School of Engineering Technology

Main Campus, Off Hennur-Bagalur Main Road, Chagalahatti, Bengaluru 562149

CAPSTONE PROJECT DESIGN REPORT

"FETAL HEALTH PREDICTION USING ML APPROACH"

By

B.G NAGADARSHAN [20BBTCS022]
CHANDAN V [20BBTCS030]
K JEEVAN REDDY [20BBTCS066]
SUPRIYA N [20BBTCS149]

Submitted in partial fulfillment of the requirement for the award of the degree of
BACHELOR OF TECHNOLOGY
in
COMPUTER SCIENCE AND ENGINEERING

Under the guidance of,

Prof. Sivakumar N

DEPARTMENT OF CSE
School of Engineering & Technology
CMR UNIVERSITY

2023-2024

School of Engineering and Technology, CMR

Main Campus, Off Hennur-Bagalur Main Road, Chagalahatti, Bengaluru-562149

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

Certified that the ***Capstone Project Design*** *Work entitled* ***"FETAL HEALTH PREDICTION USING ML APPROACH"*** *has been successfully carried out by* ***B.G NAGADARSHAN, CHANDAN V, K JEEVAN REDDY, SUPRIYA N bearing*** *USN* ***20BBTCS022, 20BBTCS030, 20BBTCS066, 20BBTCS149,*** *in partial fulfillment for the award of the* ***BACHELOR OF TECHNOLOGY*** *in* COMPUTER SCIENCE AND ENGINEERING by SCHOOL OF ENGINEERING AND TECHNOLOGY, CMR UNIVERSITY, *during the year 2023-24. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the report deposited in the departmental library. The capstone project report has been approved as it satisfies the academic requirements concerning the project work prescribed for the said Degree.*

Signature of guide	Signature of HOD	Signature of Dean
Prof. Sivakumar N	Dr. Rubini P	Dr.V. R Manjunath
Assistant Professor	Professor & Head, Dept. of CSE Dean	Dean, SOET, CMR University,
Department of CSE,	School of Engineering & Technology,	Lakeside Campus,
School of Engineering & Technology,	CMR University, Bangalore	Bangalore.
CMR University, Bangalore		

<u>Examiners</u>

Name of the examiner	Signature with date
1	
2	

DECLARATION

*We, **B.G NAGADARSHAN, CHANDAN V, K JEEVAN REDDY, SUPRIYA N,** students of School of Engineering and Technology, CMR University, hereby declares that the dissertation titled **"FETAL HEALTH PREDICTION USING ML APPROACH"** embodies the report of our capstone project carried out independently by us during the seventh semester of **Bachelor of Technology in Computer Science and Engineering,** under the supervision of **Prof. Sivakumar, Assistant Prof.,** Department of Computer Science and Engineering and this work has been submitted in partial fulfillment for the award of the **Bachelor of Technology degree.***

We have not submitted the project for the award of any other degree from any other university or institution

Date:

Place: Bengaluru

B.G NAGADARSHAN(USN:20BBTCS022)

CHANDAN V(USN:20BBTCS030)

JEEVAN REDDY(USN:20BBTCS66)

SUPRIYA N(USN:20BBTCS149)

ACKNOWLEDGEMENT

We express our foremost gratitude to ***Dr. V.R. Manjunath,*** *Dean, SOET, CMR University, Lakeside Campus, Bagalur for his constant support and guidance.*

We would like to express our thanks to ***Dr. Rubini P****, Professor and Head, Dept. of CSE, and* ***Dr.S. Saravanakumar****, Professor and Head, Dept. of IT, CCE, AIML & DS, SOET, CMR University, Lakeside Campus, Bagalur for their encouragement that motivated us for the successful completion of project work.*

We express our heartfelt sincere gratitude to ***Dr. Manjunath C R and Mr. Prabhakar. K****, Capstone Project Coordinators, Dept. of CSE, SOET, CMR University, Lakeside Campus, Bagalur for their constant support and guidance.*

We consider ourselves privileged to express gratitude and respect to our Project Guide ***Mr. Sivakumar N****, Assistant Prof., Dept of Computer Science & Engineering, SOET, CMR University, Lakeside Campus, Bagalur for her inspiration, adroit guidance, constant supervision, direction, and discussions in the successful completion of this project work.*

Our sincere thanks to all teaching and non-Teaching Staff of Computer Science & Engineering for all the facilities provided, without which, we could not have progressed with my work. Thanks to our parents who have been a great source of strength in the completion of this project work.

B.G NAGADARSHAN (USN:20BBTCS022)

CHANDAN V (USN:20BBTCS030)

K. JEEVAN REDDY (USN:20BBTCS066)

SUPRIYA N (USN:20BBTCS149)

ABSTRACT

Using machine learning techniques, this study aims to accurately forecast the baby weight during pregnancy and conduct a thorough assessment of fetal health. The main goal is to create a strong classification model that can recognize possible dangers to the health of the fetus based on a range of maternal and fetal characteristics. The dataset contains vital characteristics that are used to train a classification system, including maternal age, weight growth, blood pressure, and ultrasound measures. In order to provide physicians with useful information for improved prenatal care, the project also intends to create a regression model for predicting newborn weight. By proactively managing pregnancies and detecting difficulties early, the integration of these models into clinical practice may ultimately improve outcomes for mothers and newborns.

Proposed system Solves the critical task of fetal health classification and baby weight prediction using advanced machine learning techniques. The research encompasses the development of a robust model for classifying fetal health conditions based on a comprehensive set of maternal and fetal parameters. These parameters include maternal age, weight gain, blood pressure, and ultrasound measurements, contributing to a holistic assessment of potential risks. Concurrently, the project aims to create a regression model to predict baby weight, enabling a more accurate estimation that can guide healthcare professionals in tailoring prenatal care. By integrating these machine learning models into clinical practice, this project strives to enhance the accuracy of early risk identification and improve overall prenatal care, ultimately contributing to better maternal and neonatal outcomes.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF SNAPSHOTS

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 Introduction

Prenatal care is a pivotal aspect of maternal and neonatal health, and advancements in machine learning present an unprecedented opportunity to enhance the precision and effectiveness of this care. This project focuses on the development and application of machine learning techniques for fetal health classification and the prediction of baby weight during pregnancy. In contemporary healthcare, the ability to identify potential complications early in pregnancy is crucial for proactive intervention and improved outcomes. This project seeks to harness the power of predictive modeling to assist healthcare professionals in making informed decisions based on a comprehensive analysis of maternal and fetal parameters.

The first facet of the project involves the classification of fetal health conditions using a diverse dataset encompassing essential factors such as maternal age, weight gain, blood pressure, and ultrasound measurements. By employing sophisticated machine learning algorithms, the model aims to categorize pregnancies into various health classes, aiding in the early detection of potential risks. Concurrently, the second aspect of the project focuses on predicting baby weight, a critical metric for assessing fetal development. The integration of a regression model allows for a more accurate estimation of baby weight based on a combination of maternal characteristics and ultrasound findings.

This research holds significant implications for improving prenatal care by providing healthcare professionals with tools to tailor interventions based on individualized risk assessments. Concurrently, the second aspect of the project focuses on predicting baby weight, a critical metric for assessing fetal development The amalgamation of classification and regression models aims to create a comprehensive framework that contributes to the optimization of prenatal care strategies, ultimately leading to improved outcomes for both mothers and newborns.

1.2 Motivation

The motivation behind this project stems from the pressing need to enhance prenatal care through the integration of cutting-edge machine learning techniques. Traditional methods of assessing fetal health and predicting baby weight often rely on subjective interpretations, leading to potential oversights and delayed interventions. By leveraging the power of machine learning, we aspire to revolutionize the landscape of prenatal care, offering healthcare professionals a sophisticated toolkit for early risk identification and personalized care strategies.

The below Fig 1.1 the potential impact is immense, as the project seeks to empower clinicians with predictive models that can analyze a myriad of maternal and fetal parameters, fostering a more proactive approach to managing pregnancies. Ultimately, our motivation is rooted in the belief that the amalgamation of technology and healthcare can significantly improve maternal and neonatal outcomes, paving the way for a new era of precision prenatal care.

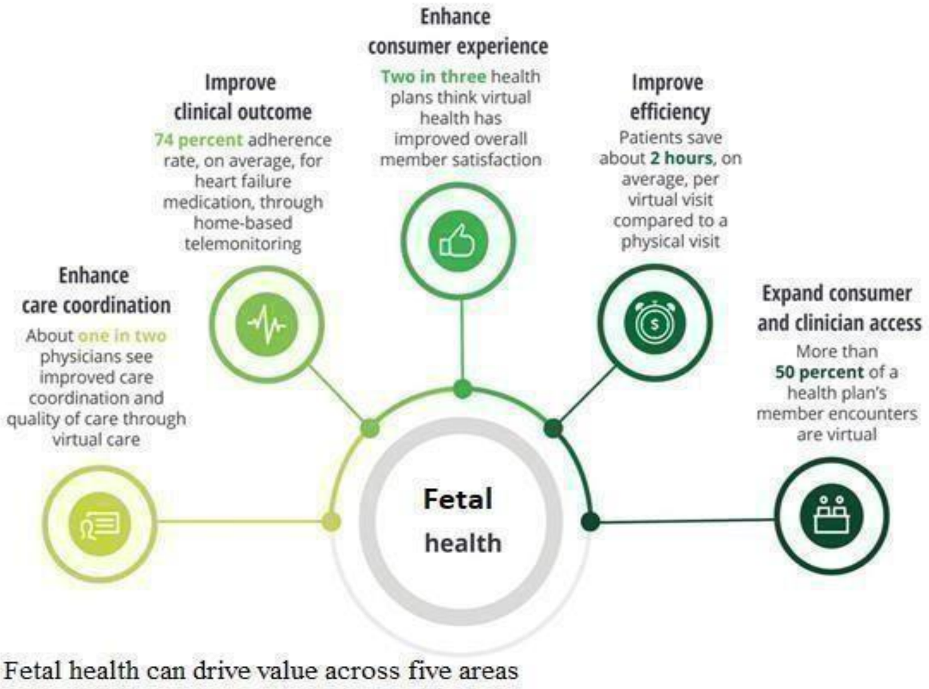


Fig 1.1: Areas of fetal health prediction

1.3 Problem Statement

Despite advancements in medical technology, the current landscape of prenatal care faces challenges in early risk detection and personalized intervention strategies. Traditional methods for assessing fetal health and predicting baby weight often lack the precision required for proactive management. The inherent complexity of maternal-fetal interactions demands a more comprehensive approach that goes beyond conventional clinical assessments. This project addresses the need for a robust solution by utilizing machine learning techniques to develop accurate classification models for fetal health and regression models for predicting baby weight. The overarching problem is the inadequacy of current practices to provide timely and precise insights into potential complications during pregnancy, underscoring the urgency for innovative approaches that leverage data-driven methodologies to enhance the quality of prenatal care.

1.4 Scope of the Project

The Scope of the Proposed work “Fetal Health Prediction” are:

- This proposed work scope encompasses the development and implementation of machine learning models to significantly enhance the precision and efficiency of prenatal care.
- By focusing on fetal health classification and baby weight prediction, the project aims to create tools that can be integrated into clinical practice, providing healthcare professionals with timely and accurate information.
- The scope extends to the exploration of diverse datasets, incorporating key maternal and fetal parameters, to ensure the models' robustness across a variety of clinical scenarios.
- Additionally, the project aims to offer a user-friendly interface for healthcare practitioners to seamlessly incorporate these predictive models into their decision-making processes.
- The ultimate goal is to contribute to a paradigm shift in prenatal care, fostering early intervention and personalized strategies that improve maternal and neonatal outcomes.

1.5 Objectives

The Objectives of the Proposed work “Fetal Health Prediction” are:

- To develop machine learning models for accurate classification of fetal health conditions, integrating maternal age, weight gain, blood pressure, and ultrasound measurements.
- To implement a regression model to predict baby weight using a combination of maternal characteristics and ultrasound findings.
- To explore and preprocess diverse datasets to ensure the robustness and generalizability of the developed models across various clinical scenarios. To implement a regression model to predict baby weight using a combination of maternal characteristics and ultrasound findings
- To create an intuitive user interface for healthcare professionals to seamlessly integrate the predictive models into their decision-making process, promoting proactive and personalized prenatal care.

1.6 Review of Literature

A Preterm Birth Risk Prediction System for Mobile Health Applications Based on the Support Vector Machine Algorithm.

Published in: 2018 IEEE International Conference on Communications (ICC)

[1] A Review of Recent Advances and Future Developments in Fetal Phonocardiography authored by Radana Kahankova, Martina Mikolasova, Rene Jaros, and Katerina Barnova, published in 2023.

Fetal phonocardiography is a promising method for fetal monitoring which is both non-invasive and passive. This makes it suitable for continuous monitoring of heart rate variability compared to the techniques currently used in the clinical practice. Moreover, it provides information about mechanical activity of the fetal heart, that is not contained in the other rising method, fetal electrocardiography. The greatest issue associated with this method is the interference from various sources, overlapping the desired signal in the time and frequency domain This makes it suitable for continuous monitoring of heart rate variability compared to the techniques currently used in the clinical practice.[1]

[2] Estimating Gestational Age From Maternal-Fetal Heart Rate Coupling Parameters, was authored by Maisam Wahbah, Raghad Al Sakaji, Kiyoe Funamoto, Anita Krishnan, Yoshitaka Kimura, and Ahsan H. Khandoker, and it was published in 2021.

Methodology: Analytical study focusing on the correlation between maternal and fetal heart rate coupling parameters to develop a method for estimating gestational age.

The results presented in this paper successfully showed that maternal and fetal physiological parameters including maternal-fetal HR coupling parameters at various ratios and fetal HRV parameters produce a reliable estimate of the GA utilizing a multivariate regression model based on recorded ECG signals for 5 min rather than 1 min recordings maternal/fetal HRV parameters produce a reliable estimate of the GA utilizing a multivariate regression model based on recorded ECG signals for 5 min rather than 1 min recordings. heart anomalies and arrhythmias coupling parameters at various ratios and fetal HRV parameters produce a reliable estimate of the GA utilizing a multivariate regression model based on recorded ECG signals for 5 min rather than 1 min recording. [2]

[3] A Novel Ultrasonic Doppler Fetal Heart Rate Detection System Using Windowed Digital Demodulation" was authored by Ming Dai, Kai Zhan, Rongchao Peng, Jinfeng Xu, Hui Luo, Yingying Liu, Liangping Luo, Huiying Wen, and Siping Chen, and it was published in 2021.

Methodology: Development and evaluation of a novel ultrasonic Doppler system utilizing windowed digital demodulation for accuracy in fetal heart rate detection.

The TREND test curve was generated by the FS-3 FHR simulation, the comparison curve between the TREND curve generated by the FS-3 FHR simulation and the measured curve of our proposed FHR detector was obtained. A comparative experiment between the commercial FM-3A FHR detector and our proposed FHR detector was carried out. The FS-3 FHR simulator produces eight test FHRs, and the generated FHRs were measured by our FHR detector and the commercial FM-3A FHR detector.[3]

[4] Multi-Chain Semi-Markov Analysis of Intrapartum Cardiotocography was authored Johann Vargas-Calixto, Yvonne Wu, Michael Kuzniewicz, Marie-Coralie Cornet, Heather Forquer, Lawrence Gerstley, Emily Hamilton, Philip A. Warrick, and Robert E. Kearney, and it was published in 2022.

