EMBRYO GROWTH ANALYSIS USING ALGORITHMIC APPROACH

Submitted in partial fulfillment of the
requirements for the award of
Bachelor of Engineering degree in Computer Science and Engineering

By

Jinugu Nikhil Kumar (Reg.No - 39110696)

Nehal Veeramachaneni (Reg.No - 39110691)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

SCHOOL OF COMPUTING

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
Accredited with Grade "A" by NAAC

JEPPIAAR NAGAR, RAJIV GANDHISALAI,

CHENNAI – 600119

April – 2023

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited with ¯A‖ grade by NAAC

Jeppiaar Nagar, Rajiv Gandhi Salai, Chennai – 600 119

www.sathyabama.ac.in

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **Jinugu Nikhil Kumar (39110696) and Nehal Veeramachaneni (39110691)** who carried out the Project Phase-2 entitled **"EMBRYO GROWTH ANALYSIS USING ALGORITHMIC APPROACH"** under my supervision from Jan 2023 to April 2023.

Internal Guide
Dr. N. SRINIVASAN, M.E, Ph. D.

Head of the Department
Dr. L. LAKSHMANAN, M.E., Ph.D.

Submitted for Viva voce Examination held on 20.04.2023

InternalExaminer **ExternalExaminer**

DECLARATION

I, **Jinugu Nikhil Kumar (39110696),** hereby declare that the Project Phase-2 Report entitled" **EMBRYO GROWTH ANALYSIS USING ALGORITHMIC APPROACH"** done by me under the guidance of **Dr. N. SRINIVASAN, M.E, Ph. D.,** is submitted in partial fulfilment of the requirements for the award of Bachelor of Engineering degree in **Computer Science and Engineering**.

DATE: 20/04/2023

PLACE: Chennai

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management** of **SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T.Sasikala M.E., Ph. D**, **Dean**, School of Computing, **Dr. L. Lakshmanan M.E., Ph.D.,** Head of the Department of Computer Science and Engineering for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **DR.N. SRINIVASAN**, **M.E, Ph. D.,** for his valuable guidance, suggestions and constant encouragement paved way for the successful completion of my phase-1 project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Computer Science and Engineering** who were helpful in many ways for the completion of the project.

ABSTRACT

A successful pregnancy is dependent on normal embryo growth, which also affects the long-term wellbeing of the unborn child. Nevertheless, it has long been difficult in both clinical practise and research to distinguish between normal and abnormal prenatal growth. Various sources and standards that are frequently used to assess prenatal growth are discussed, as well as some of the common mistakes made by existing definitions of aberrant embryo growth. The advantages and disadvantages of various methods for modifying embryo growth standards are discussed. We also go through recent developments toward a comprehensive description of embryo growth restriction. A definition like this might take into account factors like genetics, biophysical results, maternal and embryo Doppler velocimetry, embryo health status, and embryo growth. The idea of an integrated definition seems intriguing, but more research and testing are needed. Both research and clinical practise should benefit from a more precise characterization of aberrant embryo growth. Normal embryo growth is a critical component of a healthy pregnancy and influences the long-term health of the offspring. However, defining normal and abnormal embryo growth has been a long-standing challenge in clinical practice and research. The authors review various references and standards that are widely used to evaluate embryo growth, and discuss common pitfalls of current definitions of abnormal embryo growth. Pros and cons of different approaches to customize embryo growth standards are described. The authors further discuss recent advances towards an integrated definition for embryo growth restriction. Such a definition may incorporate embryo size with the status of placental health measured by maternal and embryo Doppler velocimetry and biomarkers, biophysical findings and genetics. Although the concept of an integrated definition appears promising, further development and testing are required. An improved definition of abnormal embryo growth should benefit both research and clinical practice.

TABLE OF CONTENTS

List of Figures

CHAPTER 1

INTRODUCTION

1.1 OVERVIEW

The healthcare sector is significant and has a global impact on people's lives. Recent medical developments have increased life expectancy and lowered the number of fatalities from conditions like cancer and heart disease. There is growing proof that routine screening and prompt treatment can stop serious illness or death. However, many components of health care are overlooked due to insufficient healthcare, especially in many low-income nations, frequently resulting in repeated deaths. Over the past few decades, artificial intelligence has advanced quickly and been successfully applied to a number of important fields, including finance, governance, and more. The use of machine learning techniques in healthcare is widespread, particularly in the detection and avertance of deaths that could be avoided. Machine learning is being used in a number of medical applications, such as embryo growth and anomalies, computer vision, cancer diagnosis, and medical picture analysis.

In this project, embryo anomalies are discussed, and an improved machine learning classification approach is suggested for early identification and prediction.

The advancement of machine learning strategies and image classification techniques has greatly increased the ability to detect abnormalities in the first trimester of pregnancy. Recent studies show that 3-5% of all of her pregnancies have embryo abnormalities. The second trimester is the recommended period of care in India for embryo anatomy assessments (between 18 and 22weeks). Recent research has demonstrated appreciable advancements in the detection and prognosis of embryo anomalies during the first trimester of pregnancy. Employing machine learning techniques that scientists use to detect and forecast embryo problems early on.

1.2 MACHINE LEARNING

Computer science may one day include the study of machine learning (AI). Machine learning seeks to grasp the structure of knowledge and match the data to models that people can use and comprehend. Despite being a branch of technology, machine learning is not the same as conventional process approaches. An algorithm is a set of deliberately specified instructions that a computer employs to do calculations or fix faults in the traditional computing environment. Instead, using mathematical analysis on output values that are inside a given range, machine learning algorithms enable computers to direct knowledge inputs. This is made possible by machine learning, which also enables computers to adjust decision-making processes based on knowledge inputs and easily construct models using pattern information.

1.3 Machine Learning Strategies

Tasks are often categorized into broad types in machine learning. How the lesson is accepted or how the created system provides feedback on the teacher supports these classes. Two of the most popular machine learning techniques are feeding tagged data to algorithm programmers and supervised learning, which supports algorithms with exemplary input and output data that has been annotated by humans. not learning without supervision.

1.3.1 Supervised Learning

In supervised learning, a computer is provided with model inputs and squares that are labelled with intended results. By comparing the "learned" output to the "real" output, the algorithm software is supposed to "learn" in order to identify flaws and adjust the model as necessary. As a result, supervised learning makes use of patterns to forecast label values for new unlabeled data. For instance, in supervised learning, data from photographs of sharks categorized as fish and images of the ocean labelled as water can be input to an algorithm. A supervised learning algorithm will be able to recognise unlabeled shark images as fish and unlabeled ocean images as water by training with this data.

Statistically likely future events can be predicted using historical data, which is a typical use for supervised learning. It predicts future stock market moves and filters spam emails using historical stock market data. Labeled dog pictures are frequently used as an input file in supervised learning in order to categorize unlabeled dog pictures.

CHAPTER 2

LITERATURE REVIEW

2.1 RELATED WORK

- Lesley M McCowan, Francesc Figueras, Ngaire H Anderson in [1] summarizes areas of consensus and controversy between recently published national guidelines on small for gestational age or embryo growth restriction; highlight any recent evidence that should be incorporated into existing guidelines; and identify future research priorities in this field.
- Michal Rabijewski in [2] does the assessment of embryo growth has an important effect on perinatal morbidity and mortality. To understand what tool to choose best for a given population a basic knowledge of how growth charts are developed and used has to be acquired. For this reason, this literature review was performed.
- K S Joseph, John Fahey, Robert W Platt, Robert M analyses that birth weight range was identified, at each gestational age, over which serious neonatal morbidity and neonatal mortality rates were lowest. Among singleton males at 40 weeks, serious neonatal morbidity/mortality rates were lowest between 3,012 g (95% confidence interval (CI): 3,008, 3,018) and 3,978 g (95% CI: 3,976, 3,980). The low end of this optimal birth weight range for females was 37 g (95% CI: 21, 53) less.

2.2 OPEN PROBLEMS IN EXISTING SYSTEM

- An approach employing personalised birthweight guidelines that took into account the potential for embryo growth was described in Existing. They developed a new methodology to determine the ideal embryo weight at each gestational week that is tailored to each individual profile on the basis that birthweight varies with maternal and embryo physiological parameters (such as race/ethnicity, parity, sex, pre-pregnancy or early pregnancy body mass).

- It combines birthweight information from mothers who were 40 weeks pregnant with an estimated embryo weight calculated using the Hadlock estimated embryo weight curve. For particular maternal and embryo profiles, the ultrasonography EFW curves were proportionately altered either upwards or downwards based on the birthweight at 40 weeks.
- Many significant consequences can result from embryo growth restriction (FGR). You might have to have your kid early and have to stay in the hospital. Breathing difficulties, infections, and other issues could affect your infant. There could be deaths and stillbirths.
- **Evidence-based national guidelines for the management of suspected embryo growth restriction**

  The aim of this review is to: summarize areas of consensus and controversy between recently published national guidelines on small for gestational age or embryo growth restriction; highlight any recent evidence that should be incorporated into existing guidelines; and identify future research priorities in this field. A search of MEDLINE, Google, and the International Guideline Library identified 6 national guidelines on management of pregnancies complicated by embryo growth restriction/small for gestational age published from 2010 onwards. There is general consensus between guidelines (at least 4 of 6 guidelines in agreement) in early pregnancy risk selection, and use of low-dose aspirin for women with major risk factors for placental insufficiency. All highlight the importance of smoking cessation to prevent small for gestational age. While there is consensus in recommending fundal height measurement in the third trimester, 3 specify the use of a customized growth chart, while 2 recommend McDonald rule. Routine third-trimester scanning is not recommended for small-for-gestational-age screening, while women with major risk factors should have serial scanning in the third trimester. Umbilical artery Doppler studies in suspected small-for-gestational-age pregnancies are universally advised, however there is inconsistency in the recommended frequency for growth scans after diagnosis of small for gestational age/embryo growth restriction (2-4 weekly).

