**EARLY DETECTION OF FETAL DISTRESS USING MACHINE LEARNING MODELS**

**B. Tech Project Report**

*Submitted to The University College of Engineering and Technology,*

*Acharya Nagarjuna University in partial fulfilment of Requirement for the award of Degree*

**BACHELOR OF TECHNOLOGY**

***in***

**COMPUTER SCIENCE AND ENGINEERING**

by

|  |  |
| --- | --- |
| **Telaprolu Uma Manasa** | **(Y21CS3256)** |
| **Nageti Kuladeep** | **(L22CS3269)** |
| **Gokivada Mounika** | **(Y21CS3270)** |
| **Aleti Surendra Kumar** | **(Y21CS3262)** |

***Under the Guidance of***

# Mrs. K. Prasanthi , M.Tech



**UNIVERSITY COLLEGE OF ENGINEERING & TECHNOLOGY ACHARYA NAGARJUNA UNIVERSITY**

**NAGARJUNA NAGAR -522510, GUNTUR, A.P., INDIA**

**2021-2025**

**UNIVERSITY COLLEGE OF ENGINEERING &**

**TECHNOLOGY ACHARYA NAGARJUNA UNIVERSITY**

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**



**CERTIFICATE**

This is to certify that the project entitled **“ Early Detection of Fetal Distress using Machine Learning Models”** is a bona fide record of the project work done by **Telaprolu Uma Manasa (Y21CS3256), Nageti Kuladeep(L22CS3269), Gokivada Mounika(Y21CS3270), Aleti Surendra Kumar(Y21CS3262)** under my supervision and guidance, in partial fulfilment of the requirements for the award of Degree in Computer Science & Engineering from University College of

Engineering & Technology, Guntur for the academic year 2024-25

|  |  |
| --- | --- |
| **………………………….** | **……………………** |
| **Mrs. K. Prasanthi** | **Dr. U. Sathish Kumar** |
| Project Guide, Dept. of C.S.E | HOD, Dept of C.S.E |

## ………………………

External Examiner

## DECLARATION

We hereby declare that the project entitled, **“Early Detection of Fetal Distress using Machine Learning ”** was carried out and written by me under the guidance of Mrs. K. Prasanthi, Department of Computer Science & Engineering, University College of Engineering & Technology, Acharya Nagarjuna University. This work has not been previously formed the basis for the award of any degree or diploma or certificate nor has been submitted elsewhere for the award of any degree or diploma.

|  |  |  |
| --- | --- | --- |
| Place: ANUCET | **Telaprolu Uma Manasa** | **(Y21CS3256)** |
| Date: | **Nageti Kuladeep** | **(L22CS3269)** |

**Gokivada Mounika (Y21CS3270)**

**Aleti Surendra Kumar (Y21CS3262)**

# ACKNOWLEDGEMENTS

In our opinion writing acknowledgement is a difficult but most beautiful part of the project report ending. It is difficult to express the heartfelt accomplishments and the gratitude for those who really or virtually supported for making the work successful and memorable.

We would like to convey our sincere gratitude and admiration to our Guide

**Mrs. K. Prasanthi**, University College of Engineering & Technology, Acharya Nagarjuna University (ANU), Guntur, for her excellent support and valuable guidance throughout our study. Her marvelous advice motivated us to work on this, and make it successful. Her supportive nature and thought-provoking ideas made this work possible.

We are very grateful to **Dr. U. Sathish Kumar,** HOD, Dept of CSE, University College of Engineering & Technology, Acharya Nagarjuna University, for hir keen interest, counselling, constant encouragement, personal affection and valuable suggestions in completion of this research work.

We would like to express our heartfelt thanks to **Prof. P. Siddaiah**,

Principal, University College of Engineering & Technology, Acharya Nagarjuna University, Guntur, for guidance, mentorship, and valuable insights that have been instrumental in the successful completion of this work

We are deeply indebted to **Dr. U. Sathish Kumar**, Project Coordinator, for Conducting reviews and giving constructive criticism throughout the project duration.

We are grateful to the **Authorities** of University College of Engineering & Technology, Acharya Nagarjuna University, enabling us to pursue our work.

We are very thankful to all **Teaching** and **Non-teaching staff** of department of Computer Science & Engineering, University College of Engineering & Technology, Acharya Nagarjuna University , Guntur for their encouragement, co- Operation and assistance during the work period.

We would like to express our grateful thanks to **our parents** our whole beloved family for their love, unwavering support and encouragement throughtout the time of our study.

Finally , we thank one and all, who helped us directly or indirectly for Completing this work**.**

With gratitude

**Telaprolu Uma Manasa (Y21CS3256)**

**Nageti Kuladeep (L22CS3269)**

**Gokivada Mounika (Y21CS3270) Aleti Surendra Kumar (Y21CS3262)**

# *ABSTRACT*

Fetal health monitoring is crucial for early detection of complications during pregnancy, ensuring timely medical intervention. Traditional methods of fetal health assessment rely on cardiotocography (CTG), which requires expert interpretation. This project aims to develop a machine learning model to predict fetal health status—Normal, Suspect, or Pathological—based on CTG data.

The system leverages machine learning algorithms such as Random Forest, Logistic regression, K-Nearest Neighbours, Gradient Boosting Classifier to classify fetal health. The dataset used includes key maternal and fetal parameters such as fetal heart rate (FHR), uterine contractions, and accelerations. Data preprocessing techniques, feature selection, and hyperparameter tuning are applied to improve model performance.

Through using all implemented Machine Learning models, we have managed to achieve an accuracy of 99% through Random Forest algorithm. Evaluation metrics such as accuracy, precision, recall, F1-score, are used to compare models, ensuring reliable predictions. The trained model is deployed using a Flask/Streamlit-based web application, allowing easy access for healthcare professionals. This project demonstrates the potential of machine learning in prenatal healthcare, providing an efficient and automated approach to fetal health assessment, reducing dependency on manual interpretation, and improving maternal-fetal outcomes.

**Key Words**- Cardiotocography (CTG), fetal health rate (FHR), ML models, uterine contractions (UC).

|  |  |  |
| --- | --- | --- |
|  |  | **INDEX** |
|  | **S.NO** | **CONTENT PAGE NO** |
| \ | 1. | Introduction 1 - 4  1.1 Machine Learning  1.2 Some Machine learning models  1.3 Apllications of Machine Learning  1.4 Importance of machine Learning using Python |
|  | 2. | Literature Survey 5 - 7 |
|  | 3. | Existing System 8 - 9 |
|  | 4. | Proposed System 10 -13 |
|  | 5. | System Requirements 14  5.1 Hardware Requirements  5.2 Software Requirements |
|  | 6. | System Design 15 -18  6.1 Scope of the project  6.2 Analysis  6.3 Data Pre-Processing  6.4 Feature Extraction  6.5 Model Building  6.6 Classification  6.7 Confusion Matrix |
|  | 7. | Design Overview 19 |
|  | 8. | Implementation 20 - 38 |
|  | 9. | Result Analysis 39 - 44 |
|  | 10. | Test Cases 45 -46 |
|  | 11. | User Interface 47 |
|  | 12. | Conclusion 48 |
|  | 13. | Future Scope 49 |
|  | 14. | References 50- 52 |

|  |  |
| --- | --- |
| **LIST OF FIGURES** |  |
| **S.NO CAPTION** | **PAGE NO** |
| 1. Fig 3.1 Flow chart of existing system | 8 |
| 2. Fig 4.1 Proposed System | 11 |
| 3. Fig 6.1 Visualsing the columns of Dataset | 17 |
| 4. Fig 7.1 Design overview | 19 |
| 5. Fig 9.1 Random Forest ROC curve | 39 |
| 6. Fig 9.2 Decision Tree ROC curve | 39 |
| 7. Fig 9.3 Gradient Boosting Classifier ROC curve | 39 |
| 8. Fig 9.4 Logistic Regression ROC curve | 39 |
| 9. Fig 9.5 Support Vector Machine ROC curve | 40 |
| 10. Fig 9.6 K-Nearest Neighbors ROC curve | 40 |
| 11. Fig 9.7 Random Forest Confusion Matrix | 41 |
| 12. Fig 9.8 Decision Tree Confusion Matrix | 41 |
| 13. Fig 9.9 Gradient Boosting Classifier Confusion Matrix | 41 |
| 14. Fig 9.10 Logistic Regression Confusion Matrix | 41 |
| 15. Fig 9.11 Support Vector Machine Confusion Matrix | 41 |
| 16. Fig 9.12 K-Nearest Neighbors Confusion Matrix | 41 |
| 17. Fig 9.13 Comparison of ACCURACY of algorithms | 42 |
| 18. Fig 9.14 Comparison of PRECISION of algorithms | 42 |
| 19. Fig 9.15 Comparison of RECALL of algorithms | 43 |
| 20. Fig 9.16 Comparison of F1-SCORE of algorithms | 43 |
| 21. Fig 10.1 Screen displaying features to predict | 45 |
| 22. Fig 10.2 Screen displaying prediction after input | 45 |
| 23. Fig 10.3 Screen displaying invalid input | 46 |
| 24. Fig 11.1 Output screen | 47 |

|  |  |
| --- | --- |
|  |  |
| **S.NO TABLE NAME** | **PAGE NO** |
| Table 6.1: Dataset description | 16 |
| Table 9.1: Classification report of various models | 44 |

## LIST OF TABLES

# LIST OF ACRONYMS

|  |  |
| --- | --- |
| **Acronym** | **Full Form** |
| CTG | Cardiotocography |
| FHR | Fetal Health Rate |
| ML | Machine Learning |
| UC | Uterine Contractions |
| KNN | K-Nearest Neighbors |
| SVM | Support Vector Machine |
| ROC | Receiver Operating Characteristic |
| AOC | Area Under curve |
| CSV | Comma Separated Values |
| VIF | Variance Inflation Factor |

## 1. INTRODUCTION

For analysing the fetuses health in the womb, the continuous cardiotocography is the most essential tool so far, which helps in assessing the fetal heart rate and health during pregnancy. Some previous works that have applied Machine Learning neural networks for the purpose of automatic CTG categorization depended over previous dataset with a total sample size of 2126 measurements consisting of 21 features in assessment of Fetal Health Rate and Uterine Contractions using Cardiotocography [1].These classifications were done resulting in utilization of several algorithms of Machine Learning such as Random Forests, SVM, DT and KNN accuracy, F1-score, area under ROC curve (AUC) greater than 90%[2]. However, several other researches also utilize ensemble learning concept with basic model called Random Forest followed by final candidate called LightGBM which hybridizes Gaussian Naïve Bayes optimization mapping log loss models by computing K- Fold Cross Validation. By this means a lot of real-time data have been evaluated effectively making it easier in giving better solutions and provide a platform for other models to perform classification purposes[3] (Comert et al., 2019).

One technique used in observing fetal welfare is cardiotocography (CTG), which records the heart rate of the fetus and contractions of the uterus over a period of time [4].To establish the validity of the experimental task, we utilized multiple performance measures including accuracy, precision, recall and F1-score[5]. In order to predict the wellness of a fetus, this research employs various Machine Learning algorithms to categorize cardiotocographic data with regards to the health conditions. This examines comprise twelve Machine Learning methods concentrating on FHR and UC signals allowing extraction of quantitative features. In this study, I have developed a classifier that automatically predicts fetal health using different learning machines. I have used a database of CTG data from the university of [6]California.

Fetal health monitoring is a crucial aspect of prenatal care, aimed at assessing the well-being of a fetus and identifying any potential complications during pregnancy. One of the widely used techniques for fetal monitoring is Cardiotocography (CTG)**,** which records fetal heart rate (FHR) and uterine contractions (UC) to help detect abnormalities[7]. However, traditional methods of analyzing CTG data rely on manual

interpretation by medical professionals, which is time-consuming and prone to human error. To enhance accuracy and efficiency, Machine Learning (ML) algorithms have been

1

increasingly applied in fetal health classification.