Methodology: Application of multi-chain semi-Markov analysis to intrapartum and cardiotocography data, offering a refined understanding of fetal heart rate patterns during labor. We showed that computerized tools such as Peri CALM Patterns could be used to evaluate CTG patterns objectively Our results show that MCSMMs are a promising approach to the analysis of CTG and the early detection of fetuses at risk of developing HIE. Thus, we demonstrated that there were significant differences in some MCSMM parameters as early as 12 hours before birth, encouraging their subsequent use in machine learning. In future work, CTG patterns objectively Our results show that MCSMMs are a promising approach to the analysis of CTG and the early detection of fetuses at risk of developing HIE we will integrate these parameters with other FHR variability measures such as power spectral density and nonlinear features in a machine learning procedure aimed at predicting the increased risk of developing HIE from CTG risk of developing HIE. Thus, we demonstrated that there were significant differences in some MCSMM parameters as early as 12 hours before birth, encouraging their subsequent use in machine learning. In future work, CTG patterns objectively Our results show that MCSMMs are a promising approach to the analysis of CTG and the early detection of fetuses at risk of developing HIE.[4]

1.7 Organization of the Report

The report is organized into the chapters as follows:

Chapter -1 Introduction: The chapter presents a brief description about Fetal Health Prediction. Prenatal care is a pivotal aspect of maternal health, and medical.

Chapter -2 System requirement specification: The chapter 2 presents the specific requirement, software and hardware requirements interfaces used. It also presents a brief summary about the chapter.

Chapter -3 High level Design: The chapter 3 presents the Design consideration made, system architecture of proposed system, specification of the proposed system using use case diagram. It also describes module specification, data flow diagram for every module and state chart for the proposed method. Finally, it presents a brief summary about the chapter.

Chapter -4 Detailed design: The chapter 4 briefs about the structural chart diagram and detail functionality and description of each module.

1.8 Summary

The introduction chapter serves as the foundational framework for the entire project, setting the stage by presenting the context, motivation, and problem statement. The initial portion provides a comprehensive overview of the significance of prenatal care and the critical need for advancements in risk detection and intervention strategies. The motivation behind the project is outlined, emphasizing the potential transformative impact of machine learning on the field of maternal-fetal health. A clear problem statement articulates the existing challenges in conventional prenatal care practices, highlighting the gaps that the project aims to address.

CHAPTER 2

SYSTEM REQUIREMENT SPECIFICATION

System requirement specifications gathered by extracting the appropriate information to implement the system. It is the elaborative conditions which the system needs to attain. Moreover, the SRS delivers a complete knowledge of the system to understand what this project is going to achieve without any constraints on how to achieve this goal. This SRS does not providing the information to outside characters but it hides the plan.

2.1 Specific Requirements

2.1.1 Functional Requirements:

- Prerequisite investigation is a computer programming task that overcomes any issues among machine degree programming portion and programming program format. Related to a decline in neonatal grimness and mortality.
- It permits the device engineer to specify software interface with different device factors & establishes layout constraints that the software need to satisfy.
- It presents the software application format with a representation of facts & function that may be translated to this point, architecture & procedural format.

2.1.2 Non-Functionality Requirements:

- **Security:** Mission degree protection is ready. User desires to login when they start this machine alternative is likewise provided to create the more person and degree safety. Currently customer stage safety isn't set however can be implemented with few change.
- **Reliability, Availability, Maintainability:** It is very man or woman friendly, software is at ease and there isn't always plenty protection. Project can be upgraded as regular with the requirement grade by grade.
- **Configuration and Compatibility**: Describes necessities together with the ones connected with character customization or operations in specific competing environments.
- **Usability:** Describes objects with a view to make sure the purchaser friendliness of the software software.

Feasibility System

The plausibility of the task is examined on this segment and business endeavor idea is advanced with a totally spic and span plan for the errand and a couple of expense gauges.

During machine examination the common sense look at of the proposed device is to be cultivated. This is to ensure that the proposed contraption isn't by and large a load to the affiliation. For credibility evaluation, some capacity of the most necessities for the system is significant. Three key contemplations engaged with the practicality examination are:

- Economical feasibility
- Technical feasibility
- Operational feasibility

Technical Feasibility

This view is done to check the specialized possibility, this is, the specialized necessities of the instrument. Any gadget better have than at this point don't have an over the top call for at the to be had specialized resources. This could accomplish outrageous necessities on the available particular resources. This will provoke extremist longings being arranged on the client. The made system should have a modest essential, as handiest least or invalid changes are required for completing this device.

Economical Feasibility

This investigate is finished to check the monetary effect that the framework may likewise have at the company. The measure of asset that the undertaking can fill the investigations and improvement of the device is limited. The charges should be supported. Therefore, the high-leveldevice as appropriately inside the charge assortment and this have gotten finished because of the truth limit of the innovation utilized are unreservedly to be had. Just the hand crafted items mustbe advertised.

Operational Feasibility

The part of view is to test the level of pervasiveness of the device through the customer. This fuses the strategy of setting up the buyer to use the machine capably. The sponsor need to as of now don't feel traded off with the guide of strategy for the contraption, rather ought to acknowledge transport of it as a need. The level of reputation through the customers thoroughly relies upon the systems which are utilized to show the individual buyer to use the machine capably. The sponsor need to as of now don't feel traded off with the guide of strategy for the contraption the structure and to make him acquainted with it. His period of self-acumen ought to be raised with the objective that he is furthermore prepared to ensure examination, this is welcomed, as he's the last purchaser of the contraption.

2.2 Hardware Requirements:

- Processor : Intel Core i3 and above processor
- RAM : 4/8 GB
- Hard Drive : 50 GB

2.3 Software Requirements:

- Operating system : Windows 7 or higher
- Server-side Script : Python 3.7
- IDE : PyCharm
- Programming Language : Python

2.4 Interface

The system utilizes the Flask web framework to establish a robust backend. Flask serves as the foundation for handling HTTP requests, routing, and rendering dynamic web pages. The detailed specification includes the version of Flask used, any additional extensions integrated, and the configuration settings essential for the proper functioning of the application.

2.5 Summary

In the "Interfaces" section of the SRS chapter, the focus is on the essential components driving the system's functionality. The Flask web framework is employed for establishing a robust backend, handling HTTP requests, routing, and dynamically rendering web pages. The specification includes precise details on the Flask version, integrated extensions, and configuration settings crucial for the system's optimal performance.

CHAPTER 3

HIGH LEVEL DESIGN

High-level design (HLD) explains the architecture that would be used for developing a software product. The architecture diagram provides an overview of an entire system, identifying the main components that would be developed for the product and their interfaces. The HLD uses possibly non-technical to mildly technical terms that should be understandable to the administrators of the system. In contrast low level design further exposes the logical detailed design of each of these elements for programmers.

High level design is the design which is used to design the software related requirements. In this chapter complete system design is generated and shows how the modules, sub modules and the flow of the data between them are done and integrated. It is very simple phase that shows the implementation process. The errors done here will be modified in the coming processes.

3.1 Design Consideration

In the context of a local host deployment without the need for login registration, the design considerations are meticulously crafted to prioritize simplicity and accessibility. The user interface is designed to be intuitive, offering streamlined interactions for healthcare professionals within the local environment. Emphasis is placed on clarity and efficiency, eliminating the complexities associated with user accounts while ensuring a user-friendly experience. Data security remains a paramount consideration, with encryption protocols safeguarding local data storage and transmission. The design optimizes local database integration, prioritizing swift data processing to meet the specific needs of a local host environment.

The design further addresses updates and maintenance procedures, prioritizing user-friendliness for seamless local deployment of new versions. Compatibility with commonly used local browsers is ensured, offering a consistent experience for users. Transparency in machine learning model outputs is maintained, allowing healthcare professionals to interpret predictions without the necessity of individualized user profiles. Overall, these design considerations collectively contribute to the development of a robust, secure, and locally optimized solution for fetal health classification and baby weight prediction within a local host deployment framework. Allowing healthcare professionals to interpret predictions without the necessity of individualized user profiles.

3.2 System Architecture of Fetal Health Prediction

- Gathering requirements is the main attraction of the Analysis Phase. The process of gathering requirements is usually more than simply asking the users what they need and writing their answers down. The process of gathering requirements is usually more than simply asking the users what they need and writing their answers down.
- Depending on the complexity of the application, the process for gathering requirements has a clearly defined process of its own. This process consists of a group of repeatable processes that utilize certain techniques to capture, document, communicate, and manage requirements. This process consists of a group of repeatable processes.
- Systems design is the process of defining the architecture, components, modules, interfaces, and data for a system to satisfy specified requirements. Systems design could see it as the application of systems theory to product development. There is some overlap with the disciplines of systems analysis, systems architecture and systems engineering.
- If the broader topic of product development "blends the perspective of marketing, design, and manufacturing into a single approach to product development," then design is the act of taking the marketing information and creating the design of the product to be manufactured. Systems design is therefore the process of defining and developing systems to satisfy specified requirements of the user.
- Until the 1990s systems design had a crucial and respected role in the data processing industry. In the 1990s standardization of hardware and software resulted in the ability to build modular systems. The increasing importance of software running on generic platforms has enhanced the discipline of software engineering.
- Object-oriented analysis and design methods are becoming the most widely used methods for computer systems design. The UML has become the standard language in object-oriented analysis and design. It is widely used for modelling software systems and is increasingly used for high designing non- software systems and organizations.
- System design is one of the most important phases of software development process. The purpose of the design is to plan the solution of a problem specified by the requirement documentation.

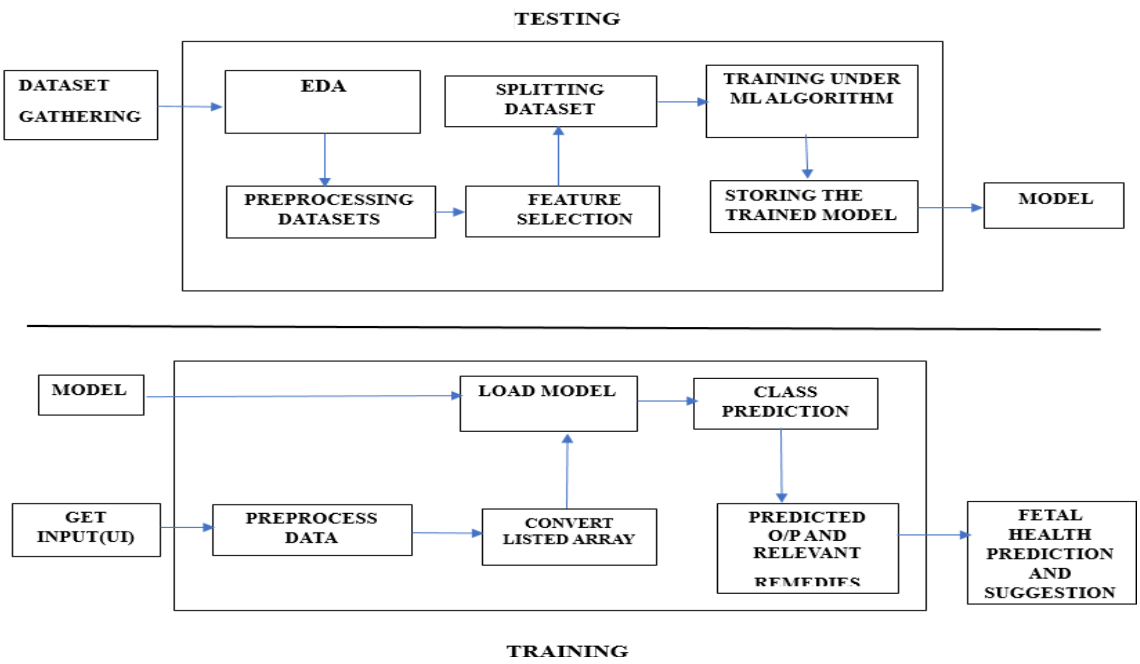


Figure 3.1: Architecture Diagram of Fetal Health Prediction

The above figure 3.1 Architecture Diagram of Fetal Health Prediction refers to design of the system is perhaps the most critical factor affecting the quality of the software. The objective of the design phase is to produce overall design of the software. It aims to figure out the modules that should be in the system to fulfil all the system requirements in an efficient manner. The design will contain the specification of all these modules, their interaction with other modules and the desired output from each module. The output of the design process is a description of the software architecture.

3.3 Specification Using Use Case Diagram

A Use Case consists of use cases, persons, or various things that are invoking the features called as actors and the elements that are responsible for implementing the use cases. Use case diagrams capture the dynamic behavior of a live system. It models how an external entity interacts with the system to make it work. Use case diagrams are responsible for visualizing the external things that interact with the part of the system. A use case diagram at its simplest is a representation of a user's interaction with the system that shows the relationship between the user and the different use cases in which the user is involved. Use case diagrams are responsible for visualizing the external things that interact with the part of the system.

A use case is a unique functionality of a system which is accomplished by a user. A purpose of use case diagram is to capture core functionalities of a system and visualize the interactions of various things called as actors with the use case. This is the general use of a use case diagram.

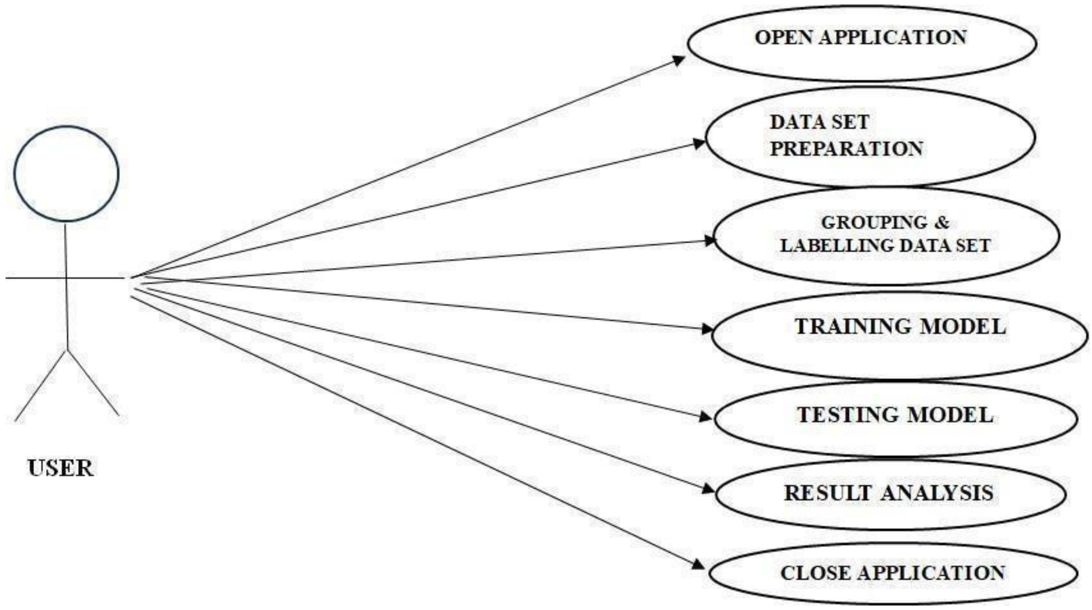


Figure 3.2: Use case Diagram of Fetal Health Prediction

The above figure 3.2 Use case Diagram of Fetal Health Prediction refers to the use case diagram is a visual representation of the functional requirements of a system from the perspective of its users. It consists of actors, representing entities external to the system that interact with it, and use cases, representing the specific functionalities or tasks the system performs to fulfill the needs of its users. In a basic use case diagram, actors are depicted as stick figures, while use cases are represented as ovals. Relationships between actors and use cases are illustrated through lines connecting them.

