X_test)
# Finding Accuracy
accuracy=accuracy_score(y_test,predicted)
print('Accuracy of MLPClassifier',accuracy*100)
# Finding Classification Report
cr=classification_report(y_test,predicted)
print('Classification Report\n\n',cr)
# Finding Confusion Matrix
cm=confusion_matrix(y_test,predicted)
print('Confusion Matrix\n\n',cm)
df2=pd.DataFrame()
df2['y_test']=y_test
df2['predicted']=predicted
df2.reset_index(inplace=True)
plt.figure(figsize=(20,5))
plt.plot(df2['predicted'][:100],marker='x',linestyle='dashed',color='red')
plt.plot(df2['y_test'][:100],marker='o',linestyle='dashed',color='green')
plt.show()
```

MODULE 4

```
# Random Forest
# import library packages
import pandas as pd
import matplotlib.pyplot as plt
import warnings
warnings.filterwarnings('ignore')
# Load Given Dataset
data=pd.read_csv('Maternal_Health.csv')
df=data.dropna()
df.columns
df.head()
from sklearn.preprocessing import LabelEncoder
label=['RiskLevel']
var=LabelEncoder()
for i in label:
    df[i]=var.fit_transform(df[i]).astype(int)
df.head()
# Preprocessing, split test and dataset, split response variable
X=df.drop(labels='RiskLevel',axis=1)
# Response Variable
y=df.loc[:,'RiskLevel']
# Splitting for train and test
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=
42,stratify=y)
```

```
print('Number of Training Dataset:',len(X_train))
print('Number of Test Dataset:',len(X_test))
print('Total Number of Dataset:',len(X_train)+len(X_test))
# Implementing Decision Tree
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import confusion_matrix, classification_report,
accuracy_score, plot_confusion_matrix
# Training
rf=RandomForestClassifier()
rf.fit(X_train,y_train)
predicted=rf.predict(X_test)
# Finding Accuracy
accuracy=accuracy_score(y_test,predicted)
print('Accuracy of Random Forest Classifier',accuracy*100)
# Finding Classification Report
cr=classification_report(y_test,predicted)
print('Classification Report\n\n',cr)
# Finding Confusion Matrix
cm=confusion_matrix(y_test,predicted)
print('Confusion Matrix\n\n',cm)
df2=pd.DataFrame()
df2['y_test']=y_test
df2['predicted']=predicted
df2.reset_index(inplace=True)
plt.figure(figsize=(20,5))
plt.plot(df2['predicted'][:100],marker='x',linestyle='dashed',color='viole')
plt.plot(df2['y_test'][:100],marker='o',linestyle='dashed',color='green')
plt.show()
```

```python
# Creating PKL File
import joblib
joblib.dump(rf,'maternal.pkl')
```

MODULE 5

```python
# Logistic Regression
# import library packages
import pandas as pd
import matplotlib.pyplot as plt
import warnings
warnings.filterwarnings('ignore')
# Load given Dataset
data=pd.read_csv('Maternal_Health.csv')
df=data.dropna()
df.columns
df.head()
from sklearn.preprocessing import LabelEncoder
var=['RiskLevel']
label=LabelEncoder()
for i in var:
    df[i]=label.fit_transform(df[i]).astype(int)
df.head()

# Preprocessing, split test and dataset, split response variable
X=df.drop(labels='RiskLevel',axis=1)
# Response Variable
```

```
y=df.loc[:,'RiskLevel']
# Splitting for Train and Test
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=
42,stratify=y)
print('Number of Training Dataset:',len(X_train))
print('Number of Test Dataset:',len(X_test))
print('Total Number of Dataset:',len(X_train)+len(X_test))
# Implementing LogisticRegression
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import confusion_matrix, classification_report,
accuracy_score, plot_confusion_matrix
 # Training
LR=LogisticRegression()
LR.fit(X_train,y_train)
predicted=LR.predict(X_test)
# Finding Accuracy
accuracy=accuracy_score(y_test,predicted)
print('Accuracy of LogisticRegression',accuracy*100)
# Finding Classification Report
cr=classification_report(y_test,predicted)
print('Classification Report\n\n',cr)
# Finding Confusion Matrix
cm=confusion_matrix(y_test,predicted)
print('Confusion Matrix\n\n',cm)
df2=pd.DataFrame()
df2['y_test']=y_test
```