The primary goal of this research is to develop an efficient automated fetal health classification system that can assist healthcare professionals in making timely and accurate diagnoses. By leveraging Machine Learning techniques, the system aims to reduce misinterpretation errors, optimize medical resources, and improve maternal and fetal outcomes[8]. The study also evaluates different models based on key performance metrics such as accuracy, precision, recall, and F1-score to determine the most effective approach for classification.

This work contributes to the ongoing advancements in AI-driven healthcare solutions**,** demonstrating how Machine Learning can transform prenatal care by enabling faster, more reliable, and data-driven decision-making for fetal health assessment.

### 1.1 Overview of Machine Learning

Machine Learning (ML) is a field of artificial intelligence (AI) that focuses on developing algorithms and models that enable computers to learn from and make decisions based on data. Unlike traditional programming, where explicit instructions are given to a computer, Machine Learning systems identify patterns in data and make predictions or decisions without human intervention.

**Types of Machine Learning:**

* **Supervised Learning**: In this approach, the model is trained on labeled data (where input-output pairs are provided). The goal is to learn a mapping from inputs to outputs, such as predicting house prices based on features like size and location. Examples include regression and classification tasks.
* **Unsupervised Learning**: In this case, the model is given data without explicit labels and must find patterns or structures within the data, such as grouping similar items together. Examples include clustering and dimensionality reduction.

2

* **Semi-Supervised Learning**: This combines both labelled and unlabelled data for training. It is used when labelling large amounts of data is expensive or time- consuming.
* **Reinforcement Learning**: This involves an agent that interacts with an environment and learns to make decisions by receiving rewards or penalties based on its actions. The goal is to maximize cumulative reward over time. It’s commonly used in robotics, gaming, and autonomous vehicles.

### 1.2 Machine Learning models

* **Linear Regression**: A model used for predicting a continuous target variable based on one or more input features.
* **Logistic Regression**: Used for binary classification tasks, where the goal is to predict one of two outcomes.
* **Decision Trees**: A flowchart-like structure that splits data into subsets based on feature values, used for both classification and regression tasks.
* **Support Vector Machines (SVM)**: A powerful classifier that separates data into classes using hyperplanes.
* **K-Nearest Neighbours (KNN)**: A simple algorithm that classifies a data point based on the majority class of its nearest neighbours.
* **Neural Networks and Deep Learning**: Models inspired by the human brain that can learn complex patterns, often used for image recognition, natural language processing, etc.

### 1.3 Applications of Machine Learning

* **Image and Speech Recognition**: Machine Learning models are used to classify objects in images or recognize spoken words.
* **Natural Language Processing (NLP)**: Used in tasks like sentiment analysis, translation, and chatbots.
* **Recommendation Systems**: ML is used in services like Netflix, Amazon, or Spotify to recommend products or media based on past behaviour.
* **Autonomous Vehicles**: Self-driving cars use Machine Learning algorithms to understand their environment and make driving decisions.

### 1.4 Importance of Machine Learning using python

Machine Learning is transforming industries by enabling automation, uncovering insights from data, and driving innovations like autonomous vehicles and personalized medicine. Python plays a crucial role in this field due to its simplicity, extensive libraries (like TensorFlow, Scikit-learn, and Pandas), and strong community support. It allows developers to efficiently handle tasks such as data preprocessing, model training, and deployment. With its versatility and integration capabilities, Python empowers organizations to create scalable, dynamic, and adaptive ML solutions, making it indispensable in applications ranging from healthcare and finance to retail and transportation.

## 2. LITERATURE SURVEY

Several research studies have explored the application of Machine Learning models for fetal health classification using Cardiotocography (CTG) data. Researchers have employed various algorithms and methodologies to improve prediction accuracy and clinical decision-making.

As per the records of WHO, 6.3 million fetal deaths occur each year [1]. One such issue on datasets in the cardiotocography is limited details from expectant mothers. So far, CTG remain only available non-invasive and economic instrument as a continuous device for monitoring of baby health. Measurement of performance for models, merged performance metric, and ROC-AUC were used to validate the outcome. By this means a lot of real-time data have been evaluated effectively making it easier in giving better solutions and provide a platform for other models to perform classification purposes [3] (Cömert et al., 2019).

Some previous works that have applied Machine Learning neural networks for the purpose of automatic CTG categorization depended over previous dataset with a total sample size of 2126 measurements consisting of 21 features in assessment of Fetal Health Rate and Uterine Contractions using Cardiotocography [4]. To establish the validity of the experimental task, we utilized multiple performance measures including accuracy, precision, recall and F1-score[7]; we also employed a scaling method called standard scalar for creating an unbiased dataset. This is so as to ensure timely intervention can be made by both patients and physicians on fetal condition. Historically, Machine Learning algorithms have revolutionized the interpretation of medical data by allowing for more effective diagnosis of disease, therapy and prognosis [8].

In this study several ML algorithms are employed to label the cardiotocographic data in terms of the health condition in order to predict fetal well-being. Presently obstetricians interpret CTG through such parameters as FHR patterns, accelerations, decelerations, uterine activity etc. But despite using standard protocols including NIP or STV [9] their interpretations remain highly subjective with large inter-observer variability. The main aim of this work is to devise models that apply Machine Learning to judge fetal condition through cardiotocography. The examines comprise [10] twelve Machine Learning methods concentrating on FHR and UC signals allowing extraction of quantitative features.

Manual evaluation of CTG tests, a standard among

obstetricians, requires too much labor and is not reliable enough. For this reason, creating effective fetal health classification models are critical for resource management and time saving in hospitals [11]. Furthermore, the inception of these algorithms in healthcare has seen great improvements in the diagnosis, treatment and prediction of diseases. The various ML algorithms are utilized in this research to predict fetal health using cardiotocographic data by tagging the health status as normal, requires guarantee, pathological [13]. In earlier times obstetricians used to manually analyse CTG data, but this task is very slow and often leads to mistakes. Thus, it is necessary to come up with a method for categorization of fetal health which may quicken diagnosis and treatment and save medical resources. In this study, we developed a classifier that automatically predicts fetal health using different learning machines. We used a database of CTG data from the University of California[14]. The use of cardiotocography CTG combined with ST analysis in two randomized

controlled trials on intrapartum fetal monitoring raises the obstetricians’ ability to detect fetal hypoxia and take appropriate actions, resulting in improved perinatal outcomes .

The following section reviews relevant literature from different authors:

Abderrazzak Rafie et al. [21]conducted a study on fetal health classification using Machine Learning models such as artificial neural networks, random forest, and support vector machines. Their research focused on classifying fetal health into normal, suspicious, and pathological categories. Among the models tested, artificial neural networks achieved the highest accuracy of 96.7 percent. However, the study had limitations in exploring ensemble methods and deeper architectures for improved accuracy and generalization.

Sahana Das et al. [22] analyzed fetal health classification during both stages of labor using Machine Learning techniques, including support vector machines, random forest, multi-layer perceptron, and bagging classifiers. Their research showed that support vector machines and random forest achieved high accuracy ranging from 97 to 98 percent in detecting suspicious cases. The study highlighted the challenge of class imbalance in fetal health datasets and emphasized the need for real-time decision- making systems to improve clinical applications.

Nabilla Rahmayanti et al. [23] compared seven Machine Learning models, including artificial neural networks, long short-term memory, extreme gradient boosting, support vector machines, k-nearest neighbors, light gradient boosting

machine, and random forest. Their study demonstrated that light gradient boosting machine and random forest showed high accuracy ranging from 89 to 97 percent. The research recommended improving real-time implementation and handling class imbalances to enhance the performance of fetal health classification models.

Abolfazl Mehbodniya et al.[24] applied various Machine Learning techniques such as support vector machines, fuzzy-forest, and k-nearest neighbor regression for fetal health classification. The findings revealed that random forest achieved an accuracy of 94.5 percent, followed by support vector machines with 93 percent accuracy. The study suggested that integrating feature engineering and Deep Learning techniques could further enhance classification accuracy and robustness for medical applications.

Pankaj Bhowmik et al.[25] utilized tree-based ensemble learning to analyze cardiotocography data and predict fetal health risks. The study demonstrated high accuracy in fetal health classification using ensemble models, which proved effective in identifying potential complications. However, the research recommended further validation using diverse datasets and proposed integrating real-time monitoring systems to improve practical applications in obstetric healthcare.

To address these issues, some researchers have employed ensemble learning techniques such as XGBoost, LightGBM, and Bagging, which improve generalization and enhance model robustness as shown in Table 1. Despite these advancements, challenges remain in implementing real-time monitoring systems and integrating predictive models into clinical practice. Feature engineering and Deep Learning techniques, such as Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) networks, offer promising solutions for further improving classification accuracy.Future research should focus on optimizing Machine Learning models for real-time applications, handling class imbalances more effectively, and validating models on diverse datasets.

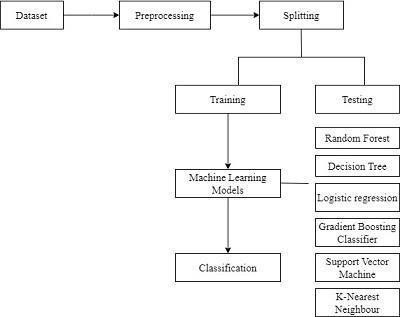
## 3. EXISTING SYSTEM

The existing system for fetal health monitoring is as shown in Fig 3.1 relies on the manual interpretation of cardiotocography data by medical professionals. [8] Cardiotocography is commonly used to assess fetal well-being by analysing fetal heart rate patterns, uterine contractions, and accelerations. However, this traditional approach has several limitations.

One of the major challenges is the subjectivity in cardiotocography interpretation, where different obstetricians may analyse the same data differently, leading to inconsistencies in diagnosis. This increases the risk of false positives or false negatives, which can delay necessary medical interventions. [11] Additionally, manual evaluation is time-consuming and requires continuous monitoring, making it difficult to assess multiple patients efficiently.

Another limitation of the existing system is the lack of real-time predictive capabilities. The current approach only allows for reactive decision-making, meaning interventions are based on already observed abnormalities rather than proactive risk assessment. The system does not integrate automated Machine Learning models, which can process large datasets and detect subtle patterns that may not be visible through traditional analysis.

The reliance on static threshold-based methods for classification further reduces efficiency. Traditional statistical techniques often fail to adapt to complex, nonlinear relationships in medical data, making them less effective in identifying high-risk pregnancies. [13]This highlights the need for an advanced approach that improves diagnostic accuracy and reduces the burden on healthcare professionals.



### Fig 3.1 Flowchart of Existing system

Limitations identified in the existing system are:

1. **Subjectivity in Interpretation** – Different authors may interpret the same cardiotocography (CTG) data differently, leading to inconsistencies in diagnosis.
2. **Time-Consuming Process** – Manual evaluation of CTG data requires continuous monitoring, making it inefficient, especially for handling multiple patients simultaneously.
3. **Lack of Real-Time Predictive Capabilities** – The current system relies on reactive decision-making rather than proactive risk assessment.
4. **No Machine Learning Integration** – The absence of automated models prevents the detection of subtle patterns that may not be visible through traditional analysis.
5. **Increased Burden on Healthcare Professionals** – Manual monitoring requires constant attention, which adds to the workload of medical staff.

## 4. PROPOSED SYSTEM

The proposed system involves creating a Machine Learning model to classify fetal health into categories like Normal, Suspect, and Pathological. It starts with data collection, followed by data cleaning and preparation, where outliers are removed, values are normalized, classes are balanced, and the dataset is made ready for analysis. Next, important features are selected using methods like correlation analysis and expert knowledge. Then, different Machine Learning algorithms such as Random Forest, Logistic Regression, SVM, and Decision Trees are used to build the model. The model’s performance is evaluated using metrics like accuracy, precision, recall, and confusion matrices. Finally, the best model is used to predict and classify fetal health into the desired categories.