In late-onset embryo growth restriction (≥32 weeks) general consensus is to use cerebral Doppler studies to influence surveillance and/or delivery timing. Embryo surveillance methods (most recommend cardiotocography) and recommended timing of delivery vary. There is universal agreement on the use of corticosteroids before birth at <34 weeks, and general consensus on the use of magnesium sulfate for neuroprotection in early-onset embryo growth restriction (<32 weeks). Most guidelines advise using cardiotocography surveillance to plan delivery in embryo growth restriction <32 weeks. The recommended gestation at delivery for embryo growth restriction with absent and reversed end-diastolic velocity varies from 32 to ≥34 weeks and 30 to ≥34 weeks, respectively. Overall, where there is high-quality evidence from randomized controlled trials and meta-analyses, eg, use of umbilical artery Doppler and corticosteroids for delivery <34 weeks, there is a high degree of consistency between national small-for-gestational-age guidelines. This review discusses areas where there is potential for convergence between small-for-gestational-age guidelines based on existing randomized controlled trials of management of small-for-gestational-age pregnancies, and areas of controversy. Research priorities include assessing the utility of late third-trimester scanning to prevent major morbidity and mortality and to investigate the optimum timing of delivery in fetuses with late-onset embryo growth restriction and abnormal Doppler parameters. Prospective studies are needed to compare new international population ultrasound standards with those in current use.

- **Diagnosis and surveillance of late-onset embryo growth restriction**

  By consensus, late embryo growth restriction is that diagnosed >32 weeks. This condition is mildly associated with a higher risk of perinatal hypoxic events and suboptimal neurodevelopment. Histologically, it is characterized by the presence of uteroplacental vascular lesions (especially infarcts), although the incidence of such lesions is lower than in preterm embryo growth restriction. Screening procedures for embryo growth restriction need to identify small babies and then differentiate between those who are healthy and those who are

pathologically small. First- or second-trimester screening strategies provide detection rates for late smallness for gestational age <50% for 10% of false positives. Compared to clinically indicated ultrasonography in the third trimester, universal screening triples the detection rate of late smallness for gestational age. As opposed to early third-trimester ultrasound, scanning late in pregnancy (around 37 weeks) increases the detection rate for birthweight <3rd centile. Contrary to early embryo growth restriction, umbilical artery Doppler velocimetry alone does not provide good differentiation between late smallness for gestational age and embryo growth restriction. A combination of biometric parameters (with severe smallness usually defined as estimated embryo weight or abdominal circumference <3rd centile) with Doppler criteria of placental insufficiency (either in the maternal [uterine Doppler] or embryo [cerebroplacental ratio] compartments) offers a classification tool that correlates with the risk for adverse perinatal outcome. There is no evidence that induction of late embryo growth restriction at term improves perinatal outcomes nor is it a cost-effective strategy, and it may increase neonatal admission when performed <38 weeks.

CHAPTER 3

REQUIREMENTS ANALYSIS

3.1 OBJECTIVE OF THE PROJECT

Using machine learning methods, this paper aims to anticipate and diagnose prenatal problems early. The broad description of our project, including user requirements, a product viewpoint, an overview of the requirements, and general restrictions, will be provided in depth in this document. It will also include the precise specifications and functionality required for this project, such as the interface, functional specifications, and performance specifications.

3.2 SOFTWARE REQUIREMENTS SPECIFICATION DOCUMENT

3.2.1 *HARDWARE REQUIREMENTS*

- System : Pentium i3 Processor
- Hard Disk : 500 GB.
- Monitor : 15'' LED
- Input Devices : Keyboard, Mouse
- Ram : 2 GB

3.2.2 *SOFTWARE REQUIREMENTS*

- **Operating System : Windows10**
- Coding Language : Python

CHAPTER 4
DESCRIPTION OF PROPOSED SYSTEMS

4.1 PROPOSED SYSTEM / MODULES

- Data Collection Module
- Analysis Module
- Feature Extraction
- Predict Embryo Growth

4.1.1 *Data Collection Module*

These recommendations serve as a desirable benchmark for gathering information on availability post immunisation, allowing for data comparison, and are suggested as an addition to information gathered for the particular study topic and context. The recommendations are not meant to serve as a guidance for FGR reporting to surveillance systems or study monitors. The case definition's criteria, which are not repeated in these guidelines, must also be consulted by researchers who are creating a data collection instrument based on these principles.

4.1.2 *Analysis Module*

In this module, we can analyse the embryo growth level and give patient feedbacks if it is life threatening or not.

4.1.3 *Feature Extraction*

In this project, a learning machine technique-based method for feature extraction and classification of embryo growth between pregnancies is presented.

4.1.4 *Predict Embryo Growth*

In this module, it is discussed how a healthy pregnancy depends on normal embryo growth, which also affects the long-term health of the unborn child. Nevertheless, it has long been difficult in both clinical practise and research to distinguish between normal and aberrant prenatal growth. Both research and clinical practise should benefit from a more precise characterization of aberrant embryo growth.

4.2 PROGRAMMING LANGUAGE

4.2.1 *PYTHON*

Python is a general-purpose interpreted, interactive, object-oriented, and high-level programming language. It was created by Guido van Rossum during 1985- 1990. Like Perl, Python source code is also available under the GNU General Public License (GPL). This **tutorial** gives enough understanding on **Python programming** language**.** The finest programming language for machine learning is Python. According to research and surveys, Python is the sixth most significant language to date as the go-to language for information science and machine learning. Python has the following benefits as a result of-

- ✓ **Easy to be told and perceive-** Because of Python's simplified structure, learning and understanding the language is quite simple, especially for beginners..
- ✓ **Multi-purpose language −** Because it offers both structured and object-oriented programming in addition to practical programming, Python may be a multi-purpose programming language.
- ✓ **Support of open supply community −** Python is maintained by an incredibly sizable developer community as an open source programming language. As a result, the Python community has been merely mounting the bugs. Due to this quality, Python is incredibly resilient and adaptable

4.3 HISTORY OF PYTHON

Python was developed by Guido van Rossum in the late eighties and early nineties at the National Research Institute for Mathematics and Computer Science in the Netherlands.

Python is derived from many other languages, including ABC, Modula-3, C, C++, Algol-68, SmallTalk, and Unix shell and other scripting languages.

Python is copyrighted. Like Perl, Python source code is now available under the GNU General Public License (GPL).

Python is now maintained by a core development team at the institute, although Guido van Rossum still holds a vital role in directing its progress.

4.4 APPLICATION OF PYTHON

- **Easy-to-learn** − Python has few keywords, simple structure, and a clearly defined syntax. This allows the student to pick up the language quickly.
- **Easy-to-read** − Python code is more clearly defined and visible to the eyes.
- **Easy-to-maintain** − Python's source code is fairly easy-to-maintain.
- **A broad standard library** − Python's bulk of the library is very portable and cross-platform compatible on UNIX, Windows, and Macintosh.
- **Interactive Mode** − Python has support for an interactive mode which allows interactive testing and debugging of snippets of code.
- **Portable** − Python can run on a wide variety of hardware platforms and has the same interface on all platforms.
- **Extendable** − You can add low-level modules to the Python interpreter. These modules enable programmers to add to or customize their tools to be more efficient.
- **Databases** − Python provides interfaces to all major commercial databases.
- **GUI Programming** − Python supports GUI applications that can be created and ported to many system calls, libraries and windows systems, such as

Windows MFC, Macintosh, and the X Window system of Unix.

- **Scalable** − Python provides a better structure and support for large programs than shell scripting.

4.5 FEATURES OF PYTHON

- It supports functional and structured programming methods as well as OOP.
- It can be used as a scripting language or can be compiled to byte-code for building large applications.
- It provides very high-level dynamic data types and supports dynamic type checking.
- It supports automatic garbage collection.
- It can be easily integrated with C, C++, COM, ActiveX, CORBA, and Java.

4.6 FEASIBILITY STUDY

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

The feasibility study investigates the problem and the information needs of the stakeholders. It seeks to determine the resources required to provide an information systems solution, the cost and benefits of such a solution, and the feasibility of such a solution.

The goal of the feasibility study is to consider alternative information systems solutions, evaluate their feasibility, and propose the alternative most suitable to the organization. The feasibility of a proposed solution is evaluated in terms of its components.

4.7 ECONOMICAL FEASIBILITY

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the

research and development of the system is limited. The expenditures must justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

4.8 TECHNICAL FEASIBILITY

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

4.9 SOCIAL FEASIBILITY

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity.