```
df2['predicted']=predicted
df2.reset_index(inplace=True)
plt.figure(figsize=(20,5))
plt.plot(df2['predicted'][:100],marker='x',linestyle='dashed',color='red')
plt.plot(df2['y_test'][:100],marker='o',linestyle='dashed',color='b')
plt.show()
```

MODULE 6

```
# Voting Classifier
# import library packages
import pandas as pd
import matplotlib.pyplot as plt
import warnings
warnings.filterwarnings('ignore')
# Load Given Dataset
data=pd.read_csv('Maternal_Health.csv')
df=data.dropna()
df.columns
df.head()
from sklearn.preprocessing import LabelEncoder
var=['RiskLevel']
label=LabelEncoder()
for i in var:
   df[i]=label.fit_transform(df[i]).astype(int)
df.head()
# Preprocessing, Split test and dataset, split response variable
```

```
X=df.drop(labels='RiskLevel',axis=1)
# Response Variable
y=df.loc[:,'RiskLevel']
# Splitting for Train and Test
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=
42,stratify=y)
print('Number of Training Dataset:',len(X_train))
print('Number of Test Dataset:',len(X_test))
print('Total Number of Dataset:',len(X_train)+len(X_test))
# Implementing Voting Classifier
from sklearn.ensemble import VotingClassifier
from sklearn.neural_network import MLPClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import
confusion_matrix,classification_report,accuracy_score,plot_confusion_matrixm
lpc=MLPClassifier()
rf=RandomForestClassifier()
LR=LogisticRegression()
vc=VotingClassifier(estimators=[('MLP Classifier',mlpc),('Random Forest
Classiifer',rf),('LogisticRegression',LR)],voting='soft')
# Training
vc.fit(X_train,y_train)
predicted=vc.predict(X_test)
# Finding Accuracy
accuracy=accuracy_score(y_test,predicted)
print('Accuracy of Voting Classifier is: ',accuracy*100)
```

```
# Finding Classification Report
cr=classification_report(y_test,predicted)
print('Classification Report\n---------------------\n',cr)
# Finding Confusion Matrix
cm=confusion_matrix(y_test,predicted)
print('Confusion Matrix\n----------------\n',cm)
df2=pd.DataFrame()
df2['y_test']=y_test
df2['predicted']=predicted
df2.reset_index(inplace=True)
plt.figure(figsize=(20,5))
plt.plot(df2['predicted'][:100],marker='x',linestyle='dashed',color='red')
plt.plot(df2['y_test'][:100],marker='o',linestyle='dashed',color='green')
plt.show()
```

6.2 Server Side Programming

MODULE 7

```
# Deployment
import numpy as np
from flask import Flask, request, jsonify, render_template
import pickle
import joblib
import serial
app = Flask(__name__)
model = joblib.load('maternal.pkl')
ser = serial.Serial()
```

```
ser.port = 'COM7'
ser.baudrate = 9600
ser.bytesize = 8
ser.parity = serial.PARITY_NONE
ser.stopbits = serial.STOPBITS_ONE
def serialget():
    value=[]
    print('serialget')
    ser.open()
    v=b'A'
    ser.write(v)
    print('data sent')
    while True:
        for line in ser.read():
            if chr(line) != '$':
                value.append(chr(line))
            else:
                print("end")
                ser.close()
                return value

@app.route('/request1')
def request1():
    str1=''
    val=[]
    va=serialget()
    print(va)
    for v in va:
        #print('request for')
```