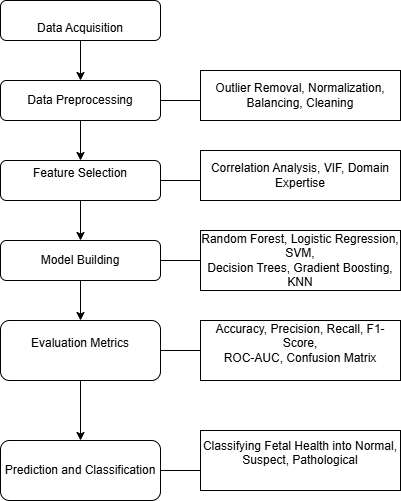
The proposed system introduces an AI-powered Machine Learning model to classify fetal health based on cardiotocography data. It aims to overcome the limitations of manual interpretation by providing an automated, accurate, and real-time classification of fetal health conditions into Normal, Suspect, and Pathological categories. The system follows a structured workflow that includes data pre-processing, feature selection, model training, and classification, ensuring efficient and reliable predictions.

The system begins with data collection, where fetal health data is obtained from cardiotocography readings. The raw data undergoes pre-processing steps such as handling missing values, normalization, outlier removal, and class balancing using techniques like SMOTE to ensure an unbiased dataset. Feature selection methods such as correlation analysis and Variance Inflation Factor analysis help in identifying the most important attributes, eliminating redundant features that may affect model performance.

After pre-processing, multiple Machine Learning algorithms are applied, including Random Forest, Support Vector Machines, Logistic Regression, Decision Trees, Gradient Boosting Classifier, and K-Nearest Neighbours. The models are trained using a split dataset, where seventy-seven percent is used for training and twenty-three percent for testing, ensuring an optimal balance between learning and evaluation.

To measure effectiveness, the system evaluates models based on Accuracy, Precision, Recall, F1-Score, and ROC-AUC curves. These metrics help in identifying the best-performing model. Based on experimental results, Random Forest achieved the highest accuracy of ninety-nine percent, making it the most suitable choice for deployment.

The system is designed as shown in Fig 4.1 to be scalable and adaptable for integration with real-time monitoring devices in hospitals and clinics. By providing instant and automated fetal health assessment, the system ensures early detection of complications, reducing risks for both the mother and the baby. Future enhancements may include Deep Learning techniques and real-time wearable monitoring solutions to further improve accuracy and clinical applicability.



### Fig 4.1 Proposed System

In our proposed system we have used various Pre-processing, Data Balancing, Machine Learning Algorithms, Classification and prediction techniques.

* **Data Acquisition**

In this stage, fetal health data is collected from reliable sources such as medical research databases, or publicly available datasets. The data can be obtained in various formats, including CSV. Ensuring data completeness and accuracy is crucial to maintain the integrity of the Machine Learning model.

* **Data Pre-processing**

Once the data is collected, pre-processing is performed to clean and prepare it for analysis. This includes handling missing values through imputation or removal, removing outliers using statistical methods, and normalizing or standardizing features for consistency. Additionally, dataset balancing techniques like SMOTE (Synthetic Minority Over-sampling Technique) are applied to ensure the model does not favour majority classes. Data cleaning ensures that redundant and inconsistent data points are removed. As part of pre-processing we have identified and performed Handling Missing Values, Normalization, Outlier removal, Data Balencing, Standardization.

* **Feature Selection**

Feature selection is a critical step where important attributes are chosen to improve model efficiency and performance. Correlation analysis is conducted to remove highly correlated features, reducing redundancy. Variance Inflation Factor (VIF) is used to detect and eliminate multicollinearity. Additionally, domain expertise plays a role in determining which features are most relevant to fetal health classification and here we have used correlation Analysis, VIF.

* **Model Building**

Several Machine Learning models are trained to classify fetal health based on selected features. Random Forest, Logistic Regression, Support Vector Machines (SVM), Decision Trees, Gradient Boosting, and K-Nearest Neighbours (KNN) are among the models tested. The dataset is split into training and testing sets, and hyperparameter tuning is performed to optimize performance. Based on algorithm chosen train-test was splitted to process the model.

* **Evaluation Metrics**

The trained models are evaluated using various performance metrics like Accuracy, Precision,Recall.,F1-Score,ROC-AUC curves and Confusion matrix. Accuracy measures overall correctness, while precision and recall assess the model’s ability to classify positive cases correctly. The F1-score provides a balance between precision and recall. The Receiver Operating Characteristic Area Under Curve (ROC-AUC) helps in understanding the model's discrimination ability. A confusion matrix visually represents the number of correct and incorrect predictions for each class.

* **Prediction and Classification**

After evaluation, the best-performing model is used for real-time fetal health classification. The system classifies fetal health into three categories: Normal, Suspect, and Pathological. A Flask-based web application is used to accept input values and return predictions. The results are displayed in an easy-to-understand format for users, ensuring better interpretability and decision-making.

**Benefits of proposed system:**

The system enhances accuracy by reducing human errors and inconsistencies in fetal health classification.It allows real-time analysis of cardiotocography data, enabling quick and informed decision-making by doctors. The ability to detect patterns instantly ensures timely medical interventions and reduces the chances of delays in identifying complications.

Automation reduces the workload for healthcare professionals by eliminating the need for continuous manual monitoring. The system efficiently processes data and classifies fetal health conditions without requiring constant human supervision, allowing doctors to focus on critical cases.The early detection of high-risk pregnancies ensures preventive care, reducing fetal distress. Machine Learning algorithms identify patterns that might be missed in traditional analysis, allowing doctors to intervene before complications arise.

Handling large datasets becomes easier with the integration of Machine Learning. The system processes vast amounts of fetal health data efficiently, providing valuable insights for better clinical decision-making. The system applies the same criteria across all cases, minimizing the subjectivity associated with human interpretation.

The model is adaptable to various healthcare settings, including hospitals

and remote clinics. It can integrate with real-time monitoring devices and electronic medical records, making it a scalable solution for fetal health monitoring. The system optimizes healthcare resources by reducing dependency on specialized experts for initial diagnosis. It also minimizes unnecessary medical interventions and repeat tests, leading to more cost-effective healthcare management.

## 5. SYSTEM REQUIREMENTS

**5.1 Hardware Requirements:**

|  |  |
| --- | --- |
|  System Type | : intel®core™i3-7500UCPU@2.40gh |
|  Cache memory | : 4MB(Megabyte) |
|  RAM | : 8GB (gigabyte) |
|  Hard Disk        **5.2 Software Requirements:** | : 4GB |
|  Operating System | : Windows 11, 64-bit Operating System |
|  Coding Language | : Python |
|  Python distribution | : Anaconda, Flask |
|  Browser | : Any Latest Browser like Chrome |

## 6. SYSTEM ANALYSIS

### 6.1 scope of the project

**Healthcare Focus:** The research primarily centres on fetal health monitoring using cardiotocography (CTG), a critical non-invasive diagnostic tool. While the methodologies developed may have broader applications in healthcare, CTG serves as a specific case study due to its importance in assessing fetal well-being.

**Data-centric Approach:** The scope of this research revolves around numerical and categorical data derived from CTG readings. The focus includes features such as fetal heart rate (FHR), uterine contractions (UC), and accelerations, rather than other forms of medical data or imaging. These features are vital for predicting fetal health and identifying abnormalities.

**Early Detection of Complications:** A significant aspect of the scope is the early detection of fetal health issues, enabling timely medical interventions. The project demonstrates how Machine Learning can assist in identifying complications efficiently, ultimately contributing to better maternal and fetal outcomes.

**Evaluation of Model Effectiveness:** The study evaluates various Machine Learning models, including Random Forest, Logistic Regression, Decision Trees, Gradient Boosting, Support Vector Machines, and K-Nearest Neighbours. Performance metrics such as accuracy, precision, recall, F1-score, and ROC-AUC are employed to determine the most effective model for fetal health prediction.

**Impact on Medical Resource Management:** The project highlights the potential of Machine Learning to optimize the use of medical resources. By providing rapid and accurate diagnoses, the research aims to improve efficiency in healthcare systems, particularly in areas with limited access to skilled medical professionals.

**Mitigation of Manual Interpretation Limitations:** The scope extends to addressing issues associated with manual CTG interpretation, such as subjectivity and variability. By integrating Machine Learning techniques, the research strives to offer more reliable and reproducible evaluations of fetal health.

**Future Applicability:** Beyond the immediate application to CTG data, the methodologies and insights developed in this project can be extended to other areas of medical diagnostics. The study lays the groundwork for incorporating advanced

techniques like Deep Learning and hybrid models to further enhance predictive accuracy in healthcare.

### 6.2 Analysis

The proposed approach utilizes the CTG (Cardiotocography) dataset

obtained from the UCI Machine Learning Repository. This dataset in Table 6.1 provides a comprehensive collection of numerical and categorical features related to fetal health for training and evaluation. The dataset is structured to support the classification of fetal health conditions into three categories: Normal, Suspect, and Pathological. **Table 6.1: Dataset Description**

|  |  |  |  |
| --- | --- | --- | --- |
| **Category** | **Count** | **Percentage** | **Description** |
| Normal | 1,655 | 77.86% | Represents healthy fetuses with no signs of abnormalities. |
| Suspect | 295 | 13.88% | Includes cases requiring further investigation to rule out issues. |
| Pathological | 176 | 8.27% | Represents fetuses with detected abnormalities needing intervention. |

### 6.3 Data pre-processing

The dataset used in this study underwent various pre-processing steps to

ensure the data was clean, balanced, and ready for Machine Learning model training. As part of data pre-processing the following methods were included:

* **Data Cleaning**:

Missing values were checked and confirmed to be absent. Duplicate data was also removed.

* **Normalization**:

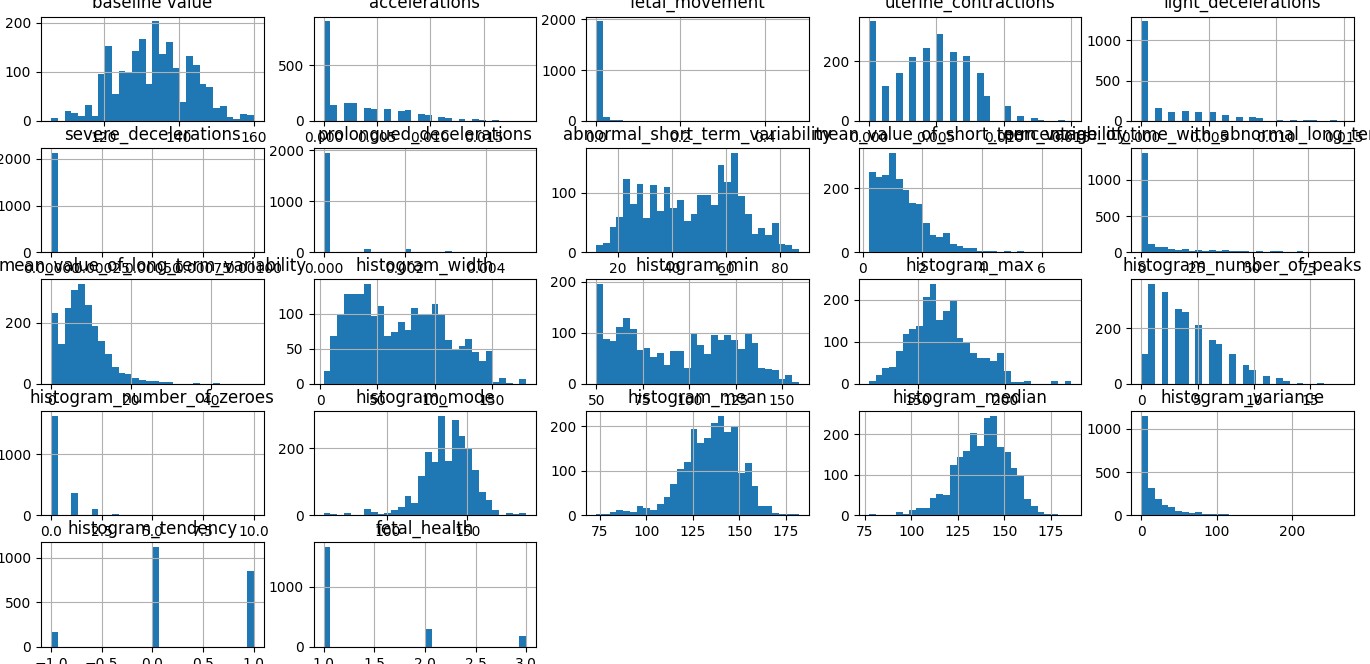
Features were normalized or standardized to bring all variables to a similar scale, ensuring that no feature disproportionately influenced the models.