3.4 Module Specification of Fetal Health Prediction

Use case Diagram for each module

1) **Dataset Gathering Module**

   Gathering a dataset involves a systematic process of collecting and compiling relevant information or data points for analysis, research, or machine learning purposes. Initially, it's crucial to clearly define the objective of the dataset, outlining the problem to be solved or the questions to be answered. Identifying suitable data sources follows, which may include publicly available datasets, online databases, APIs, web scraping, surveys, or manual data entry.

2) Preprocessing Module

Preprocessing of Fetal Health Prediction refers preprocessing module is a fundamental component within a data processing pipeline, designed to refine raw data before it's utilized for analysis or machine learning tasks. This module encompasses several crucial steps to ensure the data is cleaned, transformed, and optimized for subsequent modeling. Initially, data cleaning procedures are employed to handle missing values, eliminate duplicates, and rectify inconsistencies or errors within the dataset.

3) Feature Selection Module

Feature selection module is a crucial component within the data preprocessing pipeline, dedicated to identifying the most relevant and informative features from the dataset. This module is essential for optimizing model performance, reducing overfitting, and improving interpretability by selecting a subset of features that contribute most significantly to the predictive task at hand. improving interpretability by selecting a subset of features that contribute most significantly to the predictive task at hand.

4) Model Training Module

Model training module is a pivotal component within the machine learning pipeline, dedicated to training predictive models on labeled data to learn patterns and relationships inherent in the dataset. This module involves several key steps, beginning with the selection of an appropriate algorithm or model architecture based on the nature of the problem, the characteristics of the data, and the desired outcome. model architecture based on the nature of the problem, the characteristics of the data, and the desired outcome.

5) Evaluating Model Module

Evaluating model module serves as a pivotal phase in the machine learning pipeline, tasked with assessing the performance and efficacy of trained models on unseen data. Central to this process is the utilization of a separate dataset, commonly known as the test set, which remains distinct from the data used during model training. This segregation ensures an unbiased evaluation of the model's capacity to generalize to novel instances. This segregation ensures an unbiased evaluation of the model's capacity to generalize to novel instances.

3.5 Data Flow Diagram of Fetal Health Prediction

- **Dataset Gathering Module**

The Dataset Gathering Module serves as a foundational component within data science and machine learning systems, tasked with the crucial responsibility of acquiring, consolidating, and structuring data from diverse sources. This module operates by integrating with various data repositories, including databases, APIs, and web scraping tools, to retrieve pertinent datasets. Upon retrieval, it undertakes essential data preprocessing steps such as cleansing, normalization, and format conversion to ensure data quality and consistency. The collected data is then stored in organized repositories, typically databases or file systems, facilitating easy access and subsequent analysis

- **Preprocessing Module**

The Preprocessing Module is a vital component within data analysis and machine learning workflows, tasked with transforming raw data into a format suitable for further analysis and modeling. This module encompasses a range of essential tasks, including data cleaning to handle missing values and remove duplicates, data transformation to standardize numerical features and encode categorical variables, and feature engineering to derive new features or select relevant ones. It also addresses specific challenges such as text preprocessing for natural language data, handling imbalanced datasets, and managing outliers through techniques like oversampling or outlier removal

- **Feature Selection Module**

Feature selection of fetal health prediction refers Feature Selection Module is a component within a larger system designed to facilitate the selection of relevant features for data analysis, machine learning, or other computational tasks. This module typically serves as a tool to enhance the performance and efficiency of models. the selection of relevant features for data analysis, machine learning, or other computational tasks. This module typically serves as a tool to enhance the performance and efficiency of models.

- **Model Training Module**

The Model Training Module is a pivotal component within machine learning systems, responsible for training and optimizing predictive models based on the preprocessed data. This module encompasses a series of steps aimed at iteratively improving model performance. It begins with the selection of appropriate algorithms or model architectures tailored to the specific task and dataset characteristics. Subsequently, the module involves splitting the preprocessed data.

- **Evaluating Model Module**

The Evaluating Model Module plays a critical role in assessing the performance and effectiveness of machine learning models before deployment. This module involves a series of steps aimed at rigorously evaluating the trained models to ensure their reliability and generalization capabilities. Initially, the module employs various evaluation metrics tailored to the specific problem domain, such as accuracy, precision, recall, F1 score, or area under the ROC curve (AUC-ROC), depending on whether the task is classification, regression, or another type of predictive modeling. These metrics provide quantitative measures of how well the model performs on unseen data.

3.6 Data flow Diagram of Fetal Health Prediction

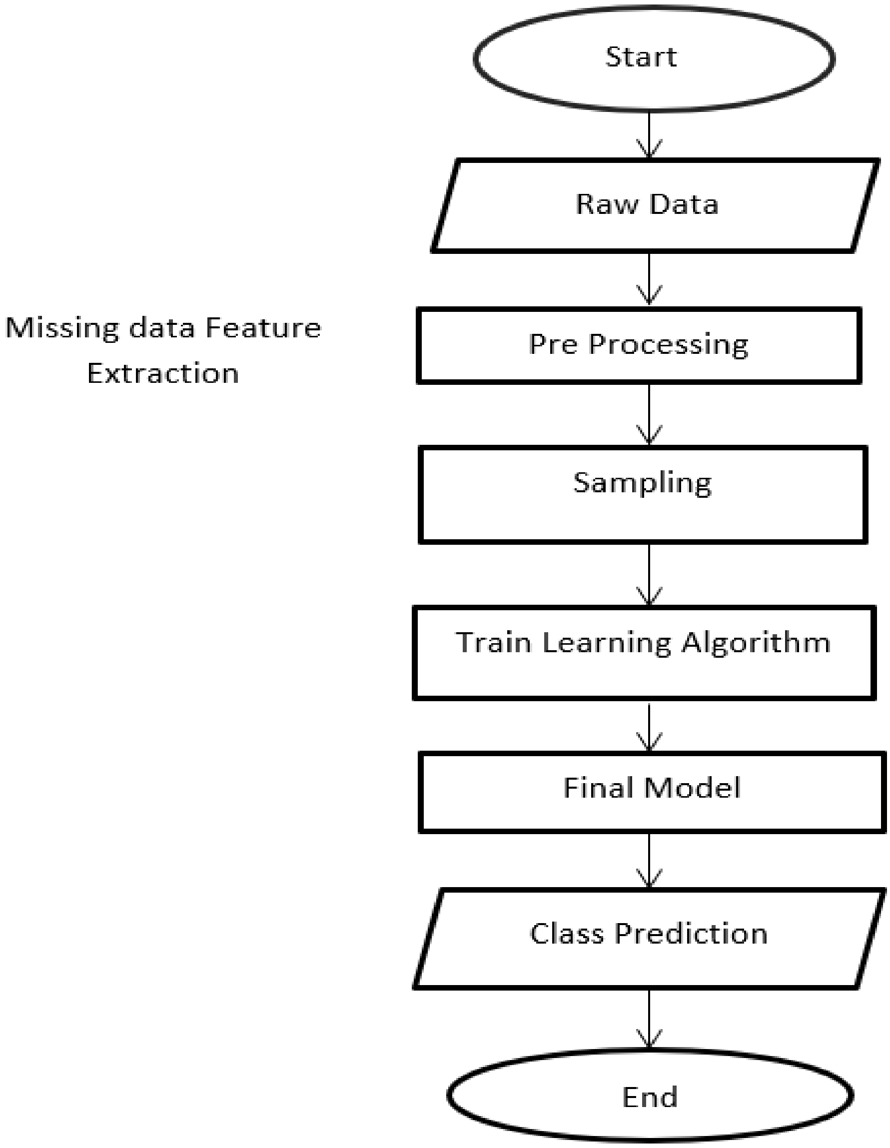


Figure 3.3: Dataflow Diagram of Fetal Health Prediction

The above figure 3.6 Dataflow diagram of Fetal Health Prediction refers dataflow diagram (DFD) for a fetal health prediction system outlines the flow of data through various components of the system. At the core of this system are external entities, including a medical data source, which provides pertinent information such as maternal age, fetal heart rate, and uterine contractions.

The process begins with data collection, where the system gathers medical data from the external source. Subsequently, the collected data undergoes preprocessing, which involves cleaning, transforming, and preparing it for analysis. Once preprocessed, the data is utilized in the model training process, where machine learning algorithms are employed to train models for fetal health prediction.

3.7 State Chart Diagram of Fetal Health Prediction

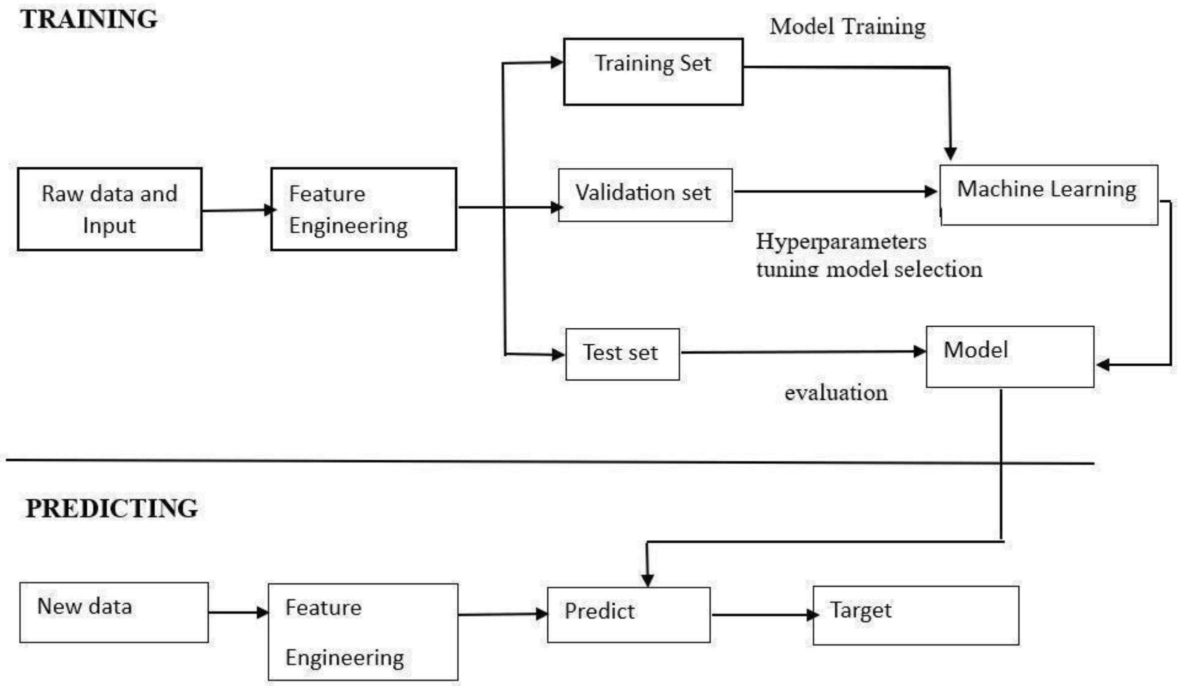


Figure 3.4: State chart diagram of Fetal Health Prediction

The above figure 3.7 State chart diagram of Fetal Health Prediction refers state chart diagram illustrates the dynamic behavior of a system by representing its various states and the transitions between them. In this diagram, each state signifies a specific phase or condition within the system, and transitions denote the events or triggers that prompt the system to move from one state to another. The diagram provides a visual representation of how the system responds to external stimuli or internal events, capturing the sequence of states the system undergoes during its lifecycle. State chart diagrams are valuable for understanding and designing the temporal aspects of a system.

3.8 Summary

This project centers on the development of a comprehensive prenatal care system utilizing advanced machine learning techniques. The primary objectives include fetal health classification and baby weight prediction, vital aspects in proactive maternal healthcare. The project unfolds in modular stages, commencing with dataset gathering, progressing through preprocessing, feature selection, and model training, and concluding with model evaluation. Leveraging Flask, HTML, CSS, and JavaScript for a local host deployment without login registration, the system aims to provide a user-friendly interface for healthcare professionals

CHAPTER 4

DETAILED DESIGN

A detail design is the process of each individual module which is completed in the earlier stage than implementation. It is the second phase of the project first is to design phase and second phase is individual design of each phase options. It saves more time and another plus point is to make implementation easier.

4.1 Structural chart of the proposed system

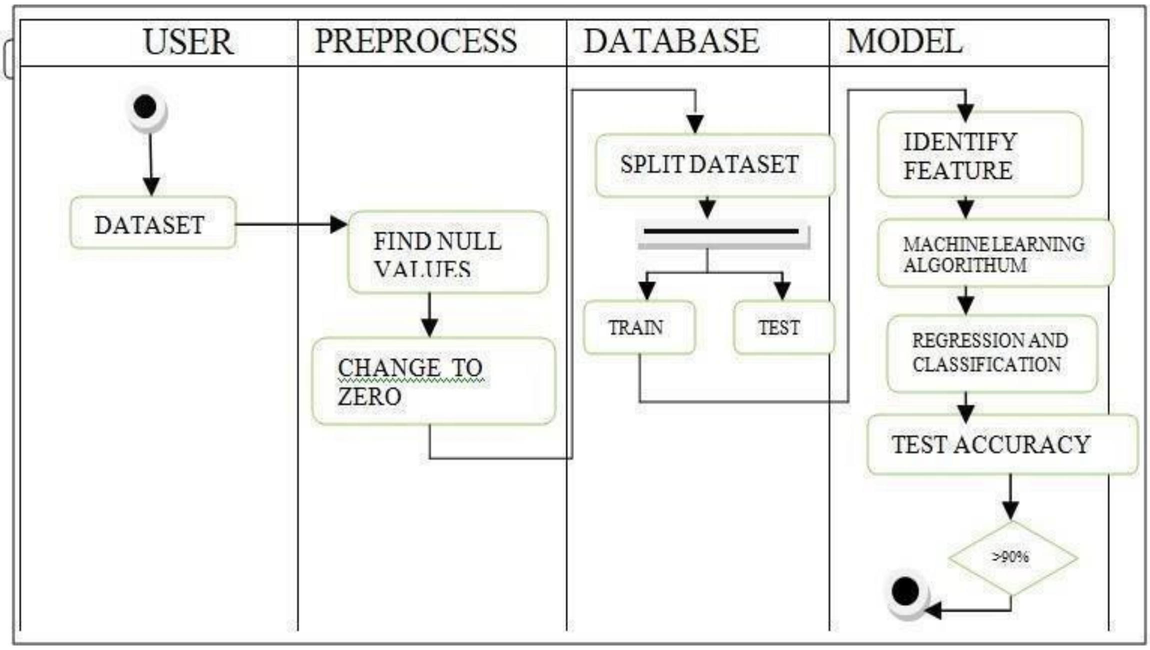


Figure 4.1: Structure chart of Fetal Health Prediction

The above figure 4.1 Structure chart of fetal health prediction refers Predicting fetal health involves a structured process encompassing data collection, preprocessing, feature extraction, model selection, training, evaluation, optimization, deployment, and ongoing monitoring. Initially, relevant data, including maternal health records, ultrasound images, and genetic information, is gathered. Subsequently, this data undergoes preprocessing to handle noise, missing values, and standardization. Feature extraction techniques are then employed to derive meaningful insights from the processed data. Following this, suitable machine learning or statistical models are selected and trained using labeled data.