```
  if(v=='*'):
            continue
        else:
            if(v!='#'):
                str1+=v
                #print(str1)
            else:
                print(str1)
                val.append(float(str1))
                str1=""
    print(val)
    return render_template('index.html',val1=val[0],val2=val[1])
@app.route('/')
def home():
    return render_template('index.html')
@app.route('/predict',methods=['POST'])
def predict():
    '''
    For rendering results on HTML GUI
    '''
    int_features = [(x) for x in request.form.values()]
    final_features = [np.array(int_features).astype(float)]
    print(final_features)
    prediction = model.predict(final_features)
    print(prediction)
    output = prediction[0]
    if output == 0:
        return render_template('index.html', prediction_text='High Risk')
    elif output == 1:
```

```
        return render_template('index.html', prediction_text='Low Risk')
    elif output == 2:
        return render_template('index.html', prediction_text='Mid Risk')
    print(output)
if __name__ == "__main__":
    app.run(host="localhost", port=7000)
```

CHAPTER 7

TESTING

7.1 UNIT TESTING

This is the first level of testing. These different modules are tested against the specifications produced during the design of the module. During this testing the number of arguments is compared to input parameters, matching of parameters and arguments etc. It is also ensured whether the file attributes are correct, whether the Files are opened before using, whether Input/output errors are handled etc.

7.2 INTEGRATION TESTING

Integration testing is a software testing technique that involves testing the integration of individual software modules or components to verify that they work together correctly as a system. The goal of integration testing is to identify any defects or issues that may arise from the interaction between different modules or components. System Integration Testing is a process of testing the interactions and interfaces between different components or systems in a larger software system. This type