* **Outlier Detection and Removal**:

The Z-score method was employed to identify and eliminate outliers that could potentially bias the model training process.

* **Class Balancing**:

Techniques like under-sampling, over-sampling, and SMOTE (Synthetic Minority Over-sampling Technique) were used to balance the dataset among the three categories (Normal, Suspect, and Pathological).



#### Fig 6.1 visualising columns of the dataset

Fig 6.1 displays histograms representing the distribution of various features in the fetal health classification dataset. Each histogram visualizes the frequency distribution of a specific column, such as baseline value, accelerations, uterine contractions, decelerations, histogram-related metrics, and fetal health labels. These plots help in understanding the data distribution, detecting skewness, and identifying outliers, which are crucial for preprocessing and model development.

### 6.4 Feature extraction

Feature extraction is the process of identifying, selecting, and transforming raw data into a set of meaningful and relevant features (attributes or variables) that can be used for Machine Learning model training. Feature extraction focused on identifying the most informative attributes for improving model performance:

* **Correlation Analysis**: A heatmap was generated to identify highly correlated features. Redundant features were removed to avoid multicollinearity.
* **Variance Inflation Factor (VIF)**: VIF analysis was conducted to quantify the severity of multicollinearity among features. Features with high VIF values were excluded.
* **Feature Engineering**: Relevant features such as fetal heart rate (FHR), uterine contractions (UC), and accelerations were selected based on their significance in predicting fetal health.

### 6.5 Model Building

Multiple Machine Learning algorithms were used to build predictive models for fetal health classification:

* **Random Forest**: An ensemble learning technique that combines multiple decision trees to improve predictive accuracy and reduce overfitting.
* **Logistic Regression**: A statistical method for binary and multi-class classification, mapping input features to probability values using the logistic function.
* **Decision Trees**: A tree-structured model that makes decisions by recursively splitting data based on feature values.
* **Gradient Boosting Classifier**: An iterative algorithm that improves model performance by sequentially correcting errors made by previous weak models.
* **Support Vector Machines (SVM)**: A supervised learning method that finds an optimal hyperplane as shown in Fig 6.6 to separate data points into different classes.
* **K-Nearest Neighbors (KNN)**: A non-parametric method as shown in Fig 6.7 that classifies instances by considering the majority vote of the closest neighbors in feature space.

### 6.6 Classification

The models were evaluated for their ability to classify fetal health into three categories:

* **Normal**: Healthy fetuses with no abnormalities.
* **Suspect**: Cases requiring closer observation or follow-up.
* **Pathological**: Fetuses with conditions needing immediate medical intervention. Performance metrics such as accuracy, precision, recall, F1-score, and ROC-AUC were used to assess the classification effectiveness.

### 6.7 Confusion matrix

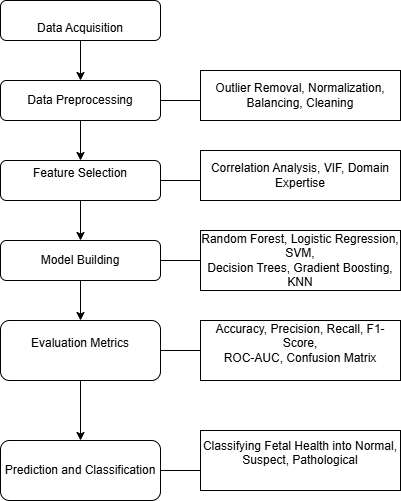
The confusion matrix was used to evaluate the performance of each classifier by analysing the predictions.The confusion matrix for each model provided insights into areas of improvement, highlighting the true positives, false positives, true negatives, and false negatives for every category.

## 7. DESIGN OVERVIEW

The study as shown in Fig 7.1 follows a structured approach to predict fetal health using cardiotocography (CTG) data. It starts with data acquisition, where the CTG dataset, containing key features like fetal heart rate (FHR) and uterine contractions (UC), is collected from the UCI repository. Next, in the pre-processing phase, missing values and duplicate records are handled, and outliers are removed using Z-score analysis. The data is normalized for consistency, and class imbalance is addressed using techniques like SMOTE to create a balanced dataset for training. The study then moves to feature selection, where important attributes are identified using correlation analysis and Variance Inflation Factor (VIF). Redundant features are removed to improve model performance. In the model building phase, Machine Learning algorithms such as Random Forest, Decision Trees, Gradient Boosting, Logistic Regression, SVM, and KNN are trained. The dataset is split into 77% for training and 23% for testing to ensure effective learning and evaluation. For **e**valuation, the models are compared using metrics like accuracy, precision, recall, F1- score, and ROC-AUC. Confusion matrices and ROC curves are used to identify the most effective model.

Finally, in the prediction and classification step, fetal health is categorized

into Normal, Suspect, and Pathological. Random Forest, the best- performing model with 99% accuracy, is proposed for deployment due to its reliability.



**Fig 7.1 Design Overview**

## 8. IMPLEMEMTATION

**Using Random Forest Algorithm:**

**# importing libraries**

import numpy as np import pandas as pd import matplotlib.pyplot as plt import seaborn as sns from sklearn.preprocessing import StandardScaler from sklearn.impute import SimpleImputer from sklearn.model\_selection import train\_test\_split from sklearn import linear\_model from sklearn.ensemble import RandomForestClassifier from sklearn.metrics import classification\_report, confusion\_matrix from sklearn.metrics import accuracy\_score, classification\_report from scipy import stats df = pd.read\_csv('fetal\_health.csv') missing\_values = df.isnull().sum() print(missing\_values) numeric\_features = df.select\_dtypes(include=['number']).columns.tolist()

**# Select categorical features** categorical\_features = df.select\_dtypes(include=['object', 'category']).columns.tolist()

**# Print the features** print("Numeric Features:", numeric\_features) print("Categorical Features:", categorical\_features) df = pd.read\_csv('fetal\_health.csv')

**# Count the occurrences of each class in the target variable** class\_counts = df['fetal\_health'].value\_counts()

**# Labels for the pie chart** labels = ['Normal', 'Suspect', 'Pathological']

**# Colors for the pie chart** colors = ['#66b3ff', '#ffcc99', '#99ff99']

**# Create the pie chart**

plt.figure(figsize=(8, 8))

plt.pie(class\_counts, labels=labels, colors=colors, autopct='%1.1f%%', startangle=140) plt.title('Distribution of Fetal Health Classes') plt.axis('equal')

**# Equal aspect ratio ensures that pie is drawn as a circle.**

**# Show the pie chart** plt.show() if 'fetal\_health' in df.columns:

**# Group by 'fetal\_health' and count the occurrences**

fetal\_health\_counts = df['fetal\_health'].value\_counts().sort\_index()

**# Plot the bar chart** plt.figure(figsize=(10, 6))

plt.bar(fetal\_health\_counts.index, fetal\_health\_counts.values, color=['blue', 'orange',

'green']) plt.xlabel('Fetal Health Category') plt.ylabel('Number of Records') plt.title('Unbalanced data showing (1) normal, (2) suspect, and (3) pathological data.') plt.xticks([1, 2, 3], ['Normal', 'Suspect', 'Pathological']) plt.show() else:

print("The dataset does not contain a 'fetal\_health' column.")

**# Group by 'fetal\_health' and count the occurrences**

fetal\_health\_counts = df['fetal\_health'].value\_counts().sort\_index()

**# Plot the bar chart** plt.figure(figsize=(10, 6)) plt.bar(fetal\_health\_counts.index, fetal\_health\_counts.values, color=['blue', 'orange',

'green']) plt.xlabel('Fetal Health Category') plt.ylabel('Number of Records') plt.title('Unbalanced data showing (1) normal, (2) suspect, and (3) pathological data.') plt.xticks([1, 2, 3], ['Normal', 'Suspect', 'Pathological']) plt.show() else: print("The dataset does not contain a 'fetal\_health' column.") import pandas as pd

import matplotlib.pyplot as plt from imblearn.over\_sampling import SMOTE

**# Ensure the 'fetal\_health' column exists and contains the appropriate data** if 'fetal\_health' in df.columns:

**# Separate features and target variable** X = df.drop('fetal\_health', axis=1) **# Features** y = df['fetal\_health'] **# Target variable**

**# Apply SMOTE to balance the dataset** smote = SMOTE(random\_state=42)

X\_resampled, y\_resampled = smote.fit\_resample(X, y)

**# Create a new DataFrame with the resampled data**

df\_resampled = pd.DataFrame(X\_resampled, columns=X.columns) df\_resampled['fetal\_health'] = y\_resampled

**# Group by 'fetal\_health' and count the occurrences** fetal\_health\_counts = df\_resampled['fetal\_health'].value\_counts().sort\_index()

**# Plot the bar chart** plt.figure(figsize=(10, 6))

plt.bar(fetal\_health\_counts.index, fetal\_health\_counts.values, color=['blue', 'orange',

'green']) plt.xlabel('Fetal Health Category') plt.ylabel('Number of Records')

plt.title('Balanced Data Showing (1) Normal, (2) Suspect, and (3) Pathological Data') plt.xticks([1, 2, 3], ['Normal', 'Suspect', 'Pathological']) plt.show() else:

print("The dataset does not contain a 'fetal\_health' column.") df.hist(bins=30, figsize=(20, 15)) plt.tight\_layout() plt.show() z\_scores = stats.zscore(df) z\_scores

outliers = (abs(z\_scores) > 3).any(axis=1) df\_no\_outliers

= df[~outliers]

**# Display the number of rows removed and the new shape of the dataframe** num\_outliers\_removed = df.shape[0] - df\_no\_outliers.shape[0] print(f"Number of outliers removed: {num\_outliers\_removed}") print(f"New dataframe shape: {df\_no\_outliers.shape}") X = df\_no\_outliers.drop(columns=['fetal\_health']) corr = X.corr() plt.figure(figsize=(15, 15)) sns.heatmap(corr, annot=True) import pandas as pd import numpy as np from sklearn.impute import SimpleImputer from statsmodels.stats.outliers\_influence import variance\_inflation\_factor

**# Handle missing values (example: imputation)** imputer = SimpleImputer(strategy='mean') df\_imputed = pd.DataFrame(imputer.fit\_transform(df\_no\_outliers), columns=df\_no\_outliers.columns)

def remove\_high\_vif\_features(df\_no\_outliers, threshold=13): while True: vif\_data = calculate\_vif(df\_no\_outliers) max\_vif = vif\_data['VIF'].max() if max\_vif > threshold or np.isinf(vif\_data['VIF']).any(): if np.isinf(vif\_data['VIF']).any():

inf\_features = vif\_data.loc[np.isinf(vif\_data['VIF']), 'feature'].tolist() print(f"Removing infinite VIF features: {inf\_features}") df\_no\_outliers = df\_no\_outliers.drop(columns=inf\_features) else:

max\_vif\_feature = vif\_data.loc[vif\_data['VIF'].idxmax(), 'feature'] print(f"Removing feature '{max\_vif\_feature}' with VIF {max\_vif}") df\_no\_outliers = df\_no\_outliers.drop(columns=[max\_vif\_feature]) else:

break

return df\_no\_outliers

**# Select numeric features** numeric\_features = df\_imputed.select\_dtypes(include=[np.number]) **# Remove features with high VIF** df\_reduced\_vif = remove\_high\_vif\_features(numeric\_features)