4.10 ARCHITECTURE / OVERALL DESIGN OF PROPOSED SYSTEM

The process of establishing the systems' constituent subsystems as well as the framework for subsystem control and communication is known as system architectural design. The overall structure of the software system is to be established by the architectural design.

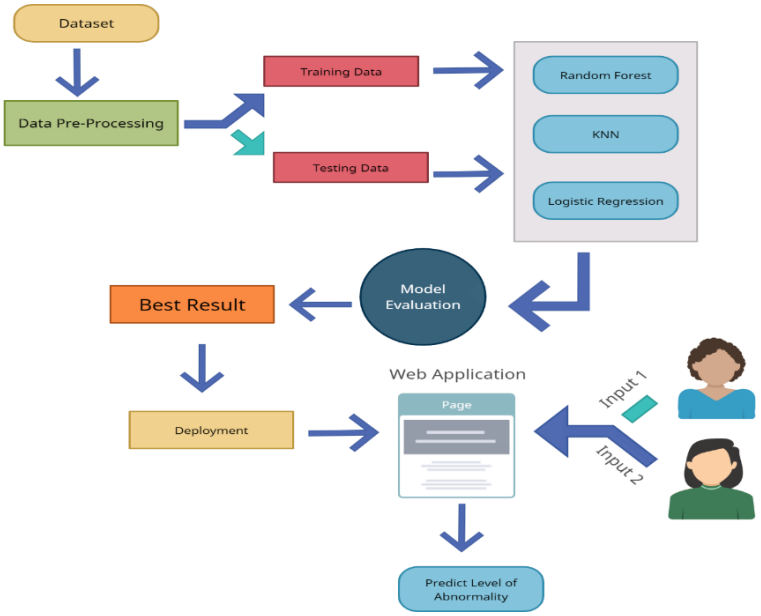


Fig 4.10 System Architecture

4.11 ALGORITHMS

4.11.1 *RANDOM FOREST ALGORITHM*

The Random Forest approach is applicable to both classification and regression-type issues. You will discover how the random forest method, used in machine learning for the classification problem, operates in this.

Popular machine learning algorithm Random Forest is a part of the supervised learning methodology. It is employed in ML for both Classification and Regression issues. based on the idea of ensemble learning, which is the act of mixing various classifiers to solve a challenging problem and enhance model performance.

In a random forest algorithm, there are many different decision trees. The random forest algorithm creates a "forest" that is trained via bagging or bootstrap aggregation. The accuracy of machine learning algorithms is increased by bagging, an ensemble meta-algorithm.

Random Forest, as the name implies, is a classifier that uses a number of decision trees on different subsets of the provided dataset and averages them to increase the

dataset's predictive accuracy. Instead, then depending on a single decision tree, the random forest uses forecasts from each tree and predicts the result based on the votes of the majority of predictions.

Below are some points that explain why we should use the Random Forest algorithm:

- It takes less training time as compared to other algorithms.
- It predicts output with high accuracy, even for the large dataset it runs efficiently.
- It can also maintain accuracy when a large proportion of data is missing.

The below diagram explains the working of the Random Forest algorithm:

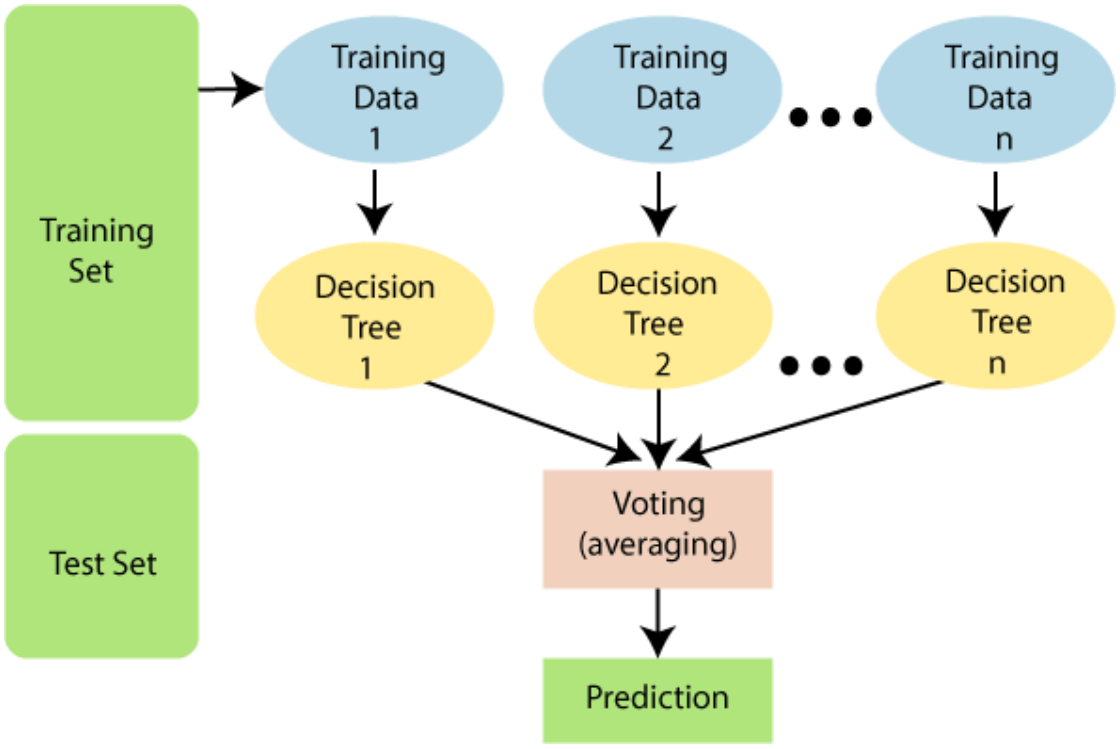


Fig 4.11.1 working of the Random Forest Regressor algorithm

Features of a Random Forest Algorithm

- It's more accurate than the decision tree algorithm.
- It provides an effective way of handling missing data.
- It can produce a reasonable prediction without hyper-parameter tuning.

- It solves the issue of overfitting in decision trees.
- In every random forest tree, a subset of features is selected randomly at the node's splitting point.

Classification in random forests

Classification in random forests employs an ensemble methodology to attain the outcome. The training data is fed to train various decision trees. This dataset consists of observations and features that will be selected randomly during the splitting of nodes.

A rain forest system relies on various decision trees. Every decision tree consists of decision nodes, leaf nodes, and a root node. The leaf node of each tree is the final output produced by that specific decision tree. The selection of the final output follows the majority-voting system. In this case, the output chosen by the majority of the decision trees becomes the final output of the rain forest system. The diagram below shows a simple random forest classifier.

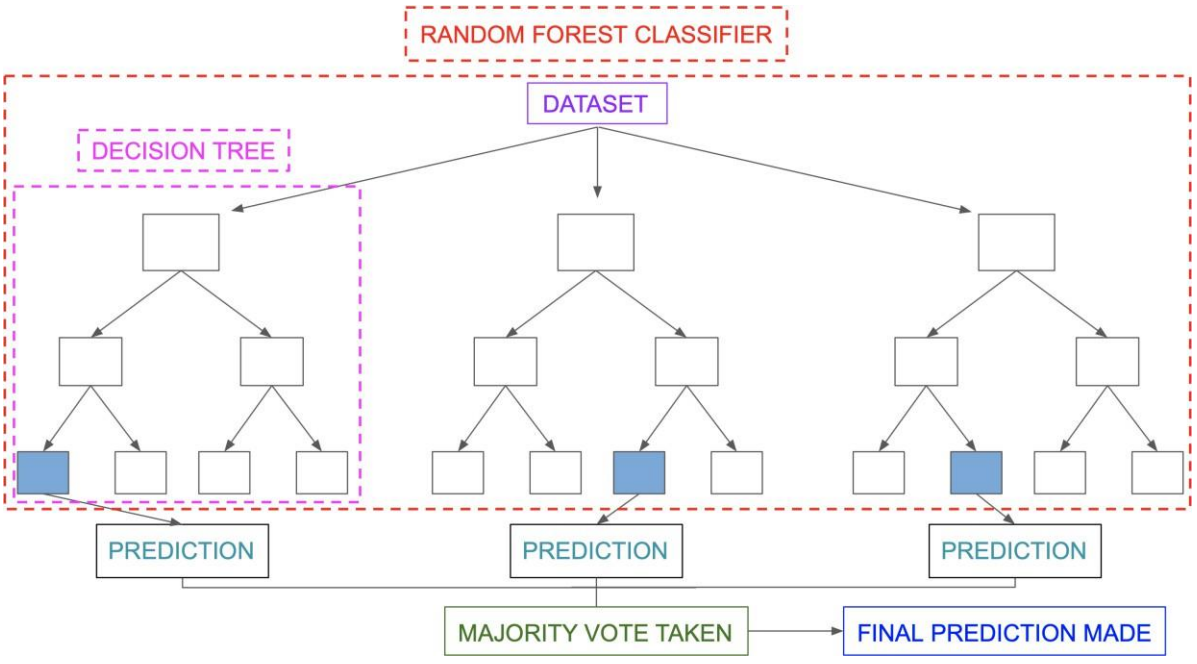


Fig 4.11.1 working of Random Forest Classifier

Random Forest Steps

1. Randomly select “k” features from total “m” features. (Where k << m)

2. Among the “k” features, calculate the node “d” using the best split point.

3. Split the node into daughter nodes using the best split.

4. Repeat 1 to 3 steps until “l” number of nodes has been reached.

5. Build forest by repeating steps 1 to 4 for “n” number times to create “n” number of trees.

The beginning of random forest algorithm starts with randomly selecting “k” features out of total “m” features. In the image, you can observe that we are randomly taking features and observations. The working of the algorithm can be better understood by the below example:
Example: Suppose there is a dataset that contains multiple fruit images. So, this dataset is given to the Random forest classifier. The dataset is divided into subsets and given to each decision tree. During the training phase, each decision tree produces a prediction result, and when a new data point occurs, then based on the majority of results, the Random Forest classifier predicts the final decision. Consider the below image:

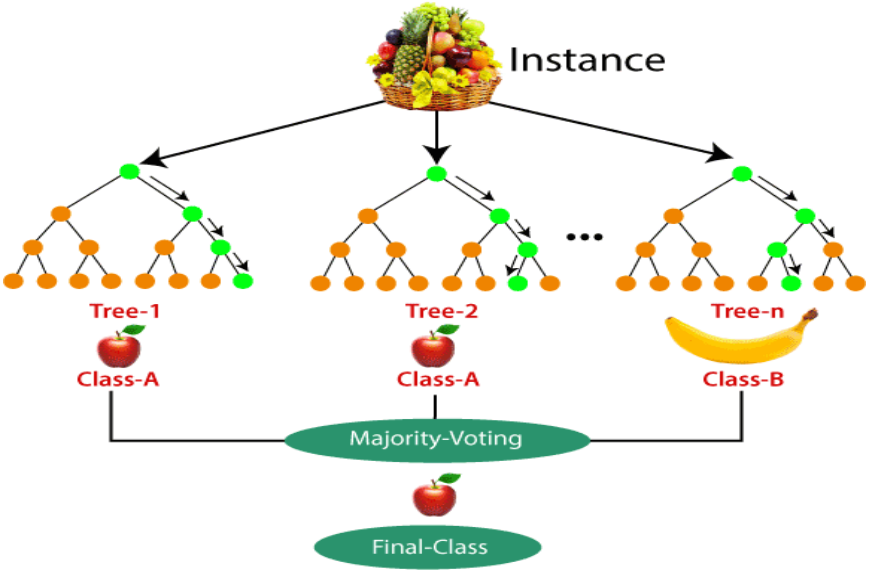


Fig 4.11.1 Random Forest Classifier with example

APPLICATIONS OF RANDOM FOREST

There are mainly four sectors where Random forest mostly used:

1. Banking: Banking sector mostly uses this algorithm for the identification of loan risk.
2. Medicine: With the help of this algorithm, disease trends and risks of the disease can be identified.
3. Land Use: We can identify the areas of similar land use by this algorithm.
4. Marketing: Marketing trends can be identified using this algorithm.

Advantages of Random Forest

Random Forest is capable of performing both Classification and Regression tasks.

- It is capable of handling large datasets with high dimensionality.
- It enhances the accuracy of the model and prevents the overfitting issue.

Disadvantages of Random Forest

- Although random forest can be used for both classification and regression tasks, it is not more suitable for Regression tasks.

4.11.2 *KNN ALGORITHMS*

One of the simplest machine learning algorithms, based on the supervised learning method, is K-Nearest Neighbour. This algorithm places the new case in the category that matches the available categories the most by assuming similarity between the new case/data and existing cases.

In order to classify a new data point based on similarity, it stores all of the existing data. This means that utilising the K-NN method, fresh data can be quickly and accurately sorted into a suitable category. Although the K-NN approach is most frequently employed for classification problems, it can also be utilised for regression.

As a non-parametric algorithm, it makes no assumptions about the underlying data. It is also known as a lazy learner algorithm since it saves the training dataset rather than learning from it immediately. Instead, it uses the dataset to perform an action when classifying data.

During the training phase, this algorithm simply saves the information, and when it receives new data, it categorises it into a category that is quite similar to the new data.

Example: Let's say we have a picture of a species that resembles both cats and dogs, but we aren't sure if it is one or the other. Therefore, since the KNN algorithm is based on a similarity metric, we can utilise it for this identification. Our KNN model will look for similarities between the new data set's features and those in the photos of cats and dogs, and based on those similarities, it will classify the new data set as either cat- or dog-related.

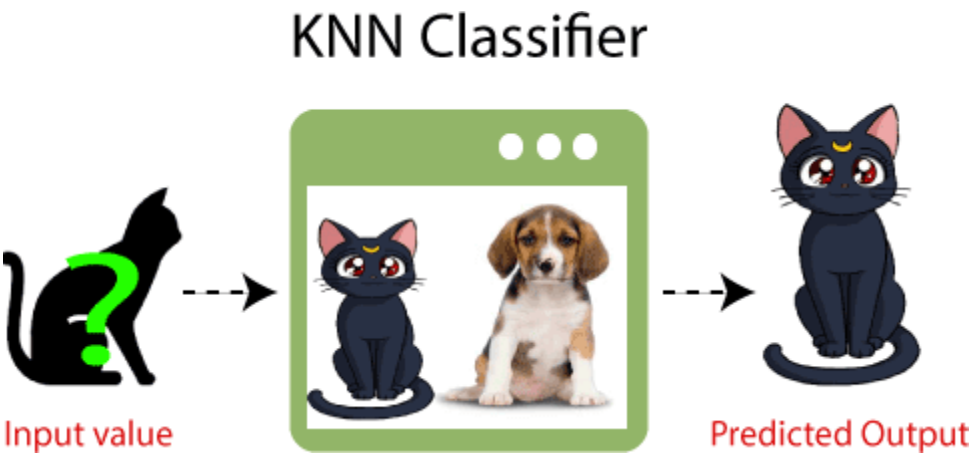


Fig 4.11.2 KNN algorithm example

WORKING OF KNN ALGORITHM

The K-NN working can be explained on the basis of the below algorithm:

- **Step-1:** Select the number K of the neighbours
- **Step-2:** Calculate the Euclidean distance of K number of neighbours
- **Step-3:** Take the K nearest neighbours as per the calculated Euclidean distance.
- **Step-4:** Among these k neighbours, count the number of the data points in each category.
- **Step-5:** Assign the new data points to that category for which the number of the neighbour is maximum.
- **Step-6:** Our model is ready.

Suppose we have a new data point and we need to put it in the required category.

- Firstly, we will choose the number of neighbors, so we will choose the k=5.
- Next, we will calculate the **Euclidean distance** between the data points. If (x_1, y_1) and (x_2, y_2) are the two points in the two-dimensional plane, the Euclidean distance formula is given by:

  Euclidean distance, $d = \sqrt{[(x_2 - x_1)^2 + (y_2 - y_1)^2]}$
- By calculating the Euclidean distance we got the nearest neighbors, as three nearest neighbors in category A and two nearest neighbors in category B.

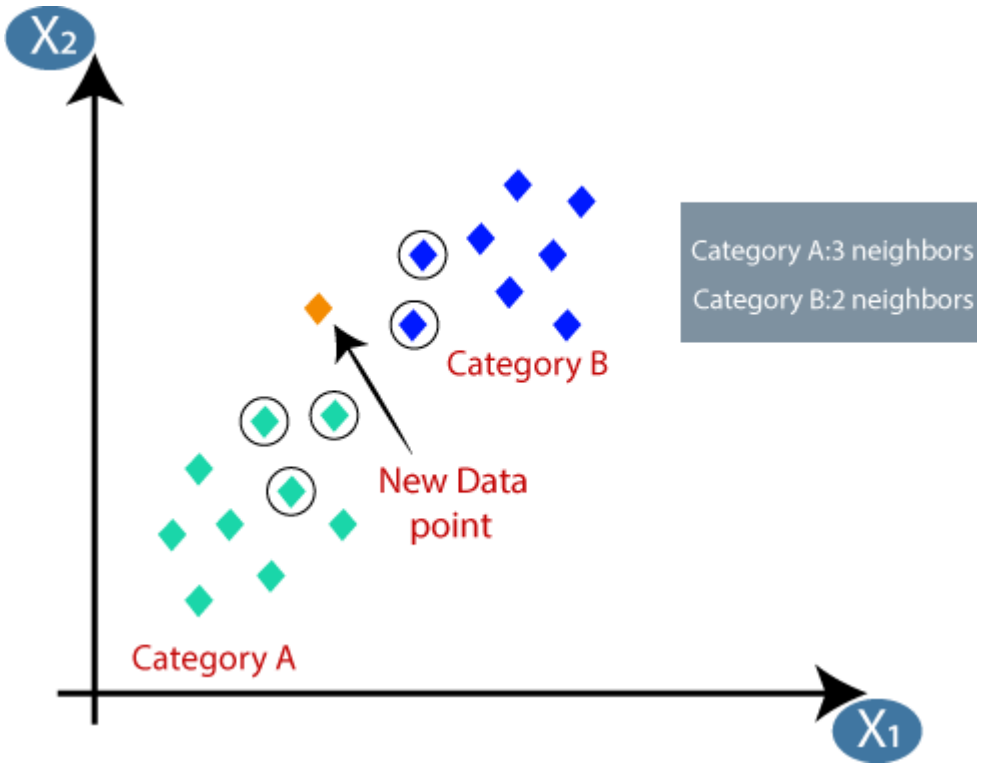


Fig 4.11.2 graphical view of knn algorithm classification

- As we can see the 3 nearest neighbors are from category A, hence this new data point must belong to category A.