of testing is essential for ensuring that all the components of the system are working together correctly and producing accurate and reliable results. In the context of Maternal Health Risk using classification model, System Integration Testing involves testing the interactions between the classification model and other components such as data pre-processing, classification, and output generation. This type of testing ensures that the system is capable of accurately predicting the risk of maternal health.

The first step in System Integration Testing is to identify the components or systems that need to be integrated and tested. In the case of Maternal Health risk prediction using classification model, this may include components for data pre-processing, classification, and output generation. Once the components have been identified, the next step is to define the interfaces between the components. This involves defining the inputs and outputs of each component and how they interact with each other. After defining the interfaces, the components are integrated and tested as a system. This involves testing the system end-to-end, from input data to output results, to ensure that all components are working together correctly and producing the desired outcomes. This testing is typically performed under different scenarios and conditions to ensure that the system can handle different types of input data and perform accurately and reliably in different settings. Finally, any issues or limitations identified during System Integration Testing are addressed and resolved. This ensures that the system is capable of accurately predicting the risk and producing reliable results in real-time scenarios.

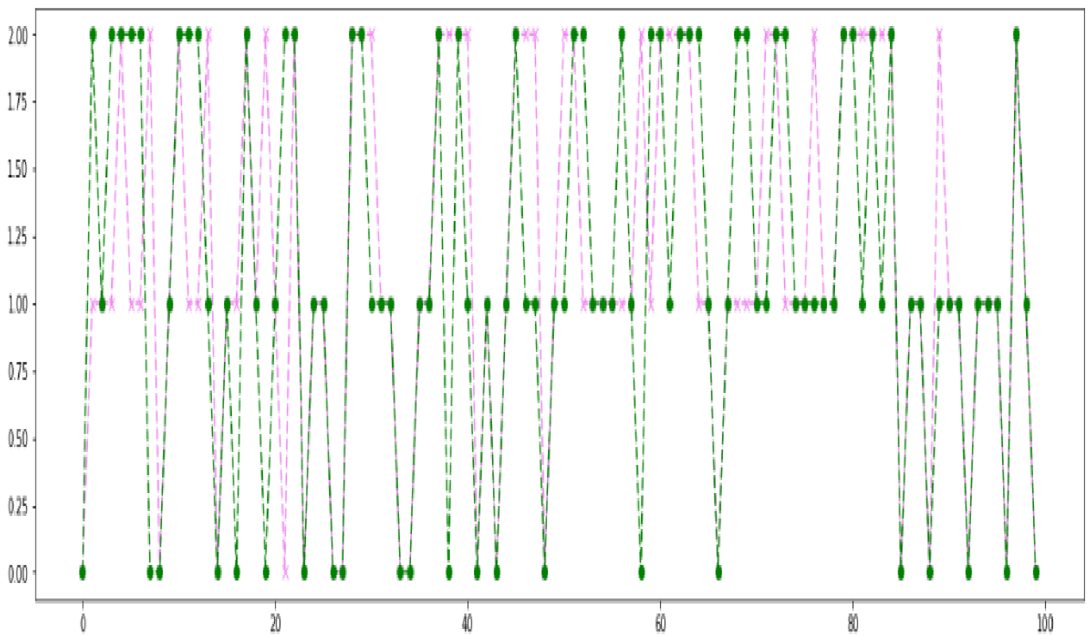


Fig 7.2.1 Random Forest Training And Testing Analysis

7.3 TEST CASES & REPORTS

TEST CASE ID	INPUT	EXPECTED OUTPUT	OBTAINED OUTPUT	PASS/FAIL	REMARKS
TC01	Dataset	Successful import	Successful import	Pass	Imported successfully
TC02	Dataset	Pre-process successful	Pre-process successful	Pass	Pre-processed successfully