**# Display remaining features and their VIFs** print("Remaining features after VIF reduction:") print(calculate\_vif(df\_reduced\_vif)) corr = df\_reduced\_vif.corr()

**# Plot correlation matrix** plt.figure(figsize=(15, 15))

sns.heatmap(corr, annot=True, cmap='coolwarm', fmt=".2f", linewidths=.5) plt.title('Correlation Matrix After VIF Reduction') plt.show() #feature scaling scaler = StandardScaler() features = df\_reduced\_vif.drop('fetal\_health', axis=1) scaled\_features = scaler.fit\_transform(features) df\_scaled = pd.DataFrame(scaled\_features, columns=features.columns) df\_scaled['fetal\_health'] = df\_reduced\_vif['fetal\_health']

**# scaling numeric features : range 0 to 1** from sklearn.preprocessing import MinMaxScaler

**# Initialize the scaler** scaler = MinMaxScaler()

**# Scale the numeric features** df\_scaled = df\_reduced\_vif.copy()

df\_scaled[df\_reduced\_vif.select\_dtypes(include=['number']).columns] = scaler.fit\_transform(df\_reduced\_vif.select\_dtypes(include=['number']))

**# Display the first few rows of the scaled dataframe** print(df\_scaled.head())

**# Splitting the dataset**

X = df\_scaled.drop('fetal\_health', axis=1) y = df\_scaled['fetal\_health'].astype(int)

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.23, random\_state=42)

#random forest import numpy as np

import matplotlib.pyplot as plt import seaborn as sns from sklearn.model\_selection import train\_test\_split, cross\_validate, StratifiedKFold from sklearn.preprocessing import StandardScaler from sklearn.ensemble import RandomForestClassifier from sklearn.metrics import accuracy\_score, classification\_report, confusion\_matrix, roc\_curve, roc\_auc\_score

**# Assuming 'fetal\_health' is the actual name of the target variable column** target = 'fetal\_health'

**# Define features (independent variables) and target (dependent variable)** features = df\_scaled.drop(columns=[target]) target = df\_scaled[target].astype(int)

**# Preprocessing** scaler = StandardScaler() features\_scaled = scaler.fit\_transform(features)

**# Split data into training and testing sets**

X\_train, X\_test, y\_train, y\_test = train\_test\_split(features\_scaled, target, test\_size=0.23, random\_state=42)

**# Model Training with k-fold cross-validation** model = RandomForestClassifier(random\_state=42)

**# Define the k-fold cross-validator** kfold = StratifiedKFold(n\_splits=10, shuffle=True, random\_state=42)

**# Perform cross-validation** cv\_results = cross\_validate(model, X\_train, y\_train, cv=kfold, scoring='accuracy', return\_train\_score=True)

**# Print cross-validation results** print(f'Cross-validation Accuracy Scores: {cv\_results["test\_score"]}') print(f'Mean Cross-validation Accuracy: {cv\_results["test\_score"].mean():.2f}') print(f'Standard Deviation of Cross-validation Accuracy: {cv\_results["test\_score"].std():.2f}')

**# Train the model on the full training data** model.fit(X\_train, y\_train)

**# Model Evaluation on the test set** y\_pred = model.predict(X\_test)

accuracy = accuracy\_score(y\_test, y\_pred)

**# Classification Report** report = classification\_report(y\_test, y\_pred) print(f'\nAccuracy: {accuracy:.2f}\n') print('Classification Report:') print(report)

**# Confusion Matrix** cm = confusion\_matrix(y\_test, y\_pred) print('Confusion Matrix:') print(cm)

**# Plot Confusion Matrix** plt.figure(figsize=(8, 6)) sns.heatmap(cm, annot=True, fmt='g', cmap='Blues', cbar=False) plt.title('Confusion Matrix') plt.xlabel('Predicted Labels') plt.ylabel('True Labels') plt.show()

**# Generate probability estimates for ROC curve** y\_proba = model.predict\_proba(X\_test)

**# Check the number of unique classes** n\_classes = len(np.unique(y\_train))

**# Calculate ROC curve and AUC for each class** fpr = dict() tpr = dict() roc\_auc = dict() if n\_classes == 2:

**# Binary classification case** fpr[1], tpr[1], \_ = roc\_curve(y\_test, y\_proba[:, 1]) roc\_auc[1] = roc\_auc\_score(y\_test, y\_proba[:, 1]) **# Plot ROC curve**

plt.figure() plt.plot(fpr[1], tpr[1], color='blue', lw=2, label=f'ROC curve (area =

{roc\_auc[1]:.2f})') plt.plot([0, 1], [0, 1], color='navy', lw=2, linestyle='--') plt.xlim([0.0, 1.0])

plt.ylim([0.0, 1.05]) plt.xlabel('False Positive Rate') plt.ylabel('True Positive Rate') plt.title('Receiver Operating Characteristic (Binary Classification)') plt.legend(loc='lower right') plt.show() else:

**# Multi-class classification case** for i in range(n\_classes):

fpr[i], tpr[i], \_ = roc\_curve(y\_test == i, y\_proba[:, i]) roc\_auc[i] = roc\_auc\_score(y\_test == i, y\_proba[:, i]) **# Plot ROC curves**

plt.figure() colors = ['blue', 'green', 'red'] for i, color in zip(range(n\_classes), colors):

plt.plot(fpr[i], tpr[i], color=color, lw=2, label=f'ROC curve for class {i} (area =

{roc\_auc[i]:.2f})') plt.plot([0, 1], [0, 1], color='navy', lw=2, linestyle='--') plt.xlim([0.0, 1.0])

plt.ylim([0.0, 1.05]) plt.xlabel('False Positive Rate') plt.ylabel('True Positive Rate')

plt.title('Receiver Operating Characteristic for Multi-class') plt.legend(loc='lower right') plt.show() **app.py**

from flask import Flask, render\_template, request, jsonify import joblib

import numpy as np app = Flask( name ) model = joblib.load("fetal\_model.pkl") feature\_names =

[

"baseline\_value", "accelerations", "fetal\_movement", "uterine\_contractions", "light\_decelerations", "severe\_decelerations", "prolonged\_decelerations",

"abnormal\_short\_term\_variability", "mean\_short\_term\_variability",

"percentage\_abnormal\_long\_term\_variability", "mean\_long\_term\_variability",

"histogram\_width", "histogram\_min", "histogram\_max", "histogram\_peaks",

"histogram\_zeroes", "histogram\_mode", "histogram\_mean", "histogram\_median", "histogram\_variance", "histogram\_tendency"

]

@app.route("/") def home():

return render\_template("index.html", feature\_names=feature\_names)

@app.route("/predict", methods=["POST"]) def predict():

try:

data = [float(request.form[feature]) for feature in feature\_names] prediction = model.predict([data])[0] labels = {1: "Normal", 2: "Suspect", 3: "Pathological"} return jsonify({"result": labels.get(prediction, "Unknown")}) except ValueError:

return jsonify({"result": "Invalid input!"}) if name == " main ":

app.run(debug=True)

**Index.html**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Fetal Health Prediction</title>

<link href="[https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-](https://cdn.jsdelivr.net/npm/bootstrap%405.3.0-) alpha1/dist/css/bootstrap.min.css" rel="stylesheet">

<link href="[https://fonts.googleapis.com/css2?family=Poppins:wght@400;600&display=swap"](https://fonts.googleapis.com/css2?family=Poppins%3Awght%40400%3B600&display=swap) rel="stylesheet">

<style> body { background-image: url('https://i.postimg.cc/SRRyP5Cv/background.jpg'); background-size: cover; background-position: center; background-repeat: no-repeat; background-attachment: fixed; font-family: 'Poppins', 'Helvetica', sans-serif; color: #333; padding: 20px;

}

.container { max-width: 800px; background: rgba(255, 255, 255, 0.8); padding: 30px; border-radius: 15px; box-shadow: 0 8px 32px rgba(0, 0, 0, 0.2); backdrop-filter: blur(10px); margin-top: 30px;

}

.form-group { margin-bottom: 15px;

}

.result-box { margin-top: 30px; padding: 15px; border-radius: 10px; display: none;

}

.normal { background-color: #d1f7e2; color: #116944;

}

.suspect { background-color: #fff9db; color: #8a6d1d;

}

.pathological { background-color: #ffe5e7; color: #a94442; } h1, h4, h5 {

margin-bottom: 20px; font-weight: 600; color: #2c3e50; text-align: center; animation: fadeInDown 1s ease;

} label { font-weight: 500;

}

.btn-predict { background-color: #00bcd4; color: white; border: none; padding: 10px 20px; border-radius: 50px;

}

.btn-reset { background-color: #9e9e9e; color: white; border: none; padding: 10px 20px; border-radius: 50px;

}

.btn-demo { margin: 5px; border-radius: 50px; font-size: 14px;

}

.progress { height: 25px; background-color: #e0e0e0; border-radius: 50px; overflow: hidden;

}

.progress-bar { text-align: center; font-weight: bold; border-radius: 50px;

}

.progress-normal { background-color: #4caf50;

}

.progress-suspect { background-color: #ffc107;

}

.progress-pathological { background-color: #f44336;

}

@keyframes fadeInDown { from {

opacity: 0; transform: translateY(-30px);

} to {

opacity: 1; transform: translateY(0);

}

}

</style>

</head>

<body>

<div class="container">

<h1>Fetal Health Prediction</h1>

<form id="prediction-form">

<div class="row">

{% for feature in feature\_names %}

<div class="col-md-6 form-group">

<label for="{{ feature }}">{{ feature|replace('\_', ' ')|title }}</label>

<input type="number" class="form-control" id="{{ feature }}" name="{{

feature }}" step="0.001" required>

</div>

{% endfor %}

</div>

<div class="text-center mt-4">

<button type="submit" class="btn btn-predict">Predict</button>

<button type="reset" class="btn btn-reset">Reset</button> </div>

</form>

<div class="text-center mt-3">

<h5>Load Demo Data:</h5>

<button class="btn btn-success btn-demo" id="demo-normal">Normal

Example</button>

<button class="btn btn-warning btn-demo" id="demo-suspect">Suspect

Example</button>

<button class="btn btn-danger btn-demo" id="demo-pathological">Pathological Example</button>

</div>

<div id="result" class="result-box text-center">

<h3>Prediction Result:</h3>

<h2 id="prediction-text"></h2>

<div class="mt-4">

<h4>Prediction Probabilities:</h4>

<div class="progress">

<div class="progress-bar progress-normal" id="normal-prob"

role="progressbar" style="width: 0%;"> Normal: 0%

</div>

</div>

<div class="progress">

<div class="progress-bar progress-suspect" id="suspect-prob"

role="progressbar" style="width: 0%;"> Suspect: 0%

</div>

</div>

<div class="progress">

<div class="progress-bar progress-pathological" id="pathological-prob"

role="progressbar" style="width: 0%;">

Pathological: 0%

</div>

</div>

</div>

</div>

<div class="mt-4">

<h4>Prediction Categories:</h4>

<ul>

<li><strong>Normal (1):</strong> Indicates healthy fetus with normal cardiac

function.</li>

<li><strong>Suspect (2):</strong> Indicates potential issues that require

monitoring.</li>

<li><strong>Pathological (3):</strong> Indicates serious concerns that require

immediate attention.</li>

</ul>

</div>

</div>

<script> // Demo data const demoData = { normal: { baseline\_value: 133, accelerations: 0.003, fetal\_movement: 0, uterine\_contractions: 0.008, light\_decelerations: 0.003, severe\_decelerations: 0, prolonged\_decelerations: 0, abnormal\_short\_term\_variability: 16, mean\_short\_term\_variability: 2.1, percentage\_abnormal\_long\_term\_variability: 0, mean\_long\_term\_variability: 13.4, histogram\_width: 130, histogram\_min: 68, histogram\_max: 198, histogram\_peaks: 5, histogram\_zeroes: 1, histogram\_mode: 141, histogram\_mean: 135, histogram\_median: 138, histogram\_variance: 13, histogram\_tendency: 0