Below are some points to remember while selecting the value of K in the K-NN algorithm:

- There is no particular way to determine the best value for "K", so we need to try some values to find the best out of them. The most preferred value for K is 5.
- A very low value for K such as K=1 or K=2, can be noisy and lead to the effects of outliers in the model.
- Large values for K are good, but it may find some difficulties.

Advantages of KNN Algorithm:

- It is simple to implement.
- It is robust to the noisy training data
- It can be more effective if the training data is large.

Disadvantages of KNN Algorithm:

- Always needs to determine the value of K which may be complex some time.
- The computation cost is high because of calculating the distance between the data points for all the training samples.

4.11.3 *LOGISTIC REGRESSION*

One of the most often used Machine Learning algorithms, within the category of Supervised Learning, is logistic regression. Using a predetermined set of independent factors, it is used to predict the categorical dependent variable. In a categorical dependent variable, the output is predicted via logistic regression.

Therefore, the result must be a discrete or categorical value. Rather than providing the exact values of 0 and 1, it provides the probabilistic values that fall between 0 and 1. It can be either Yes or No, 0 or 1, true or false, etc. With the exception of how they are used, it is very comparable to linear regression.

While logistic regression is used to solve classification difficulties, linear regression is used to solve regression problems. In logistic regression, we fit a "S" shaped logistic function, which predicts two maximum values, rather than a regression line (0 or 1). The logistic function's curve shows the possibility of several things, including whether or not the cells are malignant, whether or not a mouse is obese depending on its weight, etc.

Because it can classify new data using both continuous and discrete datasets, logistic regression is a key machine learning approach. When classifying observations using various sources of data, logistic regression can be used to quickly identify the factors that will work well.

The below image is showing the logistic function:

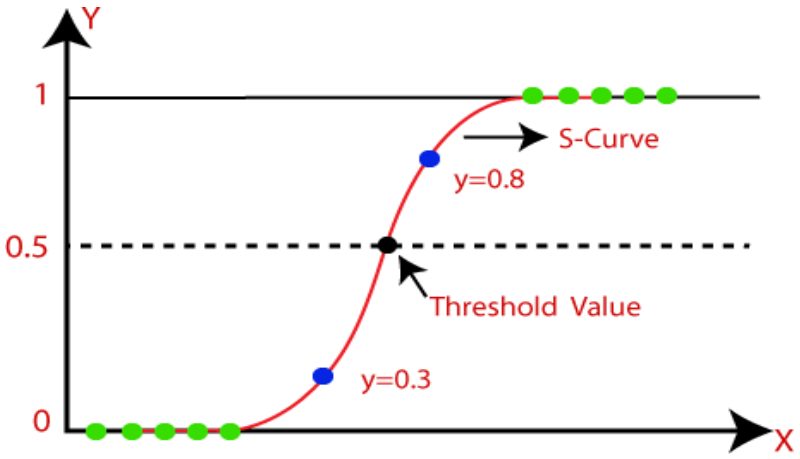


Fig 4.11.3 Logistic regression

Logistic Function (Sigmoid Function):

- The sigmoid function is a mathematical function used to map the predicted values to probabilities.
- It maps any real value into another value within a range of 0 and 1.
- The value of the logistic regression must be between 0 and 1, which cannot go beyond this limit, so it forms a curve like the "S" form. The S-form curve is called the Sigmoid function or the logistic function.
- In logistic regression, we use the concept of the threshold value, which defines the probability of either 0 or 1. Such as values above the threshold value tends to 1, and a value below the threshold values tends to 0.

Assumptions for Logistic Regression:

- The dependent variable must be categorical in nature.
- The independent variable should not have multi-collinearity.

Type of Logistic Regression:

On the basis of the categories, Logistic Regression can be classified into three types:

- **Binomial:** In binomial Logistic regression, there can be only two possible types of the dependent variables, such as 0 or 1, Pass or Fail, etc.

- **Multinomial:** In multinomial Logistic regression, there can be 3 or more possible unordered types of the dependent variable, such as "cat", "dogs", or "sheep"
- **Ordinal:** In ordinal Logistic regression, there can be 3 or more possible ordered types of dependent variables, such as "low", "Medium", or "High".

CHAPTER 5
SYSTEM ANALYSIS

5.1 PURPOSE

The purpose of this document is approach of early diagnosis and prediction of embryo abnormalities using machine learning algorithms. In detail, this document will provide a general description of our project, including user requirements, product perspective, and overview of requirements, general constraints. In addition, it will also provide the specific requirements and functionality needed for this project - such as interface, functional requirements and performance requirements.

5.2 SCOPE

The scope of this SRSdocument persists for the entire life cycle of the project. This document defines the final state of the software requirements agreed upon by the customers and designers. Finally at the end of the project execution all the functionalities may be traceable from the SRSto the product. The document describes the functionality, performance, constraints, interface and reliability for the entire life cycle of the project.

5.3 EXISTING SYSTEM

In Existing, a method using customized birthweight norms that incorporated information about embryo growth potential. Based on the premise that birthweight varies with maternal and embryo physiological parameters (e.g., race/ethnicity, parity, sex, prepregnancy or early pregnancy body mass), they defined a new methodology to calculate optimal embryo weight at each gestational week customized by individual profile. The authors combined birthweight data at 40 weeks of gestation with estimated embryo weight based on the Hadlock estimated embryo weight curve.The ultrasound EFW curves were proportionately adjusted upwards or downwards according to the birthweight at 40 weeks for specific maternal and embryo profiles.

5.4 PROPOSED SYSTEM

In proposed system, discuss with recent advances towards an integrated definition for embryo growth restriction. Such a definition may incorporate embryo size with the status of placental health measured by maternal and embryo Doppler velocimetry and biomarkers, biophysical findings and genetics. Although the concept of an integrated definition appears promising, further development and testing are required. An improved definition of abnormal embryo growth should benefit both research and clinical practice.

CHAPTER 6

SYSTEM DESIGN

6.1 INPUT DESIGN

The input design is the link between the information system and the user. It comprises the developing specification and procedures for data preparation and those steps are necessary to put transaction data in to a usable form for processing can be achieved by inspecting the computer to read data from a written or printed document or it can occur by having people keying the data directly into the system. The design of input focuses on controlling the amount of input required, controlling the errors, avoiding delay, avoiding extra steps and keeping the process simple. The input is designed in such a way so that it provides security and ease of use with retaining the privacy. Input Design considered the following things:

- What data should be given as input?
- How the data should be arranged or coded?
- The dialog to guide the operating personnel in providing input.
- Methods for preparing input validations and steps to follow when error occur.

6.2 OUTPUT DESIGN

A quality output is one, which meets the requirements of the end user and presents the information clearly. In any system results of processing are communicated to the users and to other system through outputs. In output design it is determined how the information is to be displaced for immediate need and also the hard copy output. It is the most important and direct source information to the user. Efficient and intelligent output design improves the system's relationship to help user decision-making.

The output form of an information system should accomplish one or more of the following objectives.

- Convey information about past activities, current status or projections of the
- Future.

- Signal important events, opportunities, problems, or warnings.
- Trigger an action.
- Confirm an action

6.3 DATA FLOW DIAGRAM

1. The DFD is also called as bubble chart. It is a simple graphical formalism that can be used to represent a system in terms of input data to the system, various processing carried out on this data, and the output data is generated by this system.
2. The data flow diagram (DFD) is one of the most important modeling tools. It is used to model the system components. These components are the system process, the data used by the process, an external entity that interacts with the system and the information flows in the system.
3. DFD shows how the information moves through the system and how it is modified by a series of transformations. It is a graphical technique that depicts information flow and the transformations that are applied as data moves from input to output.
4. DFD is also known as bubble chart. A DFD may be used to represent a system at any level of abstraction. DFD may be partitioned into levels that represent increasing information flow and functional detail.

6.4 UML DIAGRAMS

UML stands for Unified Modeling Language. UML is a standardized general-purpose modeling language in the field of object-oriented software engineering. The standard is managed, and was created by, the Object Management Group.

The goal is for UML to become a common language for creating models of object oriented computer software. In its current form UML is comprised of two major components: a Meta-model and a notation. In the future, some form of method or process may also be added to; or associated with, UML.

The Unified Modeling Language is a standard language for specifying, Visualization, Constructing and documenting the artifacts of software system, as well as for business modeling and other non-software systems.

The UML represents a collection of best engineering practices that have proven successful in the modeling of large and complex systems.

The UML is a very important part of developing objects oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects.

GOALS:

The Primary goals in the design of the UML are as follows:

1. Provide users a ready-to-use, expressive visual modeling Language so that they can develop and exchange meaningful models.
2. Provide extendibility and specialization mechanisms to extend the core concepts.
3. Be independent of particular programming languages and development process.
4. Provide a formal basis for understanding the modeling language.
5. Encourage the growth of OO tools market.
6. Support higher level development concepts such as collaborations, frameworks, patterns and components.
7. Integrate best practices.