Model performance is evaluated using various metrics, and optimization techniques are applied to enhance accuracy. Once the model is deemed satisfactory, it is deployed into real-world settings, such as clinical environments, with mechanisms in place for continuous monitoring and maintenance.

4.2 Detail description of each module

This section gives the detailed description of each module which includes pre-processing techniques, Noise removal, Missing Value Handling, Normalization, Dimensionality Reduction, Signal Processing.

4.2.1 Dataset Gathering Module

- **Functionality:** The primary functionality of the Dataset Gathering module is to acquire essential data required for fetal health classification and baby weight prediction. It involves sourcing data from various maternal and fetal parameters, such as age, weight gain, blood pressure, and ultrasound measurements. The module ensures a diverse and comprehensive dataset that serves as the foundation for subsequent analysis.
- **Processing**: The processing within this module includes initiating data acquisition processes, which may involve interfacing with databases or external sources. It details the systematic collection of relevant parameters, ensuring data integrity, and storing the acquired dataset for subsequent preprocessing.

4.2.2 Preprocessing Module

- **Functionality:** The Preprocessing module is designed to enhance the quality and readiness of the acquired dataset for machine learning. It includes steps such as handling missing values, addressing outliers, and normalizing data to a standardized scale. The module's functionality is crucial for ensuring the reliability and consistency of the dataset.
- **Processing**: Processing within this module involves executing algorithms and methods to clean and transform the dataset. It encompasses techniques for identifying and handling missing or irregular data points, as well as normalizing numerical features to ensure uniformity across the dataset.

4.2.3 Feature Selection Module

- **Functionality:** The Feature Selection module aims to identify and retain the most relevant features for subsequent model training. It involves exploring different techniques, such as filtering, wrapper methods, or embedded methods, to ensure that the model is trained on discriminative and impactful features.

- **Processing:** The processing steps include the application of feature selection algorithms and methods to the preprocessed dataset. It details the criteria for selecting features, reducing dimensionality if necessary, and creating a refined feature set that contributes meaningfully to the model's predictive capabilities.

4.2.4 Model Training Module

- **Functionality:** The Model Training module is tasked with selecting an appropriate machine learning algorithm and training the model on the preprocessed dataset. It encompasses the functionality of developing a predictive model capable of understanding patterns within the data.
- **Processing:** Processing in this module involves the selection of suitable algorithms based on the dataset characteristics and desired outcomes. It includes the training phase, during which the model learns from the preprocessed dataset, adjusting its parameters to optimize predictions for future instances. the model learns from the preprocessed dataset, adjusting its parameters

4.2.5 Evaluating Model Module

- **Functionality**: The Evaluating Model module assesses the performance of the trained model by employing various metrics such as accuracy, precision, recall, and F1 score. It gauges the model's effectiveness in making predictions on new data. It gauges the model's effectiveness in making predictions on new data.
- **Processing:** Processing within this module involves executing the evaluation metrics on the model's predictions. It details the testing phase, during which the model's performance is measured against ground truth data, providing insights into its accuracy and reliability. It details the testing phase, during which the model's performance is measured.

4.3 Board Work

Support Vector Machine or SVM is one of the most popular Supervised Learning algorithms, which is used for Classification as well as Regression. The goal of the SVM algorithm is to create the best line.

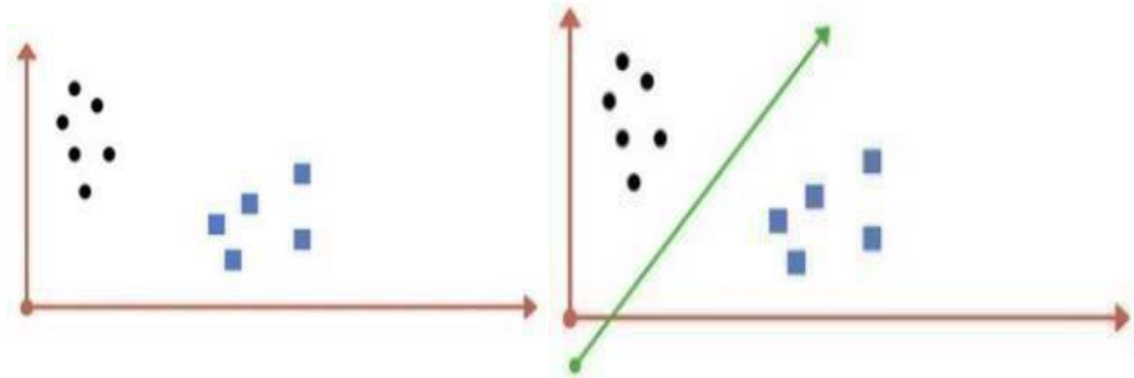


Figure 4.2: Scattering points and Hyperplane

4.3.1 Considering the dataset of Fetal Health

Table 4.1: Dataset of Fetal Health

SN	Acceleration	Fetal movement	Fetal health
1	5	2	N
2	11	1	N
3	6	8	A
4	6	3	N
5	13	4	N
6	2	3	A
7	3	4	A
8	5	9	A
9	5	0	S

The above Table 4.1 is the Dataset of Fetal Health Database, offering recordings alongside corresponding fetal outcomes, whether normal, suspect, or pathologic.

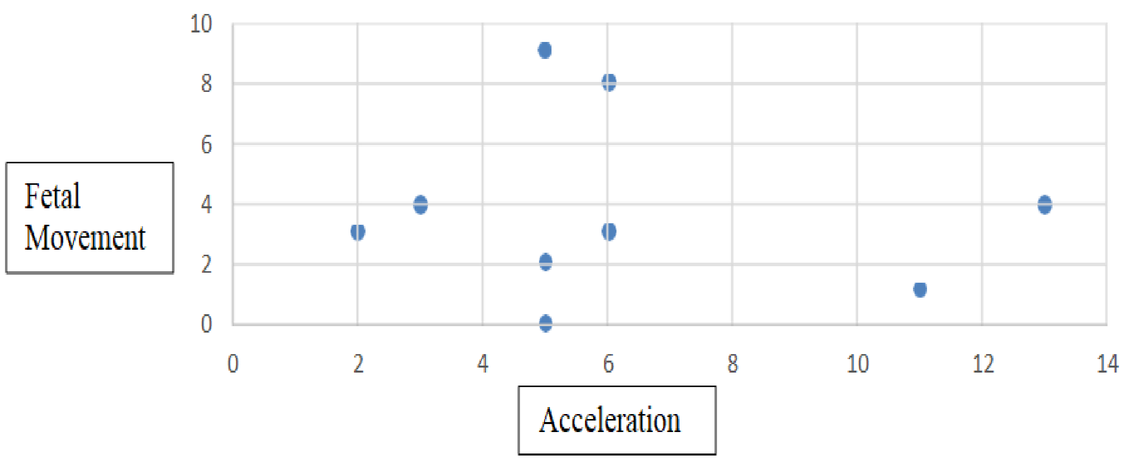


Figure 4.3: Plotting Point

The above figure 4.3: Plotting Point refers to marking the points from the datasets of fetal health prediction, the points are marked by considering two main attributes they arefetal movement and acceleration.

4.3.2 To calculate weight and bias

$$\begin{aligned}
\alpha_1\tilde{s_1}\cdot\tilde{s_1}+\alpha_2\tilde{s_2}\cdot\tilde{s_1}+\alpha_3\tilde{s_3}\cdot\tilde{s_1} &= -1\\
\alpha_1\tilde{s_1}\cdot\tilde{s_2}+\alpha_2\tilde{s_2}\cdot\tilde{s_2}+\alpha_3\tilde{s_3}\cdot\tilde{s_2} &= +1\\
\alpha_1\tilde{s_1}\cdot\tilde{s_3}+\alpha_2\tilde{s_2}\cdot\tilde{s_3}+\alpha_3\tilde{s_3}\cdot\tilde{s_3} &= +1
\end{aligned}$$

$$\tilde{w} = \sum_i \alpha_i \tilde{s}_i$$

$$\alpha_1 \begin{bmatrix} 26 \end{bmatrix} + \alpha_2 \begin{bmatrix} 51 \end{bmatrix} + \alpha_3 \begin{bmatrix} 35 \end{bmatrix} = -1$$

$$\alpha_1 \begin{bmatrix} 51 \end{bmatrix} + \alpha_2 \begin{bmatrix} 101 \end{bmatrix} + \alpha_3 \begin{bmatrix} 69 \end{bmatrix} = 1$$

$$\alpha_1 \begin{bmatrix} 35 \end{bmatrix} + \alpha_2 \begin{bmatrix} 69 \end{bmatrix} + \alpha_3 \begin{bmatrix} 53 \end{bmatrix} = 1$$

$$\bar{S1} = \begin{bmatrix} 3 \\ 4 \\ 1 \end{bmatrix} + \quad \bar{S2} = \begin{bmatrix} 6 \\ 8 \\ 1 \end{bmatrix} \quad \bar{S3} = \begin{bmatrix} 6 \\ 4 \\ 1 \end{bmatrix}$$

$$\alpha_1 \begin{bmatrix} 3 \\ 4 \\ 1 \end{bmatrix} \begin{bmatrix} 3 \\ 4 \\ 1 \end{bmatrix} + \alpha_2 \begin{bmatrix} 6 \\ 8 \\ 1 \end{bmatrix} \begin{bmatrix} 3 \\ 4 \\ 1 \end{bmatrix} + \alpha_3 \begin{bmatrix} 6 \\ 4 \\ 1 \end{bmatrix} \begin{bmatrix} 3 \\ 4 \\ 1 \end{bmatrix} = -1$$

$$\alpha_1 \begin{bmatrix} 3 \\ 4 \\ 1 \end{bmatrix} \begin{bmatrix} 6 \\ 8 \\ 1 \end{bmatrix} + \alpha_2 \begin{bmatrix} 6 \\ 8 \\ 1 \end{bmatrix} \begin{bmatrix} 6 \\ 8 \\ 1 \end{bmatrix} + \alpha_3 \begin{bmatrix} 6 \\ 4 \\ 1 \end{bmatrix} \begin{bmatrix} 6 \\ 8 \\ 1 \end{bmatrix} = 1$$

$$\alpha_1 \begin{bmatrix} 3 \\ 4 \\ 1 \end{bmatrix} \begin{bmatrix} 6 \\ 4 \\ 1 \end{bmatrix} + \alpha_2 \begin{bmatrix} 6 \\ 8 \\ 1 \end{bmatrix} \begin{bmatrix} 6 \\ 4 \\ 1 \end{bmatrix} + \alpha_3 \begin{bmatrix} 6 \\ 4 \\ 1 \end{bmatrix} \begin{bmatrix} 6 \\ 4 \\ 1 \end{bmatrix} = 1$$

w1=-49.76
w2=45
b=2.42

Weights: [-49.76,45]
Bias : 2.42
W . x – b = 0
$[w1, w2] . [x, y]$
$w1x + w2y - b = 0$

$x = \frac{b - w2y}{w1}$
put y=0,
$x = \frac{b}{w1}$
$x = \frac{2.42}{-49.76}$
x= -0.04

$y = \frac{b - w_1 x}{w2}$
put x=0,
$y = \frac{b}{w2}$
$y = \frac{2.42}{45}$
y= 0.05

Point 1= (-0.04 ,0)
Point 2= (0 , 0.05)

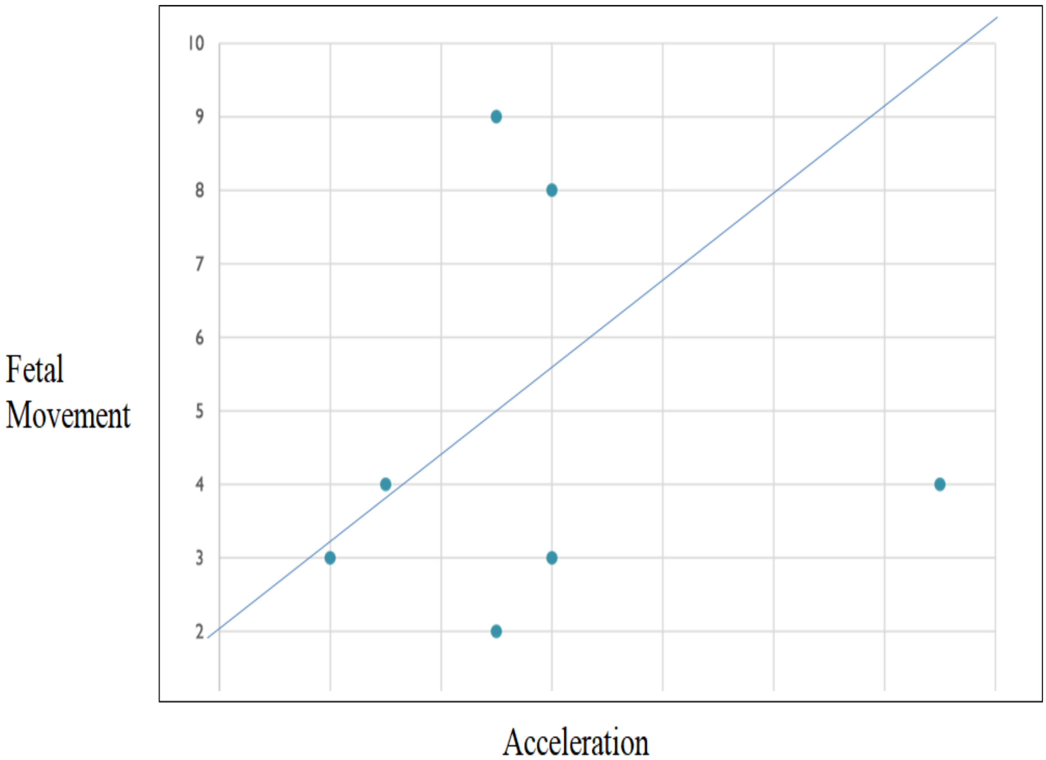


Figure 4.4: Plotting Hyper Plane

The above figure 4.4: Plotting Hyper plane refers plotting of the hyper plane by considering the datasets of fetal health prediction and the main attribute for drawing the hyper plane is acceleration and fetal movement

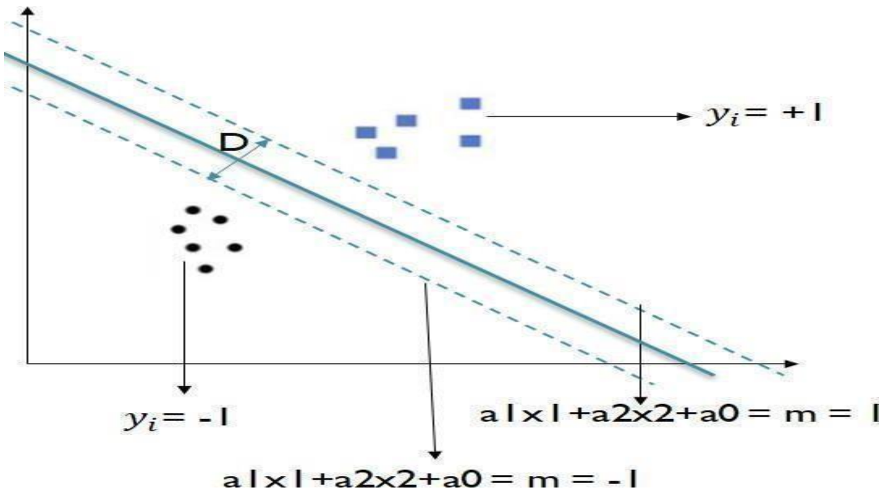


Figure 4.5: Example of Plotting Upper Bound and Lower Bound

The above figure 4.5 Example of plotting Upper Bound and Lower Bound refers plotting Upper Bound and Lower Bound in SVM is often associated with the margin, which is the distance between the decision boundary and the closest data points (support vectors). SVM aims to find the decision boundary that maximizes this margin, effectively creating a separation between different classes.