TC03	Dataset	Model creation	Model created	Pass	Model created successfully
TC04	Model	Successful compilation	Successful compilation	Pass	Model compiled successfully
TC05	Model	Successful loading of model	Successful loading of model	Pass	Model loaded successfully
TC06	Health Parameters	Successful Prediction of low risk	Successful Prediction of low risk	Pass	Prediction Successful
TC07	Health Parameters	Successful Prediction of low risk	Successful Prediction of low risk	Pass	Prediction Successful
TC08	Health Parameters	Successful Prediction of low risk	Successful Prediction of low risk	Pass	Prediction Successful

Table 7.1 Test Cases and Report

CHAPTER 8

CONCLUSION

8.1 RESULTS AND DISCUSSIONS

The inputs Heart Rate and Temperature are provided through sensors and other health conditions are provided manually. When the sensors measure the temperature and Heart rate it shows whether the mother has normal Temperature and Heart rate. When the health conditions of a mother is provided to the model it predicts the risk level of the maternal health.The results of our proposed system for Maternal Health Prediction using the Classification model trained on a real-life situations dataset are promising. The Random Forest model achieved a higher training accuracy of 92.38% and the testing accuracy of 84.26%. From the results provided by the model the Random Forest tends to be an ideal algorithm for predicting maternal health. The Testing and Training analysis of the model the number of mothers in each category is predicted as shown in the (**Fig.**).The number of mothers in High risk is found to be 40.04%. The number of mothers in Mid risk is found to be 33.14%.The number of mothers in Low risk is found to be 26.82% .

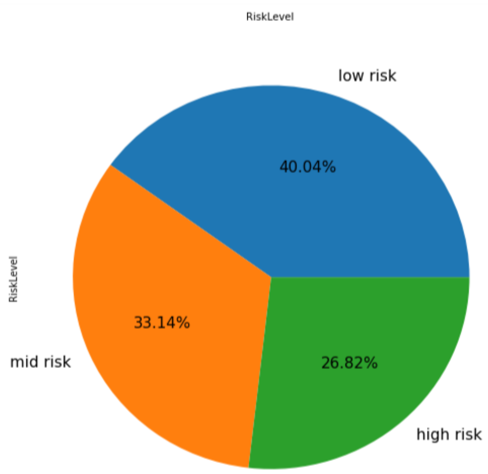


Fig 8.1.2 Risk Prediction

8.2 CONCLUSION AND FUTURE ENHANCEMENT