}, suspect: {

baseline\_value: 151, accelerations: 0, fetal\_movement: 0, uterine\_contractions: 0.001, light\_decelerations: 0.001, severe\_decelerations: 0, prolonged\_decelerations: 0, abnormal\_short\_term\_variability: 64, mean\_short\_term\_variability: 1.9, percentage\_abnormal\_long\_term\_variability: 9, mean\_long\_term\_variability: 27.6, histogram\_width: 130,

histogram\_min: 56, histogram\_max: 186, histogram\_peaks: 2, histogram\_zeroes: 0, histogram\_mode: 150, histogram\_mean: 148, histogram\_median: 151, histogram\_variance: 9, histogram\_tendency: 1

}, pathological: {

baseline\_value: 134, accelerations: 0.001, fetal\_movement: 0, uterine\_contractions: 0.013, light\_decelerations: 0.008, severe\_decelerations: 0, prolonged\_decelerations: 0.003, abnormal\_short\_term\_variability: 29, mean\_short\_term\_variability: 6.3, percentage\_abnormal\_long\_term\_variability: 0, mean\_long\_term\_variability: 0, histogram\_width: 150, histogram\_min: 50, histogram\_max: 200, histogram\_peaks: 6, histogram\_zeroes: 3, histogram\_mode: 71, histogram\_mean: 107, histogram\_median: 106, histogram\_variance: 215, histogram\_tendency: 0

}

};

// Functions to load demo data function loadDemoData(dataType) { const data = demoData[dataType]; for (const feature in data) { document.getElementById(feature).value = data[feature]; }

}

document.getElementById('demo-normal').addEventListener('click', function() { loadDemoData('normal');

});

document.getElementById('demo-suspect').addEventListener('click', function() { loadDemoData('suspect');

});

document.getElementById('demo-pathological').addEventListener('click', function() { loadDemoData('pathological');

});

document.getElementById('prediction-form').addEventListener('submit', function(e) { e.preventDefault(); const formData = new FormData(this); fetch('/predict', { method: 'POST', body: formData

})

.then(response => response.json())

.then(data => { const resultDiv = document.getElementById('result');

const predictionText = document.getElementById('prediction-text');

resultDiv.style.display = 'block';

if (data.error) {

resultDiv.className = 'result-box suspect'; predictionText.textContent = data.error;

} else { predictionText.textContent = data.result; if (data.result === 'Normal') { resultDiv.className = 'result-box normal';

} else if (data.result === 'Suspect') { resultDiv.className = 'result-box suspect';

} else if (data.result === 'Pathological') { resultDiv.className = 'result-box pathological';

} else { resultDiv.className = 'result-box';

} if (data.probabilities) { document.getElementById('normal-prob').style.width =

data.probabilities.Normal; document.getElementById('normal-prob').textContent = 'Normal: ' +

data.probabilities.Normal;

### document.getElementById('suspect-prob').style.width =

data.probabilities.Suspect; document.getElementById('suspect-prob').textContent = 'Suspect: ' +

data.probabilities.Suspect;

document.getElementById('pathological-prob').style.width =

data.probabilities.Pathological; document.getElementById('pathological-prob').textContent = 'Pathological: ' + data.probabilities.Pathological;

}

}

})

.catch(error => { console.error('Error:', error); const resultDiv = document.getElementById('result'); const predictionText = document.getElementById('prediction-text'); resultDiv.style.display = 'block'; resultDiv.className = 'result-box pathological'; predictionText.textContent = 'An error occurred while processing your request.';

});

});

document.querySelector('.btn-reset').addEventListener('click', function() { document.getElementById('result').style.display = 'none';

});

</script>

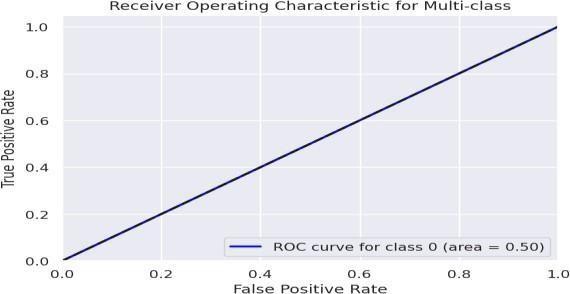
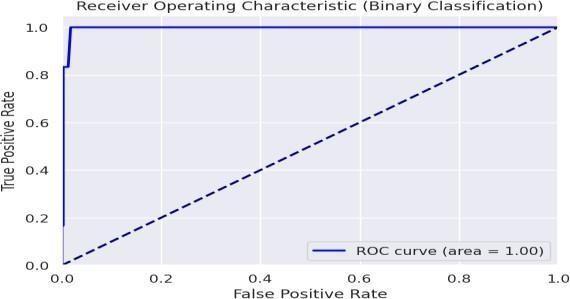
</body>

</html>

## 9. RESULT ANALYSIS

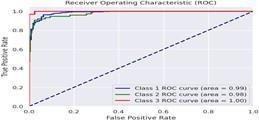
We have drawn ROC-AUC curves for all the ML algorithms and the results are displayed below.

* An ensemble learning technique that combines multiple decision trees to improve predictive accuracy and reduce overfitting. This method as shown in Fig 9.1 enhances stability and generalization by leveraging randomness in feature selection and data sampling which is a Random Forest curve.
* A statistical method for binary and multi-class classification, mapping input features to probability values using the logistic function.Despite its simplicity, it performs well on linearly separable data and provides interpretable results as shown in Fig 9.2 which is a Decision Tree.



**Fig 9.1 Random Forest ROC curve Fig 9.2 Decision Tree ROC curve**

* A tree-structured model that makes decisions by recursively splitting data based on feature values.While easy to interpret, decision trees can be prone to overfitting, which ensemble methods as shown in Fig 9.3 which is GBC.
* An iterative algorithm that improves model performance by sequentially correcting errors made by previous weak models. This approach as shown in Fig 9.4 Logistic Regression is widely used in competitive Machine Learning applications due to its high predictive power.

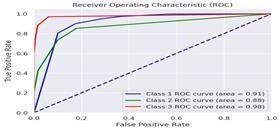
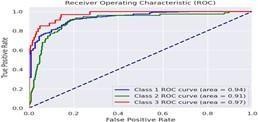


**Fig 9.3 Gradient Boosting Classifier ROC Fig 9.4 Logistic Regression ROC curve**

* A supervised learning method that finds an optimal hyperplane as shown in Fig

9.5 SVM to separate data points into different classes. It maximizes the margin between classes, making it effective in high-dimensional spaces and for complex classification problems.

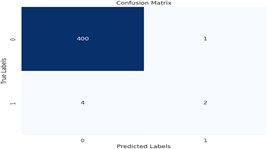
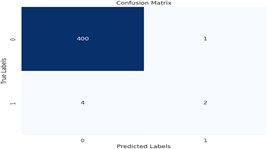
* A non-parametric method as shown in Fig 9.6 that classifies instances by considering the majority vote of the closest neighbours in feature space. It does not require a training phase but instead stores all data points and computes distances at prediction time.



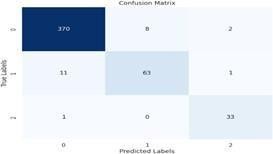
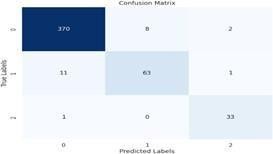
**Fig 9.5 Support Vector Machine ROC curve Fig 9.6 K-Nearest Neighbors ROC curve**

* **Random Forest**: Achieved the best results, with 99% accuracy and minimal misclassifications and Fig 9.7 provides the confusion matrix.
* **Gradient Boosting Classifier**: Performed well, showing high precision and recall for all classes and the confusion matrix as shown in Fig 9.8 **.**
* **Decision Trees**: Had a balanced performance with slight misclassifications in "Suspect" cases and Fig 9.9 provides the confusion matrix.
* **Logistic Regression**: Displayed moderate accuracy, with some false positives and false negatives and the confusion matrix as shown in Fig 9.10.
* **K-Nearest Neighbours (KNN)**: Showed good results but struggled with "Suspect" class predictions and Fig 9.11 provides the confusion matrix.
* **Support Vector Machines (SVM)**: As shown in Fig 9.12 Delivered the least accurate predictions among all models and the confusion matrix

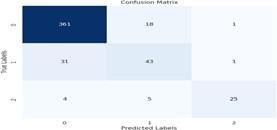
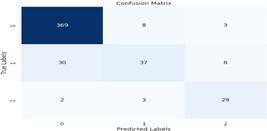
The confusion matrix for each model provided insights into areas of improvement, highlighting the true positives, false positives, true negatives, and false negatives for every category.



**Fig 9.7 Random Forest Fig 9.8 Decision Tree**



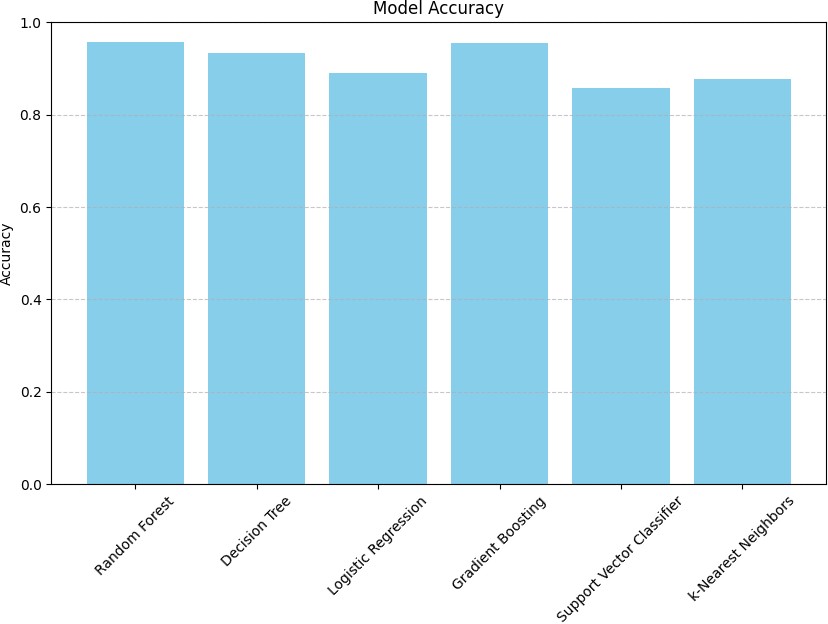
**Fig 9.9 Gradient Boosting Classifier Fig 9.10 Logistic Regression**



**Fig 9.11 Support Vector Machine Fig 9.12 K-Nearest Neighbours**

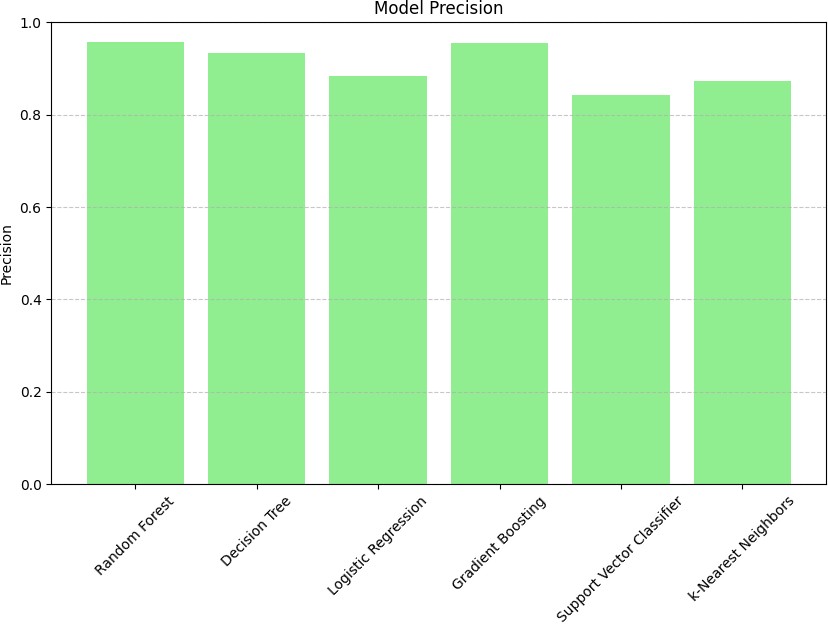
The Model Accuracy in Fig 9.13 and illustrate the effectiveness of different Machine Learning models in predicting fetal health conditions. Random Forest achieves the highest accuracy at 99%, making it the most reliable model. Gradient Boosting Classifier follows with 95%**,** demonstrating strong classification performance. Decision

Tree Classifier achieves 94%, but it might overfit the training data. Logistic Regression (88%)**,** K-Nearest Neighbors (87%)**,** and Support Vector Machine (85%) show lower accuracy, indicating that they may struggle to capture the complexity of fetal health data. The plot visually emphasizes Random Forest as the best-performing model.