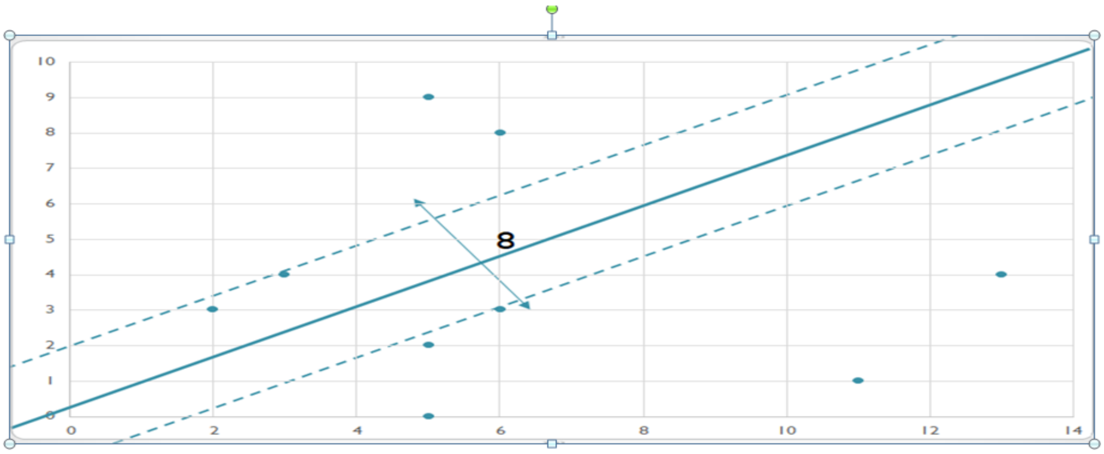


Figure 4.6: Plotting Upper Bound and Lower

The above figure 4.6 Plotting Upper Bound and Lower Bound refers a wider margin is crucial for reducing overfitting and enhancing the model's ability to generalize. Hence, in fetal health prediction SVM models, careful selection and tuning of the upper bound and lower bound (margin) parameters are pivotal for achieving a balance between accurate classification and robust generalizability to new cases.

a1x1+a2x2+a0 = -1 a1x1+a2x2+a0=1

a1(3)+a2(4)+0 = -1

a1(6)+a2(3)+0 = 1

3a1+4a2 = -1

6a1+3a2 = 1

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

$=\sqrt{(-3 - 4)^2 + (6 - 3)^2}$

$=\sqrt{49 + 9}$

=8

4.3.3 Example to Check for normal condition

Consider the point (5,2)

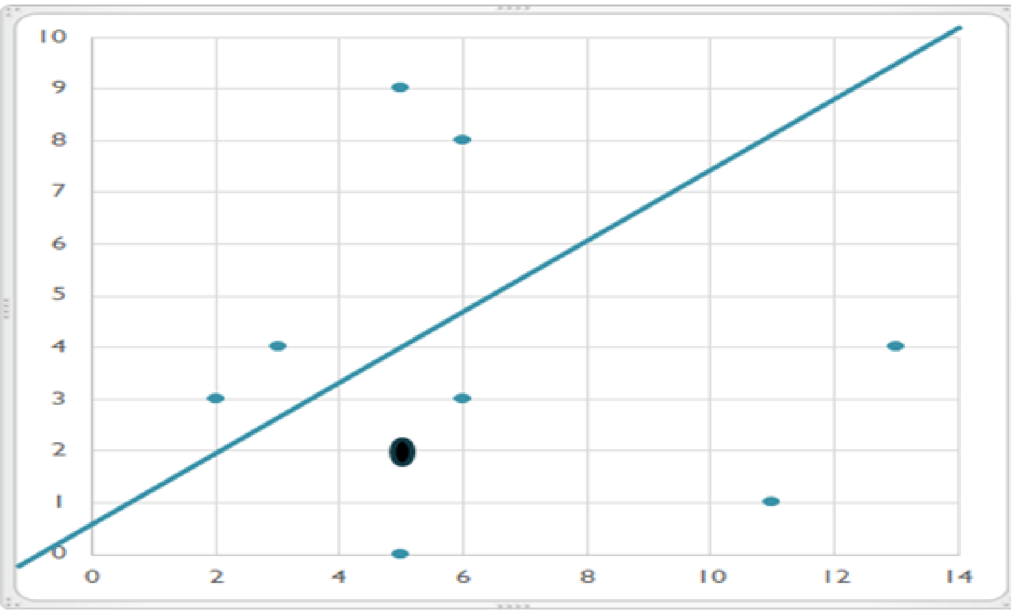


Figure 4.7: Normal Condition

The above figure 4.7 Normal Condition refers the term "normal class label" to the designation given to instances or cases where the fetal health status is considered within normal or healthy parameters. This class label is essential for training machine learning models to distinguish between normal and abnormal fetal health conditions based on available data. This typically involves reviewing medical records, diagnostic tests, and expert assessment to identify cases where fetal health indicators fall within expected ranges.

$w1x + w2y - b < 0$

= - 49.76(5) + 45(2)-2.42

= -161.22

-161.22 < 0

Hence, condition is **Normal**

4.3.4 Example to check for abnormal condition

Consider the point (6,8)

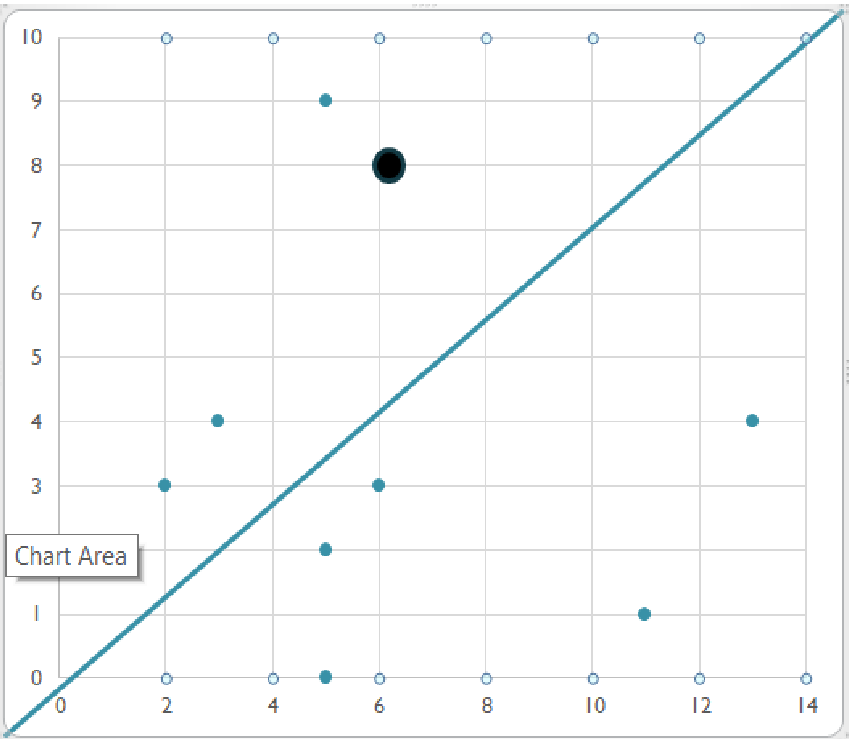


Figure 4.8: Abnormal Condition

The above figure 4.8 Abnormal Condition refers the abnormal class in fetal health prediction represents instances where fetal health status shows deviations from normal parameters, serving as an essential component for training machine learning models to accurately identify and classify various fetal health complications or abnormalities.

$w1x + w2y - b > 0$
= -49.76(6)+45(8)-2.42
= 59.02
59.02> 0
Hence, condition is Abnormal

4.3.5 Example to check for Suspect Condition

Consider the point (5,0)

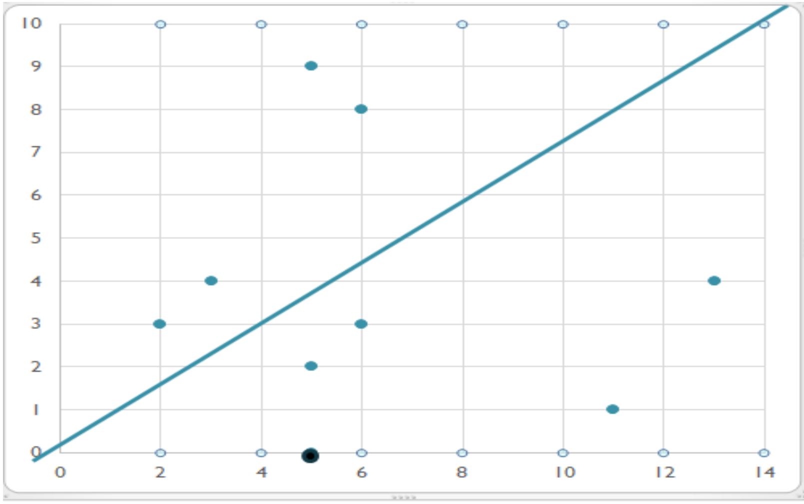


Figure 4.9: Suspect Condition

The above figure 4.9: Suspect Condition refers the "suspect class label" in fetal health prediction represents instances where there are indications of potential risk or uncertainty regarding fetal health status, serving as a valuable category for identifying borderline cases that may require additional attention or investigation.

$w1x + w2y - b = 0$
= 49.76(5)+45(0)-2.42
= 0.25
0.25= 0
Hence, condition is Suspect

4.4 Summary

This chapter intricately outlines the architectural and functional aspects of the prenatal care system, emphasizing a modular approach for fetal health classification and baby weight prediction. The chapter commences with a thorough exposition of the Dataset Gathering module, elucidating its functionality in acquiring diverse maternal and fetal parameters crucial for subsequent analysis. The Preprocessing module follows suit, detailing its role in refining the dataset through meticulous cleaning and normalization processes, ensuring data quality.

CHAPTER 5

IMPLEMENTATION

Implementation is the process of converting a new system design into an operational one. It is the key stage in achieving a successful new system. It must therefore be carefully planned and controlled. The implementation of a system is done after the development effort is completed.

5.1 Implementation requirements

5.1.1 Programming language selection

The choice of programming language plays a pivotal role in the successful implementation of the prenatal care system. After careful consideration of various factors, Python has been selected as the primary programming language for this project. Python's versatility and extensive ecosystem make it an ideal choice for machine learning applications. Its rich libraries, particularly in the form of NumPy, Pandas, and Scikit-learn, provide robust tools for data manipulation, preprocessing, and model development. Python's readability and simplicity enhance code maintainability, crucial for a project with multiple modules and iterative development.

The integration of Flask, a Python web framework, aligns seamlessly with the local host deployment requirements of the project. Flask enables the creation of a lightweight and efficient web application to present the machine learning models to healthcare professionals. The use of HTML, CSS, and JavaScript for the frontend further complements the Python backend, offering a well-rounded and interactive user interface. This programming language synergy ensures a cohesive and integrated implementation, fostering a smooth flow between data processing, model training, and user interaction.

The selection of Python is also strategic in the context of the extensive support and community engagement it enjoys in the field of machine learning and data science. The abundance of online resources, tutorials, and a vibrant community facilitates troubleshooting, accelerates development cycles, and ensures that the project remains at the forefront of technological advancements in the rapidly evolving domain of healthcare technology. Overall, the choice of Python aligns with the project's goals of efficiency, scalability, and accessibility in the implementation phase.

5.1.2 Key features of programming language selected

Python, the selected programming language for this project, brings forth a multitude of key features that align seamlessly with the complex requirements of implementing a robust prenatal care system. One of Python's standout features is its readability and simplicity, making it an exceptionally expressive language. This characteristic is invaluable in the context of a project with multiple modules, intricate data processing, and the implementation of machine learning algorithms. The clean and concise syntax of Python promotes code clarity, easing the development process and enhancing overall project maintainability.

Another crucial feature is Python's extensive ecosystem of libraries and frameworks tailored for machine learning and data science applications. Libraries such as NumPy and Pandas offer powerful tools for efficient data manipulation and preprocessing, essential in handling diverse maternal and fetal health parameters. Scikit-learn, a prominent machine learning library, provides a wide array of algorithms and functionalities, streamlining the development of predictive models for fetal health classification and baby weight prediction. This rich ecosystem significantly expedites development cycles, enabling the project to leverage pre-existing tools and focus on the unique challenges posed by prenatal care.

Python's versatility extends to its compatibility with Flask, a lightweight and modular web framework chosen for the local host deployment of the project. This integration empowers the system with a user-friendly interface, seamlessly combining the strengths of Python for backend processing with HTML, CSS, and JavaScript for frontend interactions. Python's adaptability across different domains, from data manipulation to web development, positions it as an ideal choice for a project requiring a cohesive integration of machine learning capabilities within a user-accessible interface. In summary, Python's readability, extensive libraries, and versatility make it the cornerstone for the successful implementation of the prenatal care system.

5.1.3 Description about tools, GUI used

The project leverages a carefully selected set of tools and a Graphical User Interface (GUI) to enhance the efficiency and accessibility of the prenatal care system. The Flask web framework serves as the backbone for the local host deployment, providing a robust foundation for handling HTTP requests, routing, and rendering dynamic web pages. Flask's lightweight nature aligns with the project's requirements, facilitating a responsive and scalable user interface. Additionally, Flask integrates seamlessly with Python, the chosen programming language, fostering a cohesive environment for backend development.