The analytical process started from data cleaning and processing, missing value, exploratory analysis and finally model building and evaluation. The Best accuracy on a public test set is higher accuracy score will be found out. This model can predict a woman's maternal health status based on the comparison of four algorithms. It may also allow for the early provision of additional assistance for women who are at risk of not giving birth in a medical institution, potentially improving maternal and fetal health outcomes. The paradigm described here is transferable to different situations, and the choice of input features can be readily modified to meet data accessibility and various outcomes, including those related to an unrelated to maternal health.

FUTURE ENHANCEMENT

- Maternal Health Prediction to connect the AI Model.
- To automate this process by showing the prediction result in web application or desktop application.
- To optimize the work to implement an AI Environment.

APPENDICES

A.1 SAMPLE SCREENS

A.1.1 FRONT PAGE

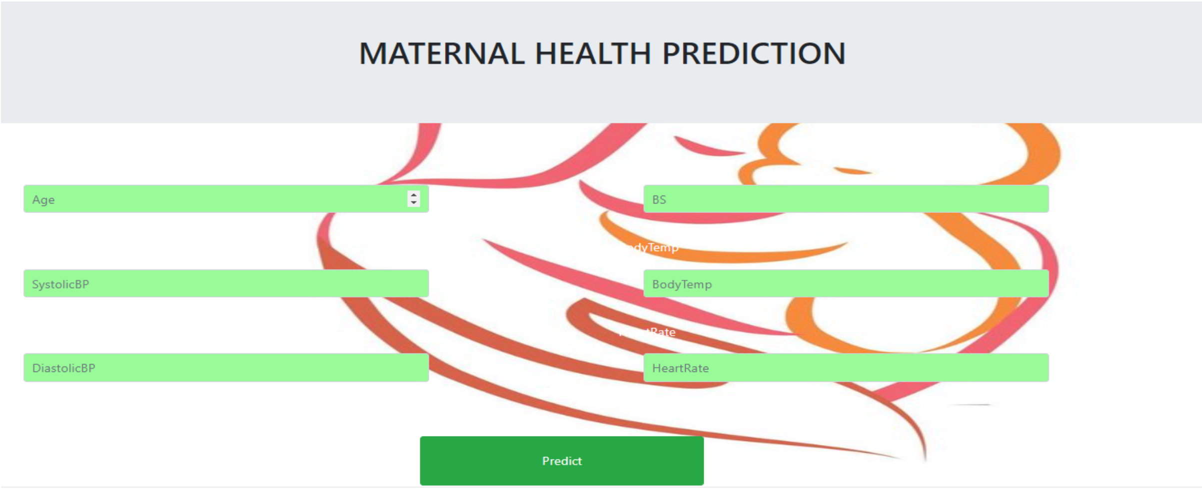


Fig A.1.1 Front Page For Maternal Health Prediction System

A.1.2 LOW RISK PREDICTION

Fig A.1.2 Risk Prediction for Low Level Risk

A.1.3 MID RISK PREDICTION

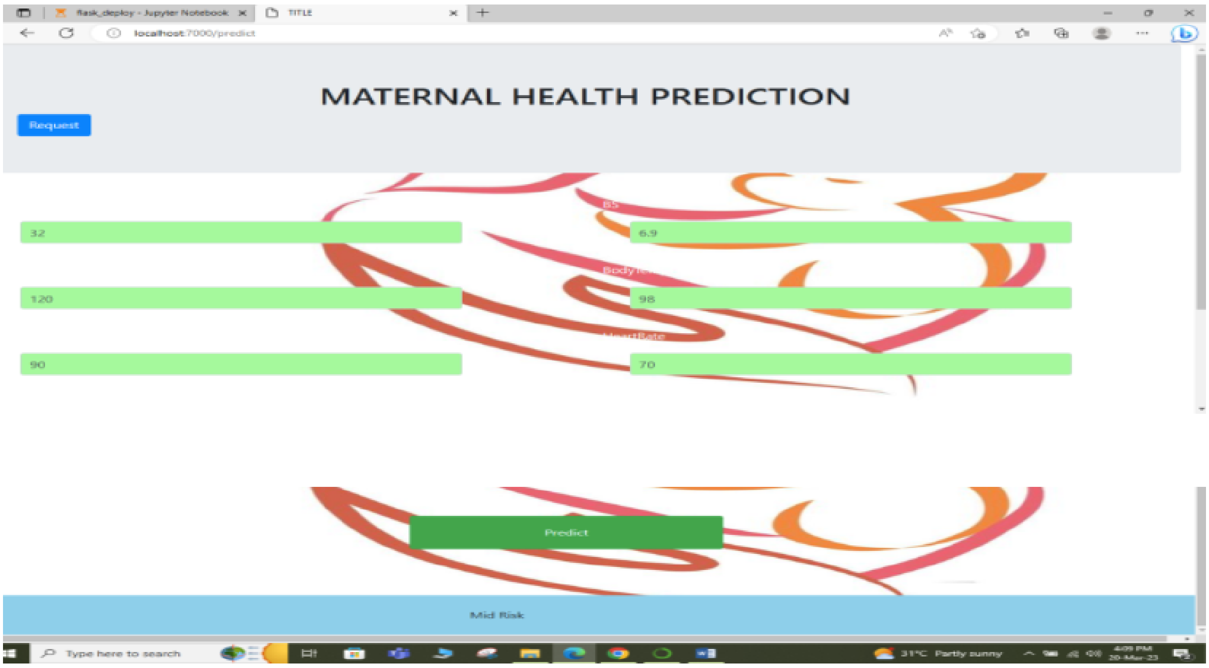


Fig A.1.3 Risk Prediction for Mid Level Risk

A.1.4 HIGH RISK PREDICTION

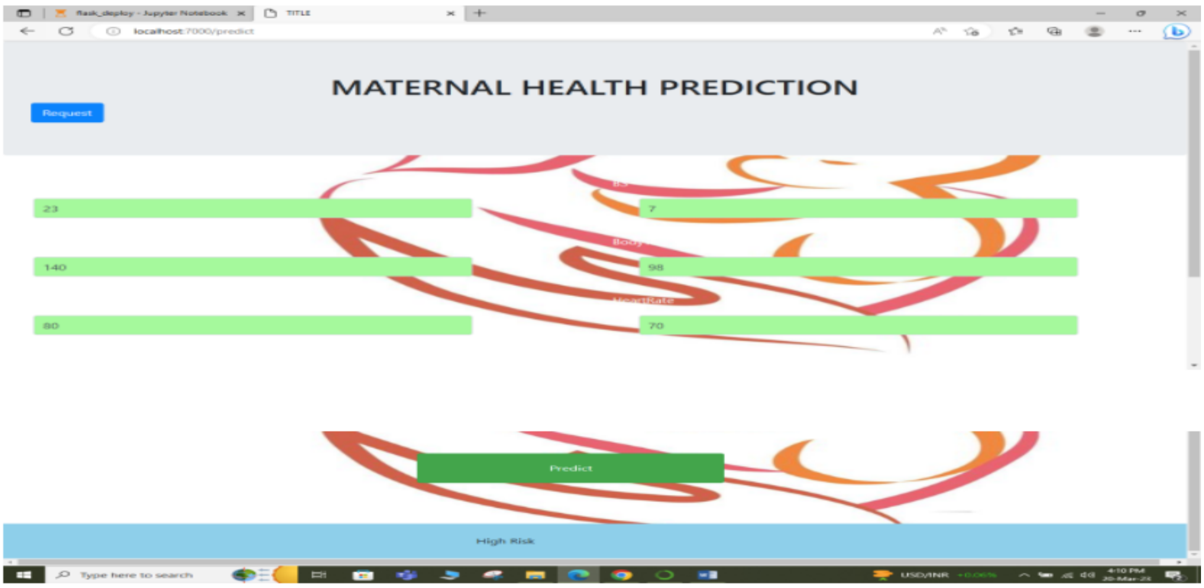


Fig A.1.4 Risk Prediction for High Level Risk

REFERENCES

[1] Quazi Adibur Rahman Adib Sidratul Tanzila Tasmi;,Shahriar Islam

Bhuiyan, Mohsin Sarker Raihan,etal,(2022)"Prediction Model for Mortality Analysis of Pregnant Women Affected With COVID-19".

[2]L. Meng, K. Ge, Y. Song, D. Yang, and Z. Lin,(2021.)“Wearable electrocardiogram signal monitoring and analysis based on convolutional neural network,”

[3] Xiaojia Wang,Yurong Wang, Shanshan Zhang, Lushi Yao, Sheng Xu,(2022)"Analysis and Prediction of Gestational Diabetes Mellitus by the Ensemble Learning Method".

[4]Julio Jerison E. Macrohon ,Charlyn Nayve Villavicencio , X. Alphonse Inbaraj and Jyh-Horng Jeng,(2022) "A Semi-Supervised Machine Learning Approach in Predicting High-Risk Pregnancies in the Philippines".

[5] Nirmala, Rekha S Kambli (2023) "Maternal Risk Level Prediction Using Ensemble Model",Volume 11.