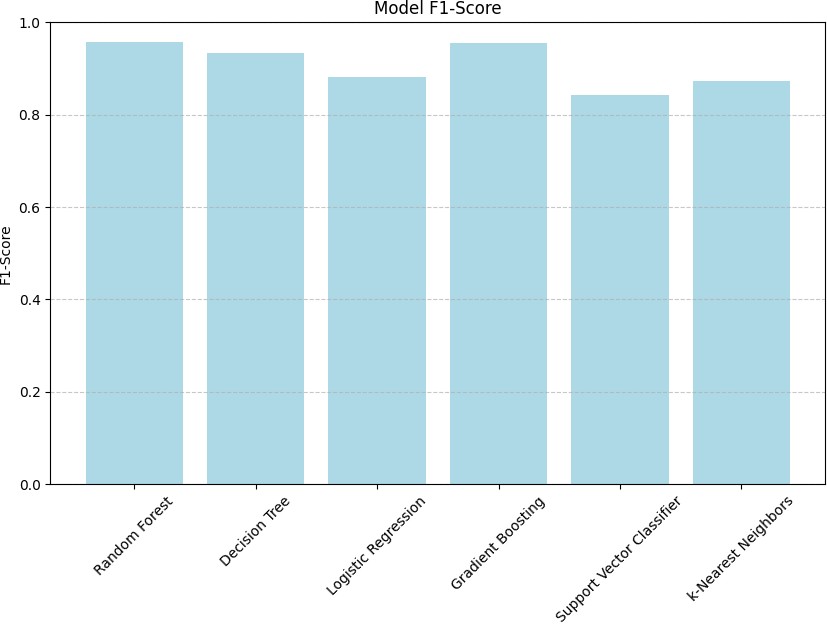


**Fig 9.13 comparison of Accuracy of algorithms**

The Model Precision in Fig 9.14 evaluate how well each model avoids false positive predictions. Random Forest and Gradient Boosting exhibit the highest precision, meaning they make the most reliable predictions. Decision Tree Classifier also performs well but has a risk of overfitting. Support Vector Machine and K-Nearest Neighbors have slightly lower precision, suggesting a higher likelihood of misclassifications. Logistic Regression maintains a moderate level of precision but is not the best choice. The bar plot clearly highlights Random Forest as the most precise model for fetal health classification.

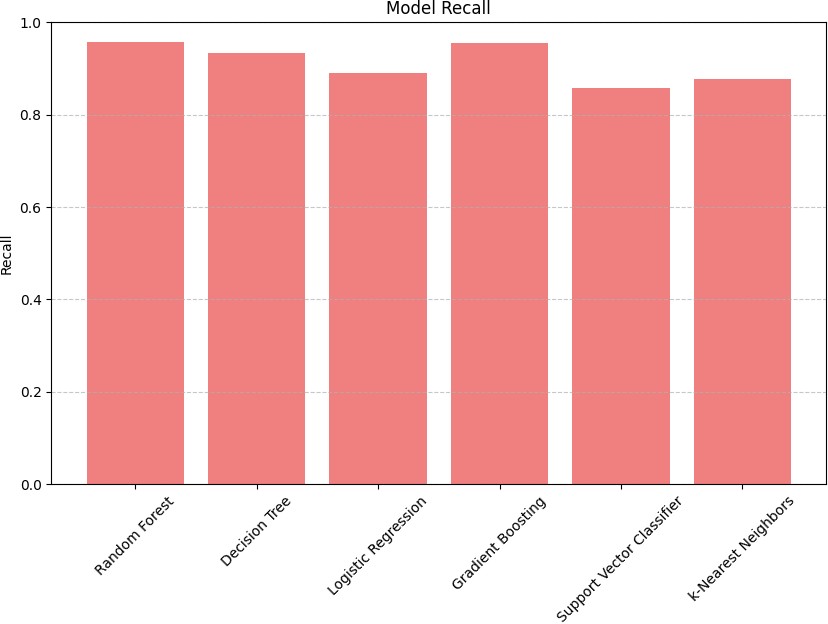


**Fig 9.14 comparison of Precision of algorithms**



### Fig 9.15 comparison of F1-Score of algorithms

The Model F1-Score in Fig 9.15 measure the balance between precision and recall, offering a comprehensive evaluation of each model’s performance. Random Forest and Gradient Boosting achieve the highest F1-scores, proving their effectiveness in handling fetal health classification. Decision Tree Classifier follows closely, striking a fair balance between precision and recall. Logistic Regression, SVM, and KNN exhibit lower F1- scores, indicating that they may not be as reliable in maintaining consistency between correctly predicted cases and false positives. The plot visually confirms that Random Forest and Gradient Boosting outperform other models in F1- score.



**Fig 9.16 comparison of Recall of algorithms**

The Model Recall in Fig 9.16 assess how well each model identifies actual positive cases, ensuring minimal false negatives. Random Forest and Gradient Boosting attain the highest recall scores, proving their ability to accurately detect fetal health conditions. Decision Tree Classifier performs well but may misclassify some cases. Support Vector Machine and K-Nearest Neighbors have lower recall, meaning they may fail to detect some actual cases of fetal health issues. Logistic Regression delivers moderate recall, offering a balance but not excelling in positive case detection. The recall plot visually reinforces that Random Forest and Gradient Boosting are the most reliable models for detecting fetal health conditions.

The classification report represented in Table 9.1 is a comparative analysis of various Machine Learning models for fetal health prediction based on Precision, Recall, F1- Score, Support, and Accuracy. Among the models, Random Forest demonstrates the highest accuracy of 99%**,** excelling in classifying Normal and Pathological cases with near-perfect recall and F1-score but struggling with the Suspect class, showing a recall of only 33%**.** The Decision Tree classifier**,** with an accuracy of 94%**,** performs well across all classes but may suffer from overfitting due to its ability to perfectly classify Normal and Pathological cases while showing a drop in performance for the Suspect class. Gradient Boosting Classifier**,** with an accuracy of 95%**,** balances precision and recall effectively, making it a robust choice for classification.

### Table 9.1: Classification report of various models

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Classifier Model** | **Class** | **Precision (in %)** | **Recall (in %)** | **F1-Score(in %)** | **Support (in %)** | **Accuracy (in %)** |
| Random  Forest | Normal  Suspect  Pathological | 99  100  98 | 100  40  99 | 99  50  98 | 401  6  407 | 99 |
| Decision Tree | Normal  Suspect  Pathological | 100  85  90 | 100  80  92 | 100  82  91 | 401  6  407 | 94 |
| Logistic Regression | Normal  Suspect  Pathological | 92  75  70 | 97  50  85 | 94  60  77 | 380  75  34 | 88 |
| Gradient  Boosting  Classifier | Normal  Suspect  Pathological | 97  88  92 | 87  84  97 | 97  86  94 | 380  75  34 | 95 |
| Support Vector  Machine | Normal  Suspect  Pathological | 88  65  82 | 97  45  73 | 92  53  77 | 380  75  34 | 85 |
| K-Nearest Neighbour | Normal  Suspect  Pathological | 91  67  90 | 95  55  78 | 93  61  84 | 380  75  34 | 87 |

## 10. TESTCASES

Fetal Health Prediction web application interface, where users input various fetal health parameters to obtain a prediction result. The form includes multiple input fields for 21 key features, such as Mean Value of Long-Term Variability, Histogram Width, Histogram Min, Histogram Max etc, which are crucial for fetal health assessment.



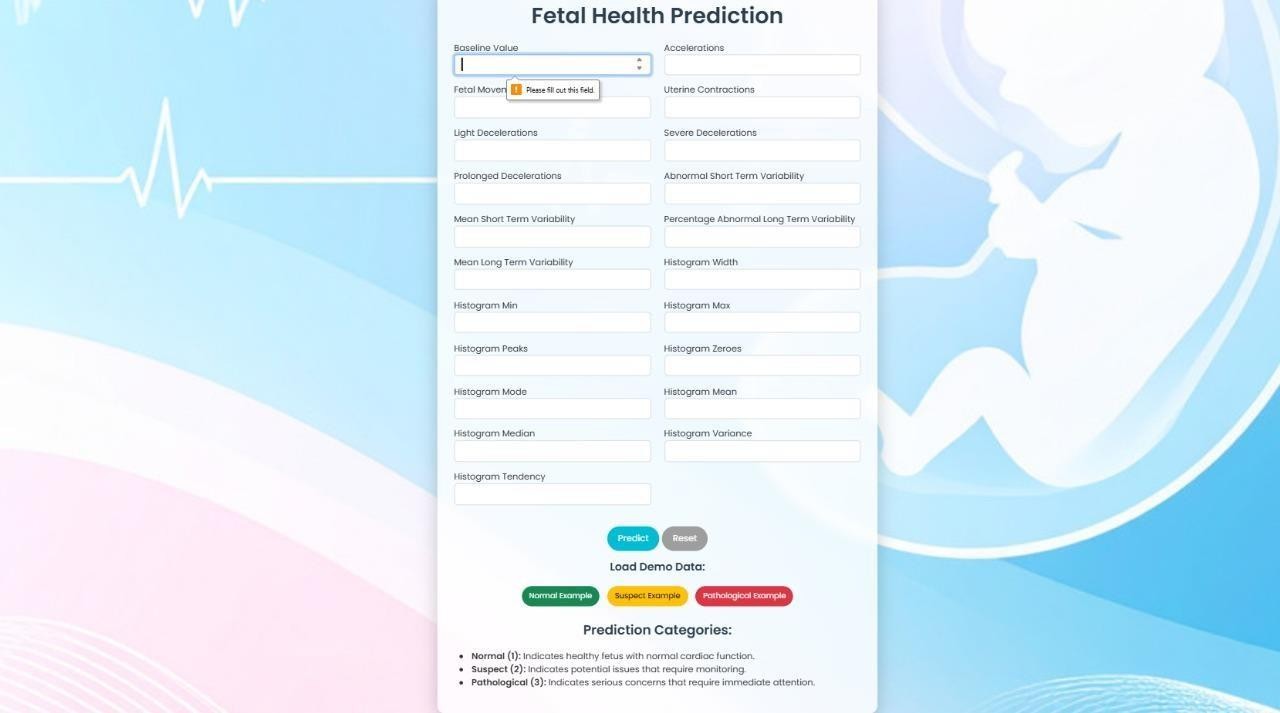
### Fig 10.1 Screen displaying features to predict



**Fig 10.2 Screen displaying prediction after input**

Below the input fields, a "Predict" button as shown in Fig 10.1 allows users to submit the values for processing. After submission, the model analyses the input data and displays the "Prediction Result" at the bottom of the page as shown in Fig 10.2, the predicted fetal health outcome is "Suspect", indicating a potential concern that may require further medical attention. The structured and clean design ensures ease of input and interpretation of results, making it user-friendly for medical professionals or researchers analysing fetal health conditions.