The GUI is crafted using a combination of HTML, CSS, and JavaScript, ensuring an interactive and visually appealing user experience. HTML defines the structure of the web pages, CSS governs their presentation and styling, and JavaScript adds dynamic behavior to enhance user interactions. This trifecta of frontend technologies facilitates the creation of an intuitive and user-friendly interface for healthcare professionals interacting with the prenatal care system. The modular design of the GUI aligns with the project's overall approach, allowing for a clear separation of concerns between the frontend and backend components.

Furthermore, the choice of tools extends to the integration of libraries such as NumPy, Pandas, and Scikit-learn within the Python ecosystem. NumPy and Pandas empower efficient data manipulation and preprocessing, ensuring the dataset's readiness for machine learning analysis. Scikit-learn provides a comprehensive set of tools for model development, training, and evaluation. This toolset enhances the project's capabilities, allowing for a seamless integration of machine learning functionalities within the GUI. Collectively, the chosen tools and GUI components contribute to a well-rounded and effective prenatal care system, emphasizing user accessibility, system responsiveness, and modular design.

HTML:

1. **Document Structure**: Begin by creating the basic structure of your HTML document. Use the <!DOCTYPE html> declaration to specify the HTML version, and create the <html>, <head>, and <body> tags.

```
<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">
```

```
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Prenatal Care System</title>
</head>
<body>
</body>
</html>
```

2. **Header Section:** Populate the **<head>** section with metadata, including character set, viewport settings, and the title of the webpage. This section may also include links to external stylesheets or scripts.

```
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Prenatal Care System</title>
  <link rel="stylesheet" href="styles.css">
  <!-- Additional meta tags or script links if needed -->
</head>
```

3. **Body Content:** Populate the **<body>** section with the main content of your webpage. This may include forms, buttons, input fields, or any other elements relevant to the functionality of the prenatal care system.

```
<body>
  <header>
    <h1>Prenatal Care System</h1>
  </header>
  <main>
```

```
        <!-- our main content goes here -->
    </main>
    <footer>
        <!-- Footer content if needed -->
    </footer>
</body>
```

CSS:

1. **External Stylesheet:** Create an external CSS file (e.g., styles.css) to keep your styles separate from HTML. Link this stylesheet in the HTML document.

```
<head>
    <!-- ... -->
    <link rel="stylesheet" href="styles.css">
    <!-- ... -->
</head>
```

2. **Styling Elements:** Use CSS rules to style HTML elements. Target specific elements, classes, or IDs and apply styles such as color, font size, margin, padding, etc.

```
body {
    font-family: 'Arial', sans-serif;
    background-color: #f4f4f4;
    color: #333;
}

header {
    background-color: #007BFF;
    color: #fff;
    text-align: center;
    padding: 1rem;
}
```

```
main {
   padding: 2rem;
}
```

3. **Responsive Design:** Implement responsive design using media queries to adjust styles based on the device's screen size.

```
@media only screen and (max-width: 600px) {
   /* Styles for small screens */
   body {
      font-size: 14px;
   }
}
```

JavaScript:

1. **Script Tag:** Include a **<script>** tag in the HTML document to link to your JavaScript file (e.g., script.js).

```
<body>
   <!-- ... -->
   <script src="script.js"></script>
</body>
```

2. **Event Handling:** Use JavaScript to handle user interactions or events. For example, use the **addEventListener** method to respond to button clicks.

```
document.getElementById('submitBtn').addEventListener('click',
function() {
   // Code to handle the submit button click
});
```

3. **DOM Manipulation:** Manipulate the Document Object Model (DOM) to dynamically update or modify HTML elements.

```
function updateContent() {
   document.getElementById('result').innerHTML = 'New Content';
}
```

4. **Asynchronous Operations:** Utilize JavaScript for asynchronous operations, such as making API calls using the **fetch** function.

```
fetch('https://api.example.com/data')
  .then(response => response.json())
  .then(data => {
    // Process the retrieved data
  })
  .catch(error => console.error('Error:', error));
```

5.2 Coding Guidelines For Programming Language Used In The Project

1. **Consistent Indentation:** Maintain a consistent indentation style throughout the code. PEP 8 recommends using 4 spaces per indentation level. Consistent indentation enhances code readability and ensures a cohesive visual structure.

```
# Good
def example_function():
    if condition:
        print("Indented block")
    else:
        print("Another indented block")
```

2. **Descriptive Variable Names:** Use descriptive and meaningful names for variables, functions, and classes. This promotes code readability and helps others understand the purpose of each element without extensive comments.

```
# Good
fetal_parameters = get_fetal_parameters()
```

3. **Modular Design:** Encourage a modular design by breaking down the code into functions and classes. Each function or class should have a clear, single responsibility. This enhances code maintainability and allows for easier testing.

```
# Good
def preprocess_data(data):
```

```
        # Implementation of data preprocessing
def train_model(features, labels):
        # Implementation of model training
```

4. **Docstrings and Comments:** Use docstrings to provide documentation for functions, classes, and modules. Additionally, use comments sparingly to explain complex or non-obvious sections of code. Follow the NumPy/SciPy documentation style for docstrings.

```
def calculate_weight(data):
        """
        Calculate baby weight based on input data.
        Parameters: -
        data (DataFrame): Input data containing relevant
parameters.
        Returns: -
        float: Predicted baby weight.
        """
        # Implementation details
        pass
```

5. **Exception Handling:** Implement proper exception handling to manage errors gracefully. Use specific exception classes when catching exceptions, and avoid using a generic **except** clause unless necessary.

```
try:
    # Code that may raise an exception
except ValueError as ve:
    # Handle specific exception
```

```
except Exception as e:
    # Handle other exceptions
```

6. **Use of Virtual Environments:** Utilize virtual environments to manage project dependencies. This ensures that the project uses specific versions of libraries, reducing compatibility issues. Common tools for virtual environments include **venv** or **conda**.

```
# Create a virtual environment
python -m venv venv
# Activate the virtual environment
source venv/bin/activate # On Unix/Linux
venv\Scripts\activate  # On Windows
```

7. **Version Control Best Practices:** Adhere to version control best practices using Git. Commit frequently with clear and concise commit messages. Follow a branching strategy, such as Gitflow, to manage feature development, releases, and bug fixes.

```
git commit -m "Add feature: Dataset Gathering"
```

8. **Testing and Test Automation:** Implement unit tests for functions and classes using a testing framework such as **pytest**. Automate the testing process to ensure that modifications to the codebase do not introduce regressions.

```
pytest tests/
```

9. **Code Reviews:** Encourage code reviews to ensure code quality and consistency. Reviews help catch errors, share knowledge, and maintain a high standard of code within the project.

```
# Code review checklist:
# - Variable naming
# - Consistent indentation
# - Documentation
```

- Test coverage

- Exception handling

These coding guidelines for Python in the project provide a comprehensive set of best practices to ensure code quality, readability, and maintainability throughout the development lifecycle. Adhering to these guidelines fosters a collaborative and efficient coding environment.

5.3 Pseudo code for each module with description

DATASET GATHERING MODULE

Description:

This module is responsible for acquiring essential data for fetal health classification and baby weight prediction. It collects data from various maternal and fetal parameters.

Pseudo Code:

```
function gather_dataset():
    # Code to connect to data sources (e.g., databases, APIs)
    data = fetch_data()
    # Code to filter relevant maternal and fetal parameters
    relevant_data = filter_data(data)
    return relevant_data
```

PREPROCESSING MODULE

Description:

The Preprocessing module enhances data quality and prepares it for machine learning. It includes steps such as handling missing values and normalizing data.

Pseudo Code:

```
function preprocess_data(raw_data):
```

```
    # Code to handle missing values
    clean_data = handle_missing_values(raw_data)
    # Code to normalize data
    normalized_data = normalize_data(clean_data)
    return normalized_data
```

FEATURE SELECTION MODULE

Description:

This module focuses on identifying and retaining the most relevant features for model training. Various feature selection techniques are explored to optimize the model's performance.

Pseudo Code:

```
function select_features(dataset):
    # Code to explore feature selection techniques (e.g., filtering, wrapper methods)
    selected_features = explore_feature_selection(dataset)
    return selected_features
```

MODEL TRAINING MODULE

Description:

The Model Training module involves selecting an appropriate machine learning algorithm and training the model on the preprocessed dataset. It creates a robust predictive model.

Pseudo Code:

```
function train_model(features, labels):
    # Code to choose a suitable machine learning algorithm
    selected_algorithm = choose_algorithm()
```

```
# Code to train the model

trained_model = train(selected_algorithm, features, labels)

return trained_model
```

EVALUATING MODEL MODULE

Description:

The Evaluating Model module assesses the performance of the trained model using various metrics. It involves testing the model on new data and analyzing its accuracy, precision, recall, and F1 score.

Pseudo Code:

Step 1: Make predictions using the trained model on the test data.

Step 2: Calculate evaluation metrics (accuracy, precision, recall, F1 score) using the predicted values and true labels.

Step 3: Return the calculated evaluation metrics (accuracy, precision, recall, F1 score) from the function.

5.4 SUMMARY

This chapter describes the implementation of the fetal health prediction using various system architectures. Implementation requirement is deliberated in Section 5.1, Section 5.2 briefs about the programming language selected. Section 5.3 describes the pseudocode for pre-processing techniques.

CHAPTER 6

SYSTEM TESTING

Testing is the process of evaluating a system or its components with the intent to find whether it satisfies the specified requirements or not. Testing is executing a system in order to identify any gaps, errors, or missing requirements in contrary to the actual requirements.

6.1 Software Test Environment

Testing Principle:

Before applying methods to design effective test cases, a software engineer must understand the basic principle that guides software testing. All the tests should be traceable to customer requirements.

Testing Methods:

There are different methods that can be used for software testing. They are,

1. **Black-Box Testing**

The technique of testing without having any knowledge of the interior workings of the application is called black-box testing. The tester is oblivious to the system architecture and does not have access to the source code. Typically, while performing a black-box test, a tester will interact with the system's user interface by providing inputs and examining outputs without knowing how and where the inputs are worked upon.

2. **White-Box Testing**

White-box testing is the detailed investigation of internal logic and structure of the code. White-box testing is also called glass testing or open-box testing. In order to perform white-box testing on an application, a tester needs to know the internal workings of the code. The tester needs to have a look inside the source code and find out which unit/chunk of the code is behaving inappropriately. The tester needs to have a look inside the source code and find out which unit/chunk of the code is behaving inappropriately. The tester needs to have a look inside the source code and find out which unit/chunk of the code is behaving inappropriately.

6.2 Test procedures

Levels of Testing:

There are different levels during the process of testing. Levels of testing include different methodologies that can be used while conducting software testing. The main levels of software testing are:

1. **Functional Testing:**

This is a type of black-box testing that is based on the specifications of the software that is to be tested. The application is tested by providing input and then the results are examined that need to conform to the functionality it was intended for. Functional testing of software is conducted on a complete, integrated system to evaluate the system's compliance with its specified requirements. There are five steps that are involved while testing an application for functionality.

- The determination of the functionality that the intended application is meant to perform.
- The creation of test data based on the specifications of the application.
- The output based on the test data and the specifications of the application.
- The writing of test scenarios and the execution of testcases.
- The comparison of actual and expected results based on the executed testcases.

2. **Non-functional Testing**

This section is based upon testing an application from its non-functional attributes. Non-functional testing involves testing software from the requirements which are non-functional in nature but important such as performance, security, user interface, etc. Testing can be done in different levels of SDLC.

6.3 Unit Test cases

Unit testing is a software development process in which the smallest testable parts of an application, called units, are individually and independently scrutinized for proper operation. Unit testing is often automated but it can also be done manually. The goal of unit testing is to isolate each part of the program and show that individual parts are correct in terms of requirements and functionality. Test cases and results are shown in the Tables.

Unit Testing Benefits:

- Unit testing increases confidence in changing/ maintaining code.
- Codes are more reusable.
- Development is faster.
- The cost of fixing a defect detected during unit testing is lesser in comparison to that of defects detected at higher levels.
- Debugging is easy.
- Codes are more reliable.

Loading the data Module Testing

The below Table 6.1 shows the successful test case for loading the fetus health detection data that is selected by the user to do the processing. Before pre-processing the data's, the data should match in dimensions. The resolution of the data's captured also plays a vital role in conveying false notions to the classifiers- thus affecting accuracy. In this unit the data's input to the system are checked whether they are loaded or not. If they are loaded then the expected output is reached.

Table 6.1: Test case for loading the data

Test Case Sl. No	1
Test Name	Load the input for Fetus health.
Test Feature	Checks whether the data selected are loaded or not.
Output Expected	The data to be loaded from the dataset.
Output Obtained	Loaded fetus health data which is used for the processing
Result	Successful

Pre-Processing Module Testing

The original data is given as input. In the preprocessing step noise reduction method is used. Where Pepper and salt method is added to original data and the clear data will be obtained. In the below table 6.2, the noise reduction unit testing is applied if the data is clear the expected output is reached.

Table 6.2: Test case for Pre-Processing

Test Case Sl. No	2
Test Name	Noise Reduction
Test Feature	Pepper and salt added to data.
Output Expected	Clear data
Output Obtained	The noise reduced data
Result	Successful.

Training Module Testing

Table 6.3: Test case for training

Test Case Sl. No	3
Test Name	Training data.
Test Feature	Training the Data dataset.
Output Expected	The data's to be trained.
Output Obtained	Trained dataset
Result	Successful.

The Table 6.3 shows the successful test case for training the Fetus health detection data. Here in training the data is loaded and then checks whether it is already loaded and trained or not, if not trained then it goes to the process of training. The data selected for training undergoes preprocessing and then feature extraction and then finally the data are trained. These processes are done in Training GUI. And after training the dataset which contains set of fetus health detection data then can test and calculate the accuracy of result obtained.

6.4 Summary

This chapter presents system testing in section 6.1, which consists of testing principle for the various modules of the Fetus health Detection. Section 6.2 gives a complete view of the testing methods which includes Test Name, Test Feature, Output Expected, Output Obtained and Result.