6.4.1 USE CASE DIAGRAM:

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.

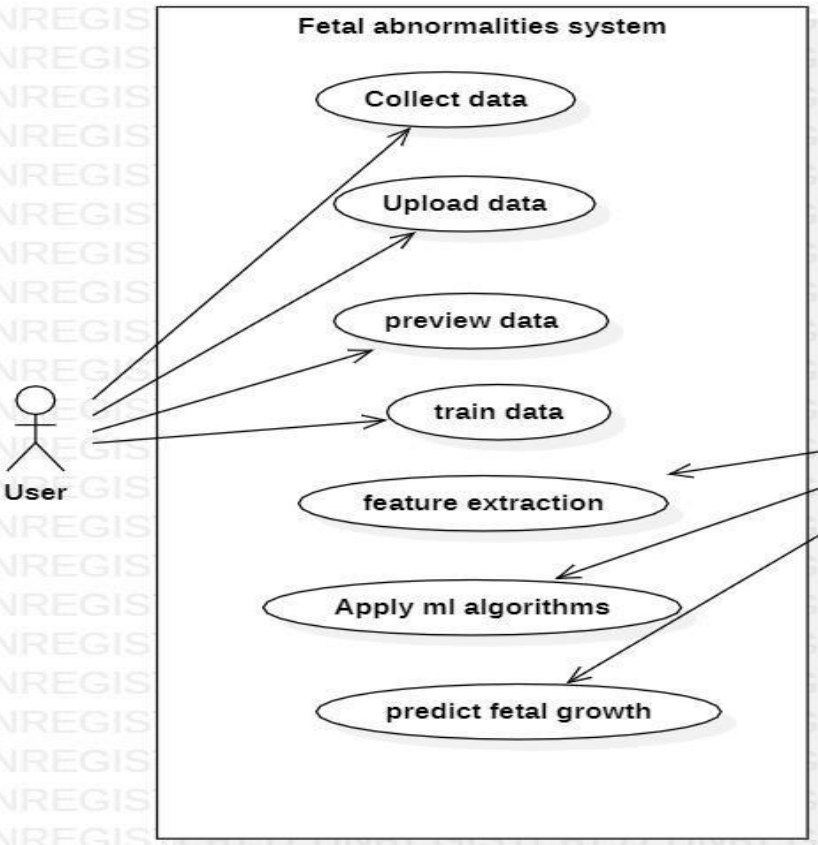


Fig 6.4.1 use case diagram

6.4.2 SEQUENCE DIAGRAM:

A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.

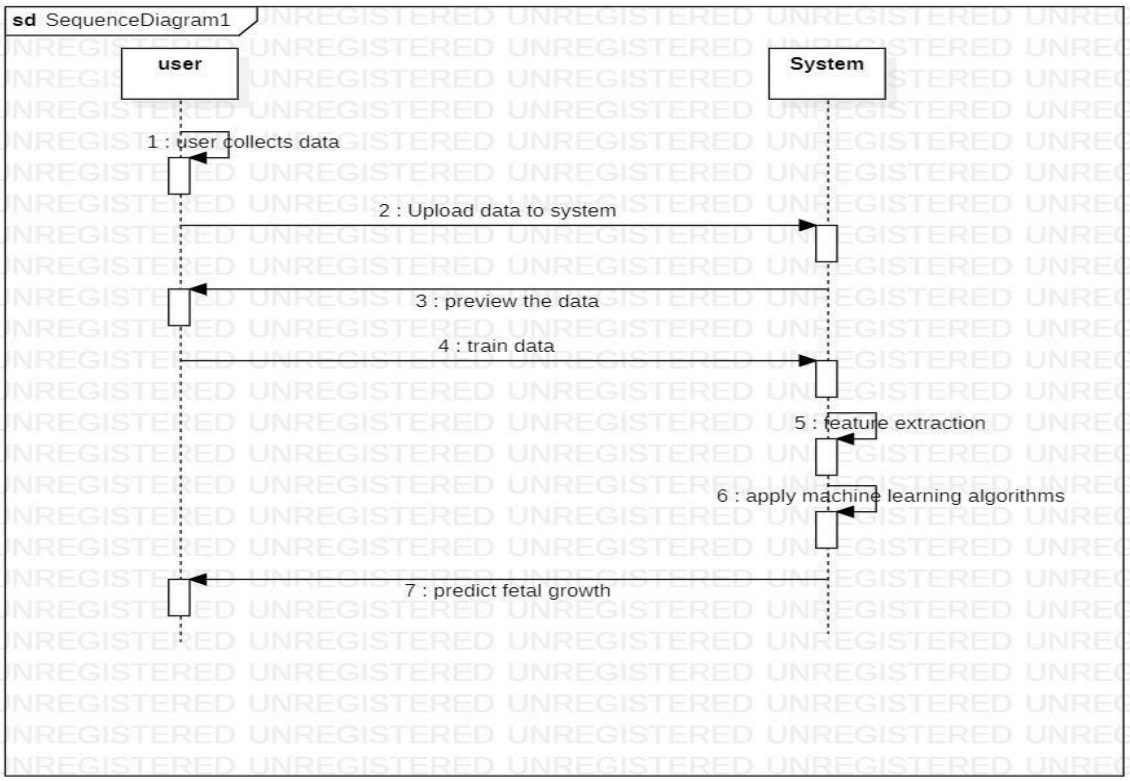


Fig 6.4.2 sequence diagram

6.4.3 ACTIVITY DIAGRAM:

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.

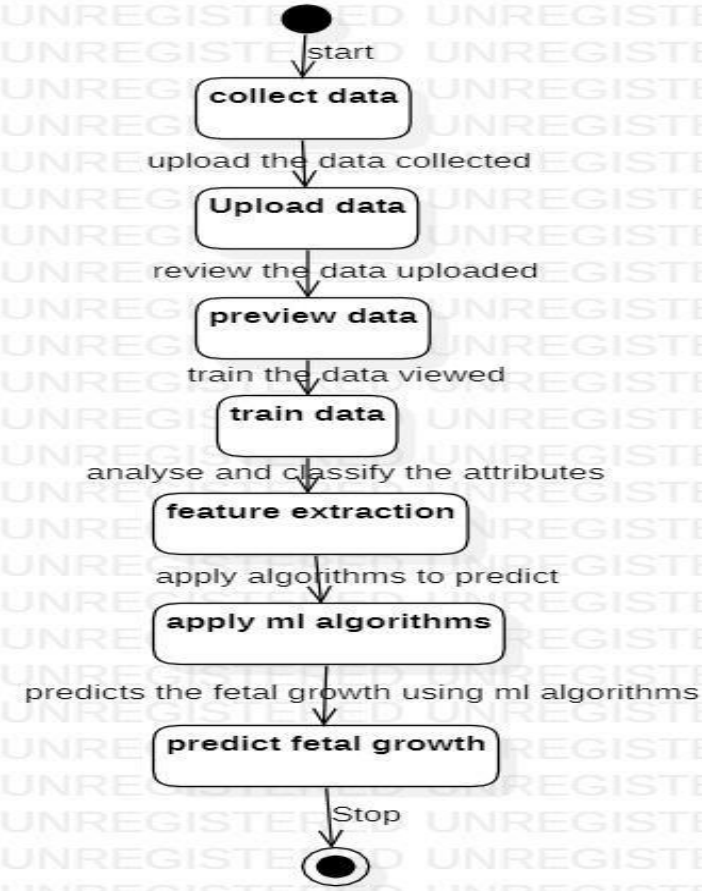


Fig 6.4.3 activity diagram

CHAPTER 7
SYSTEM TESTING

7.1 Overview

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the

Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

7.2 TYPES OF TESTS

Unit testing

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

Integration testing

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically

aimed at exposing the problems that arise from the combination of components. Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects. The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

Functional test

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

Systems/Procedures: interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

System Test

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System

testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

White Box Testing

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

Black Box Testing

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box .you cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.

Unit Testing:

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

Acceptance Testing

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

7.3 Test strategy and approach

Field testing will be performed manually and functional tests will be written in detail.

7.3.1 Test objectives

- All field entries must work properly.
- Pages must be activated from the identified link.
- The entry screen, messages and responses must not be delayed.

7.3.2 Features to be tested

- Verify that the entries are of the correct format
- No duplicate entries should be allowed
- All links should take the user to the correct page.

7.4 REQUIREMENT ANALYSIS

Requirement analysis, also called requirement engineering, is the process of determining user expectations for a new modified product. It encompasses the tasks that determine the need for analysing, documenting, validating and managing software or system requirements. The requirements should be documentable, actionable, measurable, testable and traceable related to identified business needs or opportunities and define to a level of detail, sufficient for system design.

7.5 FUNCTIONAL REQUIREMENTS

It is a technical specification requirement for the software products. It is the first step in the requirement analysis process which lists the requirements of particular software systems including functional, performance and security requirements. The function of the system depends mainly on the quality hardware used to run the software with given functionality.

Usability

It specifies how easy the system must be use. It is easy to ask queries in any format which is short or long, porter stemming algorithm stimulates the desired response for user.