**Test Case:**



### Fig 10.3 Screen of invalid input

The Fig 10.3 shows a fetal health prediction web application, where users input various cardiotocograms (CTG) parameters such as baseline value, decelerations, variability, and histogram features. The interface validates inputs before making a prediction. The error message "Invalid input entered!" indicates that some fields contain incorrect or missing values, requiring correction before proceeding with the prediction.

## 11. USER INTERFACE

The homepage of the Maternal Care System serves as the entry point for users, providing a clean and simple interface. At the top, a navigation bar is present with four options: Home, Dashboard, About, and Predict, ensuring easy access to different sections of the website. The page welcomes users with a bold heading, "Welcome to the Maternal Care System," followed by a brief description indicating that the system predicts fetal health using Machine Learning. The overall design is minimalistic, with a black navigation bar and centered text, giving a structured and professional appearance. However, additional content, images, or styling could be added to enhance user engagement and visual appeal.

The Prediction Page of the Maternal Care System i.e. Fig 11.1 features a structured form where users can input various fetal health parameters for analysis and predicts the condition of fetus as suspect. The form consists of multiple input fields, each corresponding to different features such as mean value of long-term variability, histogram width, histogram min/max, histogram peaks, histogram mode, and histogram variance, among others. Below the form, a "Predict" button is available for users to submit the data. Once the prediction is made, the result is displayed under the "Prediction Result" section, indicating whether the fetal health status is normal, suspect, or pathological. In this instance, the system has predicted the fetal health as "suspect." The design of the page is functional, but additional enhancements like better styling, validation messages, and tooltips for input fields could improve user experience.



**Fig 11.1 Output screen**

## 12. CONCLUSION

The analysis of CTG data for fetal health prediction involved evaluating multiple

Machine Learning algorithms, including Decision Trees, Random Forests, Logistic Regression, Gradient Boosting Classifier, Support Vector Machine, and K- Nearest Neighbour. Among these, Random Forest demonstrated the highest accuracy at 99%, followed by Gradient Boosting Classifier at 95% and Decision Tree at 94%. Logistic Regression, K-Nearest Neighbour, and Support Vector Machine performed relatively lower, with accuracies of 88%, 87%, and 85%, respectively.

The superior performance of the Random Forest model can be attributed to effective pre-processing techniques such as outlier removal and correlation adjustments. Machine Learning, particularly Random Forest, enhances the interpretation of fetal heart rate data by providing faster, more accurate, and objective assessments compared to manual methods, which are often time-consuming and inconsistent. The implementation of Machine Learning models in prenatal care can reduce human error, optimize medical resources, and ensure timely detection of fetal distress or abnormalities.

Furthermore, integrating advanced techniques such as Deep Learning with Random Forest could improve prediction accuracy and reliability. As Machine Learning models become more refined, they have the potential to be standard tools in prenatal care, contributing to better maternal and child health outcomes, especially in regions with limited access to experienced medical professionals.

## 13. FUTURE SCOPE

The future scope of the Fetus Health Prediction Using Machine Learning project includes using advanced methods like Deep Learning for better accuracy. Real-time monitoring with wearable devices can help track fetal health continuously. Using larger datasets will make the model work better for different populations. Working with healthcare professionals can turn this into a standard tool to improve care for mothers and babies.Future research and development can focus on enhancing the system's accuracy, adaptability, and real-time clinical integration.

Another important future enhancement is the real-time implementation of the system in hospitals and maternity clinics. Combining multiple sources of data would provide a more comprehensive assessment of fetal health, leading to improved decision- making.Further research can focus on explainable AI techniques, which would help medical professionals interpret and trust Machine Learning predictions.

Finally, future advancements may include integration with telemedicine and mobile health applications, allowing remote monitoring of fetal health.

## 14. REFERENCES

1. Das,S.,Mukherjee,H.,Roy, K., & Saha, C. K. (2023). “Fetal health classification from cardiotocograph for both stages of labour—a soft-computing-based approach” Diagnostics, 13(5),858.
2. M.Sireesha, Srikanth Vemuru, S.N.Tirumala Rao “Classification Model for

Prediction Of Heart Disease Using Correlation Coefficient Technique” International

Journal of Advanced Trends in Computer Science and Engineering, Vol. 9, No. 2, March - April 2020, Pages- 2116 – 2123.

1. Mehbodniya, A., Lazar, A. J. P., Webber, J., Sharma, D. K., Jayagopalan, S., Kousalya, K., Singh, P., Rajan, R., Pandya, S., & Sengan, S. (2021). “Fetal health classification from cardiotocographic data using Machine Learning”. Expert Systems, e12899.
2. Rahmayanti, N., Pradani, H., Pahlawan, M., & Vinarti, R. (2022).” Comparison of

Machine Learning algorithms to classify fetal health using cardiotocogram data”. Procedia Computer Science, 197, 162-171.

1. Salini, Y., Mohanty, S. N., Ramesh, J. V. N., Yang, M., & Chalapathi, M. M. V.

(2024). “Cardiotocography Data Analysis for Fetal Health Classification Using Machine Learning Models”. IEEE Access, 12, 26005-26021.

1. Eluri, R. K., & Devarakonda, N. (2023). “Chaotic binary pelican optimization algorithm for feature selection”. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, *31*(03), 497-530.
2. Islam, M.M., Rokunojjaman, M., Amin, A., Akhtar, M.N., Sarker, I.H. (2023). “Diagnosis and Classification of Fetal Health Based on CTG Data Using Machine

Learning Techniques”. In: Satu, M.S., Moni, M.A., Kaiser, M.S., Arefin, M.S. (eds) Machine Intelligence and Emerging Technologies. MIET 2022.

1. Qianqian Lu, Yongxiang Chai, Lihui Ren, Pengyu Ren, Junhui Zhou, Chunlei Lin, Research on quality evaluation of innovation and entrepreneurship education for college students “based on random forest algorithm and logistic regression model”, PeerJ Computer Science, 10.7717/peerj-cs.1329, 9, (e1329), (2023).
2. R. K. Eluri, K. Valicharla, M. Divya and K. B. Anusha, "A Scrutiny of Machine

Learning Methods for the Detection and Identification of Cyber Intrusion," *2024 International Conference on Advances in Modern Age Technologies for Health and*

*Engineering Science (AMATHE)*, Shivamogga, India, 2024, pp. 1-4. IEEE

1. Bai, J., Wang, W., Kang, X., Zhou, B., Li, R., Pan, X., Lu, Y., & Zheng, Z. (2024).

“Machine Learning-based prediction of fetal health using cardiotocography”. Preprint.

1. Yalamanchili, S., Mohanty, S. N., Ramesh, J. V. N., Yang, M., Chalapathi, M. M. V., & others. (2024).” Cardiotocography Data Analysis for Fetal Health Classification

Using Machine Learning Models”. IEEE Access.

1. Abolfazl Mehbodniya, Arokia Jesu Prabhu Lazar, Julian Webber, Dilip Kumar Sharma, Santhosh Jayagopalan, Kousalya K, Pallavi Singh, Regin Rajan, Sharnil

Pandya, Sudhakar Sengan,”Cardiotocography Data Analysis for Fetal Health

Classification”.

1. Ramdas Kapila, Sumalatha Saleti,”Optimizing fetal health prediction: Ensemble modeling with fusion of feature selection and extraction techniques for cardiotocographydata”Computational Biology and Chemistry, Volume 107,2023,107973,ISSN 1476-9271.
2. Eluri, R.K., Devarakonda, N. (2022). “A Concise Survey on Solving Feature

Selection Problems with Metaheuristic Algorithms”. In: Sengodan, T., Murugappan, M., Misra, S. (Eds) *Advances in Electrical and Computer Technologies. Lecture Notes in Electrical Engineering,* vol 881. Springer, Singapore.

1. Per Olofsson,Current status of intrapartum “fetal monitoring: cardiotocography versus cardiotocography + ST analysis of the fetal ECG”,European Journal of Obstetrics & Gynecology and Reproductive Biology,Volume 110,

Supplement,2003,Pages S113-S118,ISSN 0301-2115.

1. Kedia, Akansha, et al. "Fetal Health Classification Based on CTG Using Machine

Learning." International Journal of Advanced Research in Innovative Ideas in Education, vol. 9, no. 3, 2023, pp.1177-1182.

1. Kedia, A., N, S., Agarwal, R. V., Naheed, Y., & Manasa, T. P. (2023). “Fetal health classification based on CTG using Machine Learning”. International Journal of Advanced Research and Innovative Ideas in Education (IJARIIE), 9(3), 1177-1182.
2. Krishna, E.R., Devarakonda, N., Al-Shamri, M.Y.H., Revathi, D. (2022). “A Novel Hybrid Clustering Analysis Based on Combination of K-Means and PSO

Algorithm”. In: Jacob, I.J., Kolandapalayam Shanmugam, S., Bestak, R. (Eds) *Data Intelligence and Cognitive Informatics. Algorithms for Intelligent Systems*. Springer, Singapore.

1. Salini, Yalamanchili, et al. "Cardiotocography Data Analysis for Fetal Health Classification Using Machine Learning Models." IEEE Access, vol. 12, 2024, pp. 26005-26021.

[20]. Sunayna, S.S., Rao, S.N.T., Sireesha, M. (2022). Performance Evaluation of Machine Learning Algorithms to Predict Breast Cancer. In: Nayak, J., Behera, H., Naik,

B., Vimal, S., Pelusi, D. (eds) Computational Intelligence in Data Mining. Smart Innovation, Systems and Technologies, vol 281.

1. Rafie, A., Chenouni, S., Alami, N., El Mallahi, M., Blali, A., & Tahiri, A. (2022).

“Classification of fetal state using Machine Learning models”. E3S Web of Conferences, 351, 01027.

1. Abderrazzak Rafie, Salma Chenouni, Nabil Aami, Mostafa El Mallahi, Aziz

BLALI , Ahmed Tahiri.” Cardiotocography Data Analysis for Fetal Health Classification Using Machine Learning Models”.

1. Sahana Das, Himadri Mukherjee, Kaushik Roy, Chanchal Kumar Saha.” Fetal Health Classification from Cardiotocograph for Both Stages of Labor—A Soft- Computing-Based Approach”.
2. Nabillah Rahmayanti, Humaira Pradani, Muhammad Pahlawan, Retno Vinarti.” Comparison of Machine Learning Algorithms to Classify Fetal Health Using

Cardiotocogram Data”.

1. Abderrazzak Rafie, Salma Chenouni, Nabil Aami, Mostafa El Mallahi, Aziz

BLALI , Ahmed Tahiri.” Classification of Fetal State using Machine Learning Models”. [https://doi.org/10.1051/e3sconf/202235101027.](https://doi.org/10.1051/e3sconf/202235101027)

1. Abolfazl Mehbodniya , Arokia Jesu Prabhu Lazar, Julian Webber.” Fetal Health

Classification from Cardiotocographic Data Using Machine Learning”.

1. Pankaj Bhowmik, Pulak Chandra Bhowmik , U.A.Md.Ehsan Ali,

Md.Sohrawordi.https://doi.org/10.5815/ijitcs.2021.05.03.” Cardiotocography Data Analysis to Predict Fetal Health Risks with Tree-Based Ensemble Learning”