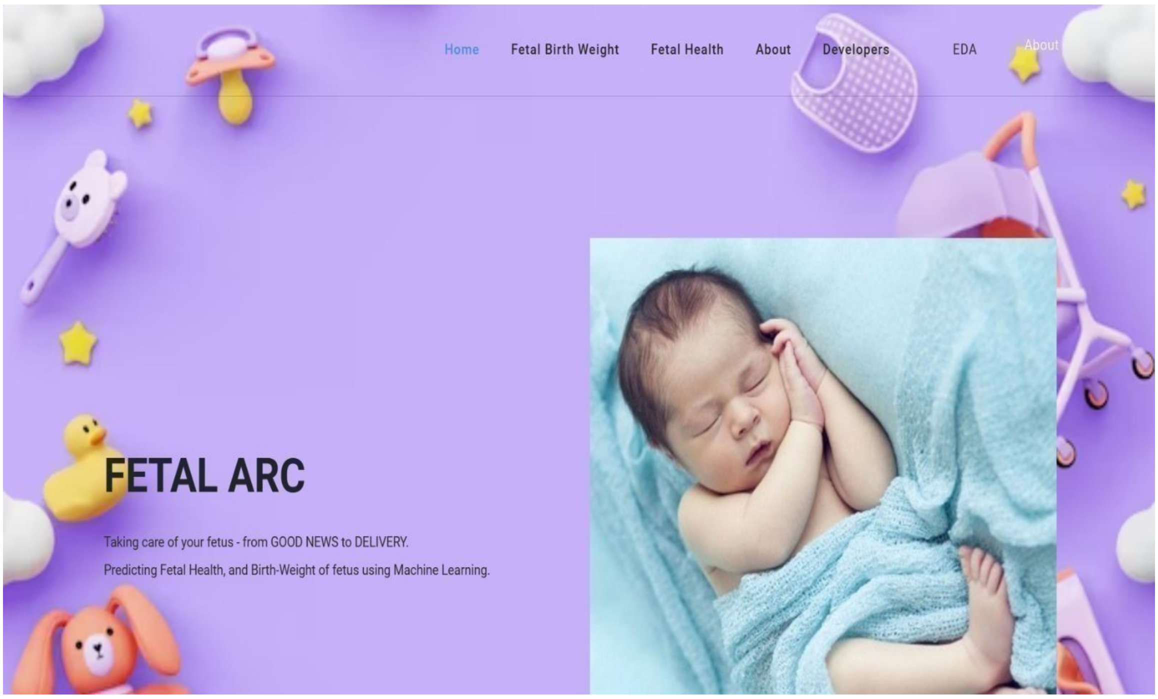
CHAPTER 7

RESULTS AND DISCUSSION

Fetal health prediction is a crucial area of prenatal care, aiming to assess the well-being of the fetus and predict potential complications during pregnancy. Various factors, including maternal health, fetal biophysical parameters, and genetic factors, can influence fetal health. Predictive models for fetal health often leverage machine learning algorithms to analyze these complex interactions and make accurate predictions.

7.1 EXPERIMENTALRESULTS

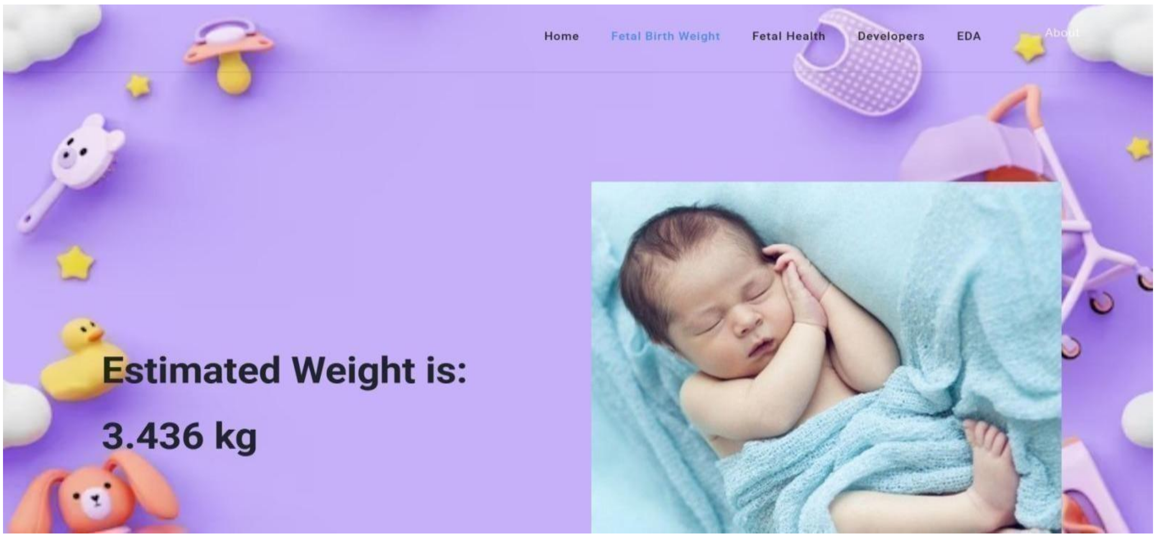
7.1.1 Snapshot of Fetal Health Prediction Home Page



Snapshot7.1: Snapshot of Fetal Health Prediction Homepage

As shown in the above Snapshot 7.1 It shows fetal health prediction Home Page serves as a central hub for accessing and interpreting predictive models designed to assess the health of fetuses. It provides a user-friendly interface for healthcare professionals and researchers to input relevant data, such as omics data, temporal information, and other clinical variables, Additionally, the home page offers tools for model interpretability, enabling users to understand the underlying factors driving the predictions. Overall, the fetal health prediction home page aims to revolutionize prenatal care by providing cutting-edge predictive models that support informed decision-making and improve outcomes for both mothers and babies.

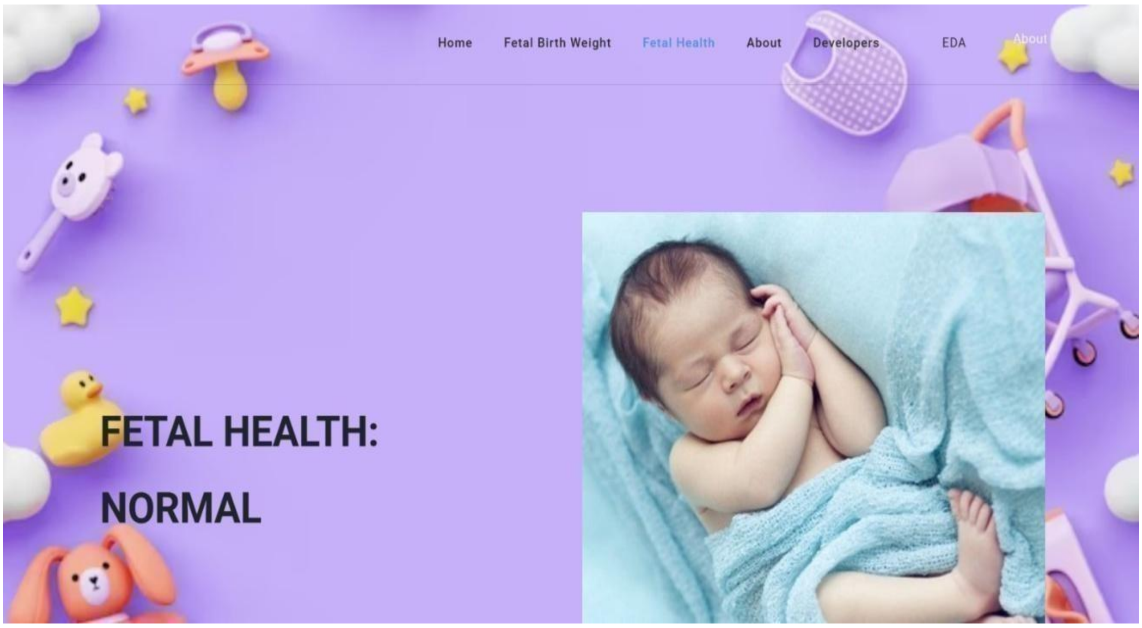
7.1.2 Snapshot of Fetal Birth Weight



Snapshot 7.2: Snapshot of Fetal Birth Weight

As shown in the above Snapshot 7.2 It shows Fetal birth weight refers to the weight of a baby at the time of birth and is a critical indicator of the baby's health and development. It is influenced by various factors, including genetics, maternal health, nutrition, and gestational age. Fetal birth weight is typically estimated during prenatal in managing pregnancy and delivery. Babies with low birth weight may face an increased risk of health complications, including respiratory issues, infections, and developmental delays and interventions may be recommended to optimize birth outcomes based on these measurements.

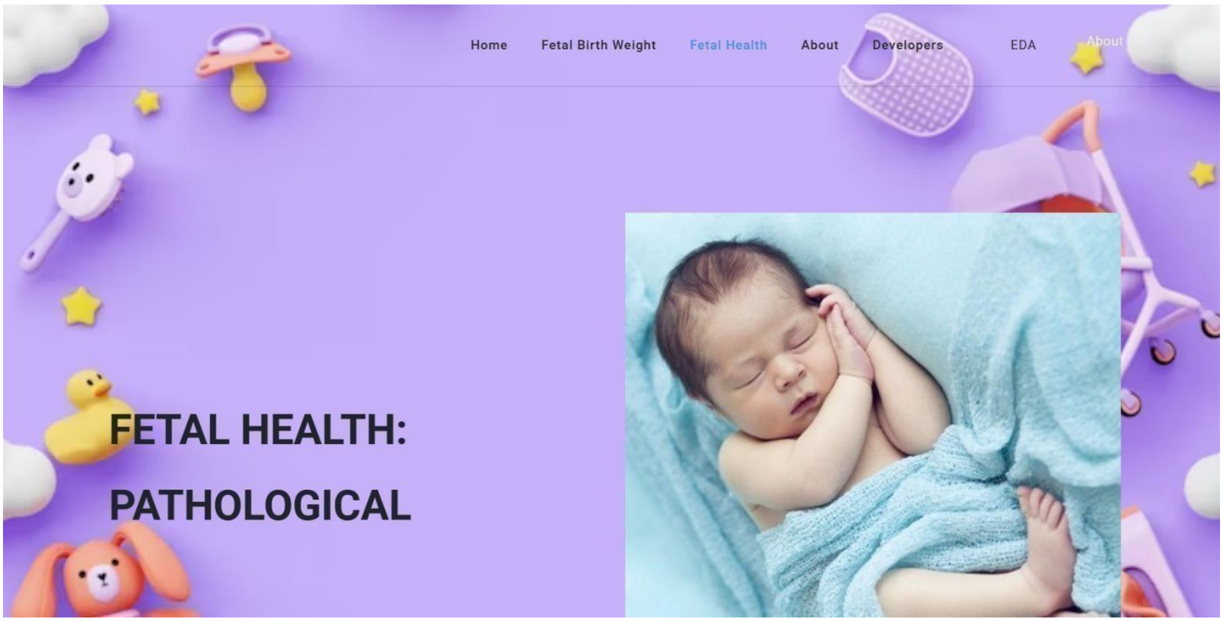
7.1.3 Snapshot of Normal Fetus



Snapshot 7.3: Snapshot of Normal Fetus

As shown in the above Snapshot 7.3 It shows shows Predicting normal fetal health involves assessing various factors to determine the well-being of the fetus during pregnancy. This prediction typically involves monitoring maternal health, examinations and other diagnostic tests and fetal movements are closely monitored to ensure the fetus is developing as expected. Additionally, maternal factors such as blood pressure, blood sugar levels, and overall health are also considered. By combining these factors and monitoring them over time, healthcare providers can make informed predictions about the normal health of the fetus and take necessary actions to ensure a healthy pregnancy and delivery.

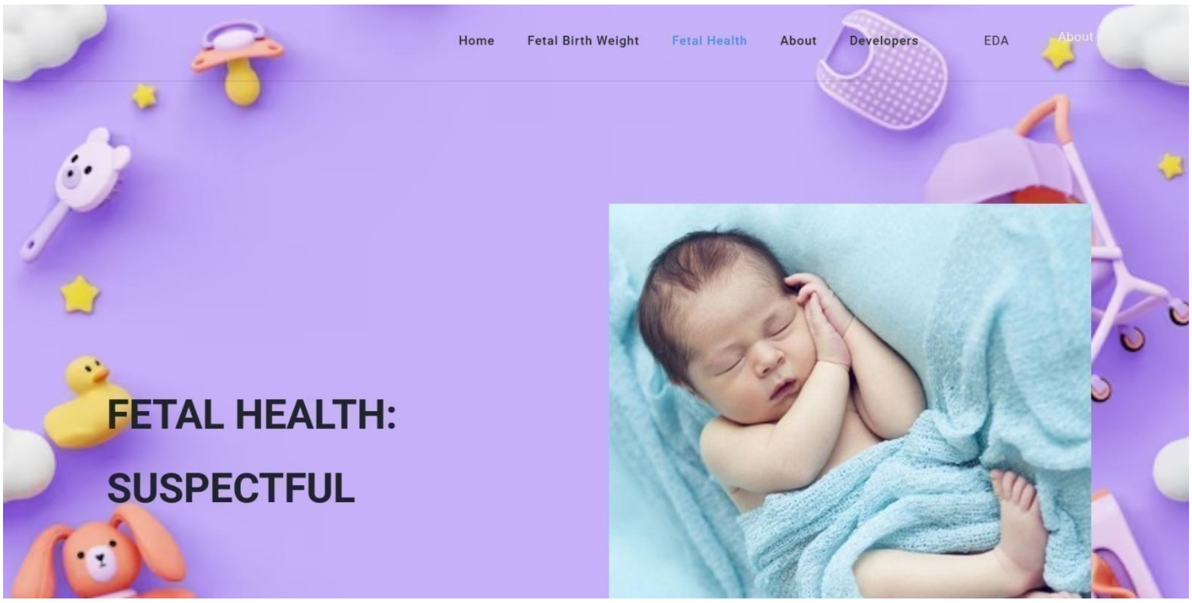
7.1.4 Snapshot of Pathological Fetus



Snapshot 7.4: Snapshot of Pathological Fetus

As shown in the above Snapshot 7.4 It shows Predicting pathological fetus. Health involves identifying potential risks or abnormalities in the development of the fetus during pregnancy. This prediction often relies on a combination of factors, including maternal health history, prenatal screening tests, and ultrasound examinations. Healthcare providers look for signs of fetal distress or abnormalities, such as abnormal growth patterns, congenital anomalies, or signs of fetal infection. Maternal factors such as high blood pressure by carefully monitoring these factors and conducting appropriate tests, healthcare providers can predict the likelihood of pathological fetal health and take steps to manage or mitigate these risks to ensure the best possible outcome for both the mother and the baby.

7.1.5 Snapshot of Successful Fetus



Snapshot 7.5: Snapshot of Successful Fetus

As shown in the above Snapshot 7.5 It shows Predicting suspect fetal health involves identifying potential issues or uncertainties regarding the health of the fetus during pregnancy. This prediction often arises when there are certain indicators or findings that are not definitive enough to diagnose a pathology but suggest a need for closer monitoring or further investigation. These indicators could include slightly abnormal fetal growth, variations in amniotic fluid levels, or borderline results from prenatal screening tests. In such cases While suspect fetal health does not necessarily indicate a serious problem, it warrants careful attention and proactive management to address any potential issues early and minimize risks to both the mother and the baby.