Robustness

It refers to a program that performs well not only under ordinary conditions but also under unusual conditions. It is the ability of the user to cope with errors for irrelevant queries during execution.

Security

The state of providing protected access to resource is security. The system provides good security and unauthorized users cannot access the system there by providing high security.

Reliability

It is the probability of how often the software fails. The measurement is often expressed in MTBF (Mean Time Between Failures). The requirement is needed in order to ensure that the processes work correctly and completely without being aborted. It can handle any load and survive and survive and even capable of working around any failure.

Compatibility

It is supported by version above all web browsers. Using any web servers like localhost makes the system real-time experience.

Flexibility

The flexibility of the project is provided in such a way that is has the ability to run on different environments being executed by different users.

Safety

Safety is a measure taken to prevent trouble. Every query is processed in a secured manner without letting others to know one’s personal information.

7.6 NON- FUNCTIONAL REQUIREMENTS

Portability

It is the usability of the same software in different environments. The project can be run in any operating system.

Performance

These requirements determine the resources required, time interval, throughput and everything that deals with the performance of the system.

Accuracy

The result of the requesting query is very accurate and high speed of retrieving information. The degree of security provided by the system is high and effective.

Maintainability

Project is simple as further updates can be easily done without affecting its stability. Maintainability basically defines that how easy it is to maintain the system. It means that how easy it is to maintain the system, analyse, change and test the application. Maintainability of this project is simple as further updates can be easily done without affecting its stability.

CHAPTER 8
CONCLUSION

Embryo Health denotes the health and growth of the embryo and frequent contacts in the uterus of the pregnant women during pregnancy. Maximum pregnancy period complexities leads embryo to a severe difficulty which limits right growth that causes deficiency or death. Harmless pregnancy period by predicting the risk levels before the occasion of difficulties boost right embryo growth. This paper presented the various approaches and researches done so far for forecasting the embryo health and growth state from a set of pre-classified patterns knowledge with its accuracy. Predication of embryo health is vital in developing a predictive classifier model using Machine Learning Algorithms. Complications during pregnancy can result in severe difficulties for the fetus, hindering proper growth and leading to deficiencies or even death. To promote healthy embryo development, it is beneficial to predict the likelihood of potential risks before they occur during pregnancy. This approach can help ensure a safe and healthy pregnancy, allowing the fetus to grow appropriately. This study discusses the various methods and investigations that have been conducted to accurately forecast embryo growth condition using pre- classified patterns. Developing a predictive classifier model using machine learning algorithms is essential for predicting embryo health accurately

REFERENCES

[1]. Ali Gholipour, Estroff JudyA, BarnewoltCarolE, Connolly SusanA, Warfield SimonK. Embryo brain volumetry through MRI volumetric reconstruction and segmentation. Int. J. Comput. Assist. Radiol. Surg. 2011;6(3):329–39.

[2]. Anton-Rodriguez J. M, Peter Julyan, Ibrahim Djoukhadar, David Russell, D. Gareth Evans, Alan Jackson, and Julian C. Matthews, (2019) “Comparison of a Standard Resolution PET-CT Scanner With an HRRT Brain Scanner for Imaging Small Tumors Within the Head,” IEEE Transactions on Radiation and Plasma Medical Sciences, vol. 3, no. 4.

[3]. Challa M., Chinnaiyan R. (2020) Optimized Machine Learning Approach for the Prediction of Diabetes-Mellitus. In: Smys S., Tavares J., Balas V., Iliyasu A. (eds) Computational Vision and BioInspired Computing. ICCVBIC 2019. Advances in Intelligent Systems and Computing, vol 1108. Springer, Cham

[4]. G. Sabarmathi and R. Chinnaiyan, "Big Data Analytics Framework for Opinion Mining of Patient Health Care Experience," 2020 Fourth International Conference on Computing Methodologies and Communication (ICCMC), Erode, India, 2020, pp. 352-357

[5]. G. Sabarmathi and R. Chinnaiyan, "Investigations on big data features research challenges and applications," 2017 International Conference on Intelligent Computing and Control Systems (ICICCS), Madurai, 2017, pp. 782-786

[6]. G. Sabarmathi and R. Chinnaiyan, "Reliable Machine Learning Approach to Predict Patient Satisfaction for Optimal Decision Making and Quality Health Care," 2019 International Conference on Communication and Electronics Systems (ICCES), Coimbatore, India, 2019, pp. 1489-1493.

[7]. Optimized Machine Learning Approach for the Prediction of Diabetes-Mellitus, Challa M.,Chinnaiyan R. 2020. Smys, S., Tavares, J., Balas, and A. Iliyasu. (eds) Computational Vision and BioInspired Computing. ICCVBIC 2019.. Springer, Cham.

[8]. Van Leeuwen, P., Lange, S., Bettermann, H., Grönemeyer, D. and Hatzmann, W., 1999. Embryo heart rate variability and complexity in the course of pregnancy. Early human development, 54(3), pp.259-269

[9]. Liston R, Sawchuck D, Young D, Brassard N, Campbell K, Davies G, Ehman W, Farine D, Farquharson D, Hamilton E, Helewa M. Embryo health surveillance: antepartum and intrapartum consensus guideline. Journal of obstetrics and gynaecology Canada. 2007 Sep 1;29(9):S3- 4.