[6] Faheem Akhtar,Jianqiang Li,Janith Zahid Hussain Khand,etal,(2022)"An Efficient Small for Gestational Age Prognosis System Using Stacked Generalization Scheme (SGS)",DOI: 10.1109/ COMPSAC 54236.2022.00231

[7]Ali Raza,Hafeez Ur Rehman Siddiqui,Kashif Munir,Mubarak Almutairi ,Furqan Rustam,Imran Ashraf,(2022)"Ensemble learning-based feature engineering to analyze maternal health during pregnancy and health risk prediction"https://doi.org/10.1371/journal.pone.027655

[8] Dawit S. Bogale, Tesfamariam MAbuhay ,Belayneh E. Dejene,(2022)"Predicting perinatal mortality based on maternal health status and health insurance service using homogeneous ensemble machine learning methods",https://doi.org/10.1186/s12911-022-02084-1

[9] Argyro Mavrogiorgou , Athanasios Kiourtis , Spyridon , etal ,(2022)"A Catalog of Machine Learning Algorithms for Healthcare Risk Predictions",Doi:10.3390/s22228615

[10] Alma Fredriksson, Isabel R. Fulcher, Allyson L. Russell,etal,(2022)"Machine learning for maternal health: Predicting delivery location in a community health worker program in Zanzibar",Volume4, https://doi.org/10.33/fdgth.2022.855236.

[11]Muhammad Nazrul Islam, Sumaiya Nuha Mustafina, Tahasin Mahmud & Nafiz Imtiaz Khan(2022)"Machine learning to predict pregnancy outcomes: a systematic review, synthesizing framework and future research agenda ",Article number: 348.

[12] João Alexandre Lobo Marques,Tao Han, Wanqing Wu,Aloísio Vieira Lira Neto,etal,(2021) "IoT-Based Smart Health System for Ambulatory Maternal and FetalMonitoring",DOI:10.1109/JIOT.2020.3037759.

[13] Saswati Chaudhuri , Biswajit Mandal(2021) "Predictive behavior of maternal health inputs and child mortality in West Bengal – An analysis based on NFHS-3" , https://doi.org/10.1016/j.heliyon.2020.e03941

[14] Rema Ramakrishnan, Shishir Rao,Jian-Rong He,(2021)"Perinatal health predictors using artificial intelligence: A review",DOI: 10.1177/17455065211046132

[15] KabirAhmad,Enamul Kabir,(2021)"Maternal health and health-related behaviors and associations with child health: Evidence from an Australian birth cohort", https://doi.org/10.1371/journal.pone.0257188