7.1.6 Snapshot of Classification Report

TRAIN CLASSIFICATION REPORT

	precision	recall	f1-score	support
NORMAL	0.99	0.93	0.96	1322
PATHOLOGICAL	0.96	0.97	0.96	147
SUSPECT	0.72	0.96	0.82	231
accuracy			0.94	1700
macro avg	0.89	0.95	0.91	1700
weighted avg	0.95	0.94	0.94	1700

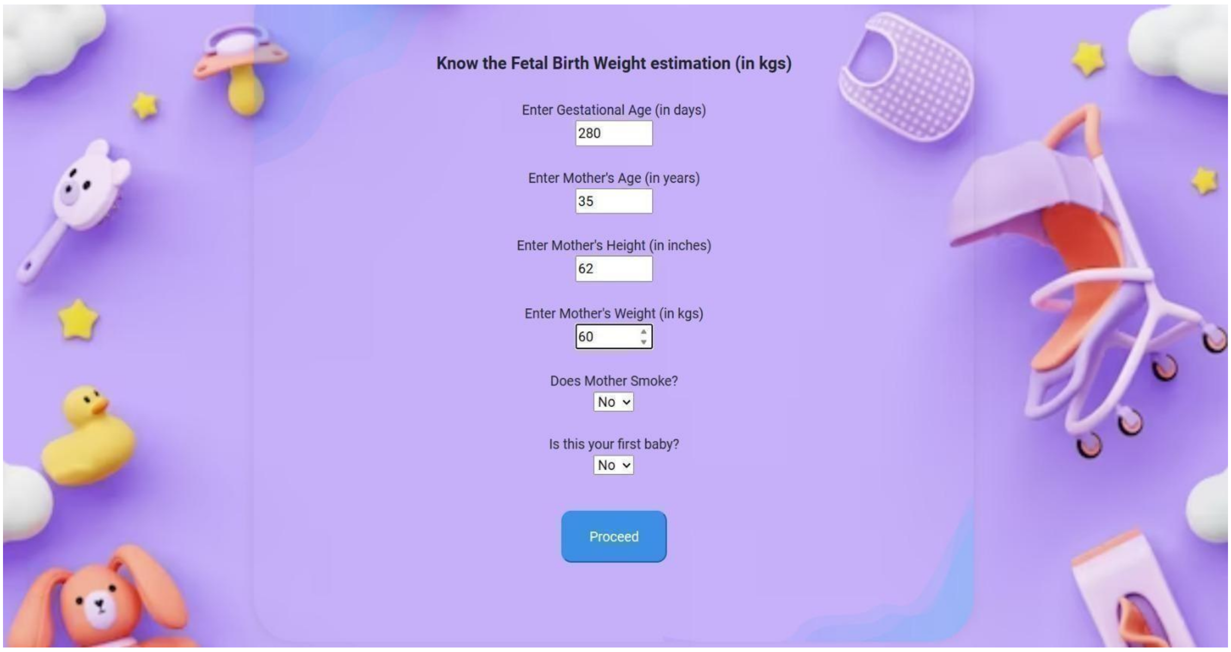
TEST CLASSIFICATION REPORT

	precision	recall	f1-score	support
NORMAL	0.99	0.91	0.95	333
PATHOLOGICAL	0.84	0.93	0.89	29
SUSPECT	0.69	0.92	0.79	64
accuracy			0.92	426
macro avg	0.84	0.92	0.87	426
weighted avg	0.93	0.92	0.92	426

Snapshot 7.6: Snapshot of Classification Report

As shown in the above Snapshot 7.6 it shows A Classification Report is a summary of the performance of a classification model that is used in machine learning. It provides a detailed breakdown of how well the model is performing in terms of its ability to correctly classify instances into different classes. The report typically includes metrics such as precision, recall, F1-score, and support for each class in the dataset. Precision is the ratio of true positives to the sum of true positives and false positives. It measures the accuracy of the positive predictions made by the model.

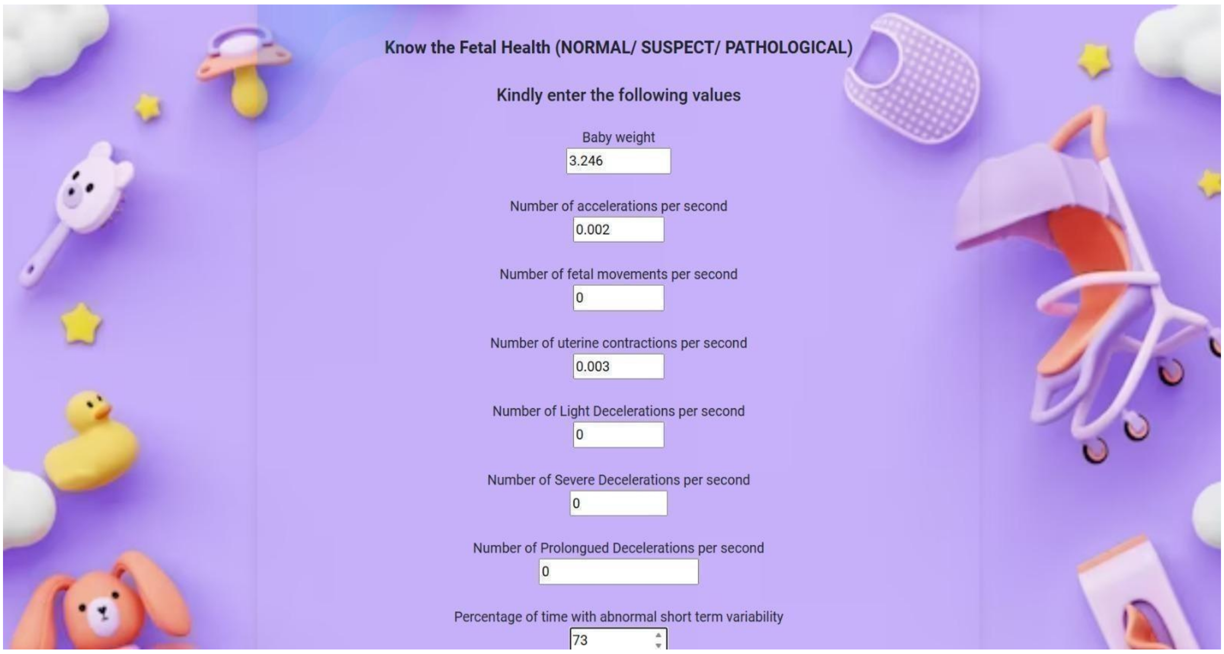
7.1.7 Snapshot of Fetal Birth Weight



Snapshot 7.7: Snapshot of Fetal Birth Weight

As shown in the above Snapshot 7.7 it shows Fetal birth weight refers to the weight of a baby at the time of birth and is an important indicator of the baby's health and development. It is influenced by various factors, including genetics, maternal health, and environmental factors. Estimating fetal birth weight is crucial for monitoring fetal growth and assessing the risk of complications during pregnancy and delivery, such as macrosomia (large birth weight) or intrauterine growth restriction (small birth weight). Healthcare providers use various methods, such as ultrasound measurements, maternal characteristics, and fetal biometric parameters, to estimate fetal birth weight and manage pregnancies accordingly to ensure the health and well-being of both the mother and the baby. It is influenced by various factors, including genetics, maternal health, and environmental factors. Estimating fetal birth weight is crucial for monitoring fetal growth and assessing the risk of complications during pregnancy

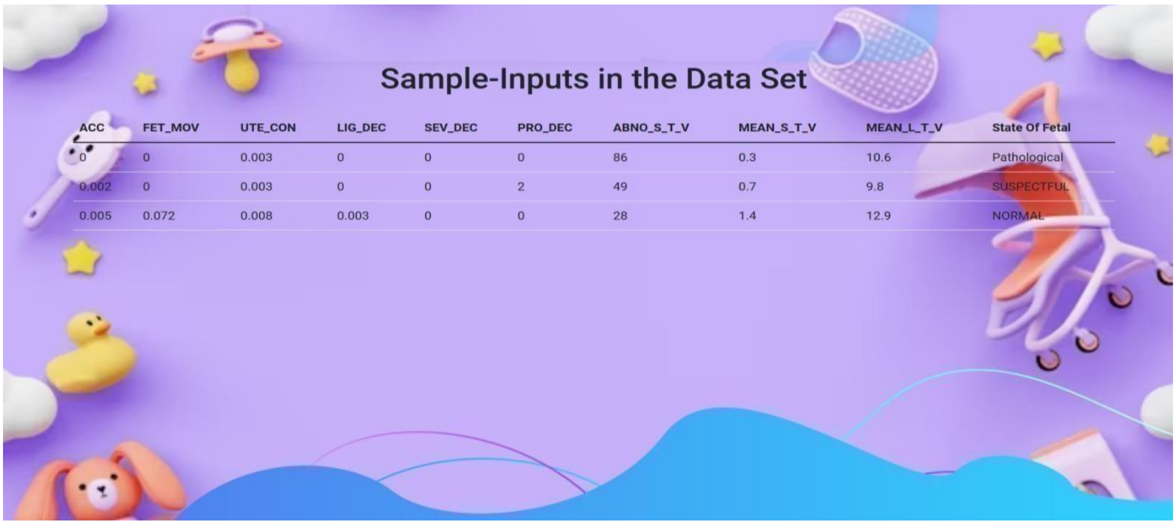
7.1.8 Snapshot of Fetal Health



Snapshot 7.8: Snapshot of Fetal Health

As shown in the above Snapshot 7.8 it shows Fetal health refers to the well-being and development of the fetus during pregnancy. It is influenced by a variety of factors, including maternal health, genetics, and environmental conditions. Monitoring fetal health is essential to ensure a successful pregnancy and the delivery of a healthy baby. Healthcare providers assess fetal health through various methods, such as ultrasound scans, fetal heart rate monitoring, and maternal blood tests. Regular prenatal care, proper nutrition, and avoiding harmful substances are important for promoting fetal health. Any concerns about fetal health should be promptly addressed by healthcare professionals to optimize pregnancy outcomes.

7.1.9 Snapshot of Sample inputs in the dataset



Snapshot 7.9: Snapshot of Sample inputs in the dataset

As shown in the above Snapshot 7.9 Sample inputs in the dataset. It shows a paragraph of text describing a dataset, sample inputs could be described in a more general way, focusing on the types of information that are included for each data point. For instance, in a dataset about cars, sample inputs might include the make and model of the car, the year it was manufactured, the mileage, and so on. These inputs help to provide context and structure to the dataset, allowing researchers to understand the information that is being collected and how it is organized.

7.2 SUMMARY

Fetal health prediction snapshots are a collection of data points and insights used to assess and predict the health of a fetus during pregnancy. These snapshots typically include information such as maternal health status, ultrasound images, fetal heart rate monitoring data, and other relevant medical data. By analyzing these snapshots, healthcare providers can identify potential risks to the fetus and take proactive measures to ensure a healthy pregnancy and delivery.

CHAPTER 8

CONCLUSION AND FUTURE ENHANCEMENTS

8.1 CONCLUSION

In conclusion, the development of the prenatal care system represents a significant stride towards advancing maternal healthcare through the integration of machine learning technologies. The successful implementation of modules, from dataset gathering to model evaluation, underscores the project's commitment to providing healthcare professionals with a powerful tool for early risk detection and personalized care strategies. By harnessing the predictive capabilities of machine learning algorithms, the system not only facilitates accurate fetal health classification and baby weight prediction but also empowers clinicians with actionable insights, ultimately contributing to improved maternal and neonatal outcomes.

As the project culminates, its impact extends beyond the realms of technology to the realm of healthcare, promising a paradigm shift in prenatal care. The synergistic blend of sophisticated backend algorithms and an intuitive frontend interface positions the system as a valuable asset in the hands of healthcare practitioners. With the potential to revolutionize proactive maternal care, this project stands as a testament to the transformative power of technology in enhancing and safeguarding the health of both mothers and newborns.

8.2 FUTUREENHANCEMENTS

Looking ahead, there are several avenues for future enhancements to further elevate the capabilities and impact of the prenatal care system. Firstly, the integration of additional maternal and fetal health parameters, such as genetic data, biomarkers, and lifestyle factors, could enrich the dataset, providing a more comprehensive basis for machine learning analysis. This expansion would contribute to a more nuanced understanding of the intricate dynamics influencing fetal health, enabling the system to offer even more accurate predictions and personalized recommendations to healthcare professionals.

REFERENCES

[1] M.M.Van Dyne, L.K. Woolery, J.GryzmalaBusse, C.Tsatsoulis, "Using Machine Learning and Expert Systems to Predict Preterm delivery in Pregnant Women",DOI: 10.1109/CAIA 10.1109/CAIA.1994.323655, IEEE

[2] Linda Goodwin, Sean Maher," Data Mining for Preterm Predic- tion", 2000 ACM 1-58114-239-5/00/003.

[3] Christina Catley, Monique Frize, C.Robin Walker and Dorina C. Petriu, 'Predicting High-Risk Preterm Birth Using Artificial Neural Networks', IEEE Transactions on Information Technolo- gyin Biomedicine, Vol.10, No.3, July 2006

[4] YavarNaddaf, MojdehJalaliHeravi, Amit Satsangi, 'Predicting Preterm Birth based on Maternal and Fetal data', Google Scholar, 2008

[5] Luiz Fernando, Paloma Maria S, RachaRizol, Luciana B.Abiuzi, ' Establishing the risk of neonatal mortalityusing a fuzzypredic- tive model', Cad SaudePublica, 2009

[6] Matharage S. Alahakoon O., Alahakoon D, Kapurubandara S, Nayyar R, Mukherjee M, Jagadish U, Yim S, Alahakoon, 'Ana- lysing Stillbirth Data using Dynamic Self Organizing Maps', 2011 IEEE DOI 10.1109/DEXA.2011.14

[7] IlliaVovsha, AshwathRajan, AnsafSalleb-Aouissi, Anita Raja, AxiniaRadeva, Hatim Diab, Ashish Tomar, Ronald Wapner, "Predicting Preterm Birth is Not Elusive: Machine Learning paves way to individual Wellness', AAAI Symposium Series, 2014

[8] R.R. Pate, M. Pratt, S.N. Blair, W.L. Haskell, C.A. Macera, C. Bouchard, D. Buchner, W. Ettinger, G.W. Heath, A.C. King, Physical activity and public health: a recommendation from the centers for disease control and prevention and the american college of sports medicine, JAMA 273 (5) (1995) 402– 407, doi:10.1001/jama.1995.03520290054029. jama_273_5_029.pdf

[9] C. for Disease Control, P. (CDC), About Adult BMI, 2017, [Online; Last accessed: 5 April 2018], https://www.cdc.gov/healthweight/assessing/bmi/adult_ bmi/index.html.

[10] N.K. Brad Severtson, G. Ericson, choose parameters to optimize your algorithms in Azure machine learning, 2017, [Online; Last accessed: 5 April 2018], https://docs.microsoft.com/en-us/azure/machine-learning/ machine-learning-algorithm-parameters-optimize.

[11] B.N. Lakshmi, T.S. Indumathi, N. Ravi, Prediction based health monitoring in pregnant women, in: 2015 International Conference on Applied and Theoretical Computing and Communication. doi:10.1109/ICATCCT.2015.7456954

[12] R. Sawant, N. Gaikwad, Hybrid prediction method for pregnancy data set, in: 2015 1st International Conference on Next Generation Computing Technologies (NGCT), 2015, pp. 918–920, doi:10.1109/NGCT.2015.7375253