APPENDIX

A. SOURCE CODE

```
# Import all the tools we need

# Regular EDA (exploratory data analysis) and plotting libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# we want our plots to appear inside the notebook
%matplotlib inline

# Models from Scikit-Learn
from sklearn.linear_model import LogisticRegression
from sklearn.neighbors import KNeighborsClassifier
from sklearn.ensemble import RandomForestClassifier

# Model Evaluations
from sklearn.model_selection import train_test_split, cross_val_score
from sklearn.model_selection import RandomizedSearchCV, GridSearchCV
from sklearn.metrics import confusion_matrix, classification_report
from sklearn.metrics import precision_score, recall_score, f1_score
from sklearn.metrics import plot_roc_curve
df = pd.read_csv("fetal_health.csv")
df.shape
df.head()
df.tail()
# Let's find out how many of each class there are
df["fetal_health"].value_counts()
# % of total
df["fetal_health"].value_counts(1)
df["fetal_health"].value_counts(1).plot(kind="bar", color=["yellowgreen", "gold", "firebrick"]);
height = (df["fetal_health"].value_counts(1)*100).to_list()
bars = ('Normal', 'Suspect', 'Pathologic')
y_pos = np.arange(len(bars))
plt.bar(y_pos, height, color=('yellowgreen','gold','firebrick'))

# use the plt.xticks function to custom labels
plt.xticks(y_pos, bars, rotation=45, horizontalalignment='right')

# remove labels
plt.tick_params(labelbottom='off')

plt.title('Percent of Patients by Fetal Health Status');
df.info()
```

```
df.isna().sum()
df.dtypes
df.describe()
len(df["histogram_variance"].value_counts())

# Length tells us that there are 133 different values
df["histogram_variance"].value_counts()

# Create another figure
plt.figure(figsize=(10, 6))

# Scatter with normal examples
plt.scatter(df.baseline_value[df.fetal_health==1],
            df.accelerations[df.fetal_health==1],
            c="yellowgreen")
# Scatter with suspect examples
plt.scatter(df.baseline_value[df.fetal_health==2],
            df.accelerations[df.fetal_health==2],
            c="gold")

# Scatter with pathologic examples
plt.scatter(df.baseline_value[df.fetal_health==3],
            df.accelerations[df.fetal_health==3],
            c="firebrick")

# Add some helpful info
plt.title("Fetal Health Status in function of Baseline FHR and Accelerations")
plt.ylabel("Accelerations")
plt.xlabel("Baseline FHR")
plt.legend(["Normal", "Suspect", "Pathologic"]);
df.baseline_value.plot.hist(bins=np.arange(100,180,10),
                            figsize=(5,5));
df['baseline_value'].hist(by=df['fetal_health'],
                          bins=np.arange(100,180,10),
                          figsize=(10,10));
len(df["histogram_number_of_zeroes"].value_counts())
pd.crosstab(df.histogram_number_of_zeroes, df.fetal_health)
# Make the crosstab more visual
pd.crosstab(df.prolongued_decelerations,  df.fetal_health).plot(kind="bar",
                                  figsize=(10,6),
                                  color=["yellowgreen", "gold", "firebrick"])

# Add some communication
plt.title("Number of Patients with FHS per Histogram Number of Zeroes")
plt.xlabel("Histogram Tendency")
plt.ylabel("Amount")
plt.legend(["Normal", "Suspect", "Pathologic"])
plt.xticks(rotation=0);
# Make the crosstab more visual
```

```
pd.crosstab(df.fetal_health,  df.prolongued_decelerations).plot(kind="bar",
                                 figsize=(10,6),
                                 color=["yellowgreen", "gold", "firebrick"])

# Add some communication
plt.title("Number of Patients with FHS per Histogram Number of Zeroes")
plt.ylabel("Histogram Tendency")
plt.xlabel("Amount")
plt.xticks(rotation=0);
df['prolongued_decelerations'].hist(by=df['fetal_health'],
                                 bins=np.arange(0,0.0065,0.0005),
                                 figsize=(10,10));
df['abnormal_short_term_variability'].hist(by=df['fetal_health'],
                                 bins=np.arange(0,110,10),
                                 figsize=(10,10));
df['percentage_of_time_with_abnormal_long_term_variability'].hist(by=df['fetal_health'],
                                 bins=np.arange(0,110,10),
                                 figsize=(10,10));
df['accelerations'].hist(by=df['fetal_health'],
                                 bins=np.arange(0,0.0105,0.0005),
                                 figsize=(10,10));
# Let's make our correlation matrix a little prettier
corr_matrix = df.corr()
fig, ax = plt.subplots(figsize=(15,10))
ax = sns.heatmap(corr_matrix,
                 annot=True,
                 linewidths=0.5,
                 fmt=".2f",
                 cmap="YlGnBu");
# Split data into X and y
X = df.drop("fetal_health", axis=1)
y = df["fetal_health"]
# Split data into Train and Test sets
RANDOM_STATE = 42
np.random.seed(42)

X_train, X_test, y_train, y_test = train_test_split(X,
                                                    y,
                                                    test_size=0.2,
                                                    random_state=RANDOM_STATE)
y_train_df = pd.DataFrame(y_train)
y_train_df.value_counts()
y_test_df = pd.DataFrame(y_test)
y_test_df.value_counts()
# Remove Histogram Tendency from our numerical_cols list b/c it is categorical
numerical_cols = list(X_train.columns)
numerical_cols.remove('histogram_tendency')
# Create Normalized Data
```

```
from sklearn.preprocessing import MinMaxScaler

X_train_norm = X_train.copy()
X_test_norm = X_test.copy()

for col in numerical_cols:

    min_max_scaler = MinMaxScaler()
    scale = min_max_scaler.fit(X_train_norm[[col]])
    X_train_norm[col] = scale.fit_transform(X_train_norm[[col]])
    X_test_norm[col] = scale.transform(X_test_norm[[col]])
# Create Standardized Data

from sklearn.preprocessing import StandardScaler

X_train_stand = X_train.copy()
X_test_stand = X_test.copy()

for col in numerical_cols:

    scale = StandardScaler().fit(X_train_stand[[col]])
    X_train_stand[col] = scale.transform(X_train_stand[[col]])
    X_test_stand[col] = scale.transform(X_test_stand[[col]])
# Unaltered data

rf = RandomForestClassifier()
rf.fit(X_train, y_train)
y_preds = rf.predict(X_test)
print('Accuracy:', rf.score(X_test, y_test))
print('F1:', f1_score(y_test, y_preds, average='macro'))
print(confusion_matrix(y_test, y_preds))
test = classification_report(y_test, y_preds, target_names=['Normal', 'Suspect',
'Pathologic'])
print(test)
test = classification_report(y_test, y_preds, target_names=['Normal', 'Suspect',
'Pathologic'], output_dict=True)
test
test2 = pd.DataFrame(test)
test2.T
print(str(test2.T))
# Normalized data

rf.fit(X_train_norm, y_train)
y_preds_norm = rf.predict(X_test_norm)
print('Accuracy:', rf.score(X_test_norm, y_test))
print('F1:', f1_score(y_test, y_preds_norm, average='macro'))
print(confusion_matrix(y_test, y_preds_norm))
# Standardized data

rf.fit(X_train_stand, y_train)
```

```
y_preds_stand = rf.predict(X_test_stand)
print('Accuracy:', rf.score(X_test_stand, y_test))
print('F1:', f1_score(y_test, y_preds_stand, average='macro'))
print(confusion_matrix(y_test, y_preds_stand))# Put models in a dictionary
models = {"Logistic Regression": LogisticRegression(max_iter = 10000),
          "KNN": KNeighborsClassifier(),
          "Random Forest": RandomForestClassifier()}

# Create a function to fit and score models
def fit_and_score(models, X_train, X_test, y_train, y_test):
    """
    Fits and evaluates given machine learning models.
    models : a dict of different Scikit-Learn machine learning models
    X_train : training data (no labels)
    X_test : testing data (no labels)
    y_train : training labels
    y_test : test labels
    """
    # Make a dictionary to keep model scores
    model_scores = {}
    model_scores_f1 = {}
    # Loop through models
    for name, model in models.items():
        # Fit the model to the data
        model.fit(X_train, y_train)
        # Make predictions
        y_preds = model.predict(X_test)
        # Evaluate the model and append its score to model_scores
        model_scores[name] = model.score(X_test, y_test)
        model_scores_f1[name] = f1_score(y_test, y_preds, average='macro')
    return model_scores, model_scores_f1
model_scores, model_scores_f1 = fit_and_score(models, X_train, X_test, y_train, y_test)
print(model_scores)
print(model_scores_f1)
model_compare = pd.DataFrame(model_scores, index=["accuracy"])
model_compare.T.plot.bar();
model_compare_f1 = pd.DataFrame(model_scores_f1, index=["F1 Macro"])
model_compare.T.plot.bar();
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score

rf.fit(X_train, y_train)

# Make some predictions
y_preds = rf.predict(X_test)

# Evaluate the classifier
print("Classifier metrics on the test set")
print(f"Accuracy: {accuracy_score(y_test, y_preds)*100:.2f}%")
print(f"Precision: {precision_score(y_test, y_preds, average='macro')}")
print(f"Recall: {recall_score(y_test, y_preds, average='macro')}")
```

```
print(f"F1: {f1_score(y_test, y_preds, average='macro')}")
from joblib import dump, load
dump(rf, filename="baseline.joblib")
# Create a hyperparameter grid for LogisticRegression
log_reg_grid = {"C": np.logspace(-4, 4, 20),
                "solver": ["liblinear"]}

# Create a hyperparameter grid for RandomForestClassifier
rf_grid = {"n_estimators": np.arange(10, 1000, 50),
           "max_depth": [None, 3, 5, 10],
           "min_samples_split": np.arange(2, 20, 2),
           "min_samples_leaf": np.arange(1, 20, 2)}
# Tune LogisticRegression

# Setup random hyperparameter search for LogisticRegression
rs_log_reg = RandomizedSearchCV(LogisticRegression(),
                                param_distributions=log_reg_grid,
                                cv=5,
                                n_iter=20,
                                scoring='f1_macro',
                                verbose=True)

# Fit random hyperparameter search model for LogisticRegression
rs_log_reg.fit(X_train, y_train)
model_scores
```

Launch.py

```
from flask import Flask, render_template, request, jsonify
from utils import predictSentiments
import json
import requests

app = Flask(__name__)

@app.route("/")
def home():
    return render_template("index.html", title='Home')

@app.route("/login")
def login():
    return render_template("login.html", title='login')

@app.route("/result",methods=['POST'])
def result():
    req_PDs = int(request.form.get('PDs'))
    req_ASTV = int(request.form.get('ASTV'))
    req_POTWALTV = int(request.form.get('POTWALTV'))
```

```
route='/predict'
    url='http://127.0.0.1:5000'+route
    param={ 'PDs': req_PDs,
    'ASTV': req_ASTV,
    'POTWALTV': req_POTWALTV }

    r=requests.post(url,data=param)
    predicts = r.json()


    return render_template("index.html", title='Prediction', PDs = req_PDs, ASTV =
req_ASTV, POTWALTV = req_POTWALTV,
    avis = predicts['Résultat'], proba = predicts['Proba'])

@app.route("/predict",methods=['POST'])
def predict():
    req_PDs = int(request.form.get('PDs'))
    req_ASTV = int(request.form.get('ASTV'))
    req_POTWALTV = int(request.form.get('POTWALTV'))
    predictResult = predictSentiments(req_PDs, req_ASTV, req_POTWALTV)
    if(predictResult[0] == 1.0):
        strAvis = "Normal"
    elif(predictResult[0] == 2.0) :
        strAvis = "Suspected"
    else :
        strAvis = "Pathological"

    return jsonify({'Résultat': strAvis, 'Proba': predictResult[1]})

if __name__== "__main__":
    app.run()
```

B. SCREENSHOTS

Module import ¶

In [1]:
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
from sklearn.model_selection import train_test_split
```

In [2]:
```
df = pd.read_csv("fetal_health.csv")
df.shape
```

Out[2]:
```
(2126, 22)
```

In [3]:
```
df.head()
```

Out[3]:

	baseline_value	accelerations	fetal_movement	uterine_contractions	light_decelerations	severe
0	120	0.000	0.0	0.000	0.000	
1	132	0.006	0.0	0.006	0.003	
2	133	0.003	0.0	0.008	0.003	
3	134	0.003	0.0	0.008	0.003	
4	132	0.007	0.0	0.008	0.000	

5 rows × 22 columns

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

```
df.columns
```

```
Index(['baseline_value', 'accelerations', 'fetal_movement',
       'uterine_contractions', 'light_decelerations', 'severe_decelerations',
       'prolongued_decelerations', 'abnormal_short_term_variability',
       'mean_value_of_short_term_variability',
       'percentage_of_time_with_abnormal_long_term_variability',
       'mean_value_of_long_term_variability', 'histogram_width',
       'histogram_min', 'histogram_max', 'histogram_number_of_peaks',
       'histogram_number_of_zeroes', 'histogram_mode', 'histogram_mean',
       'histogram_median', 'histogram_variance', 'histogram_tendency',
       'fetal_health'],
      dtype='object')
```

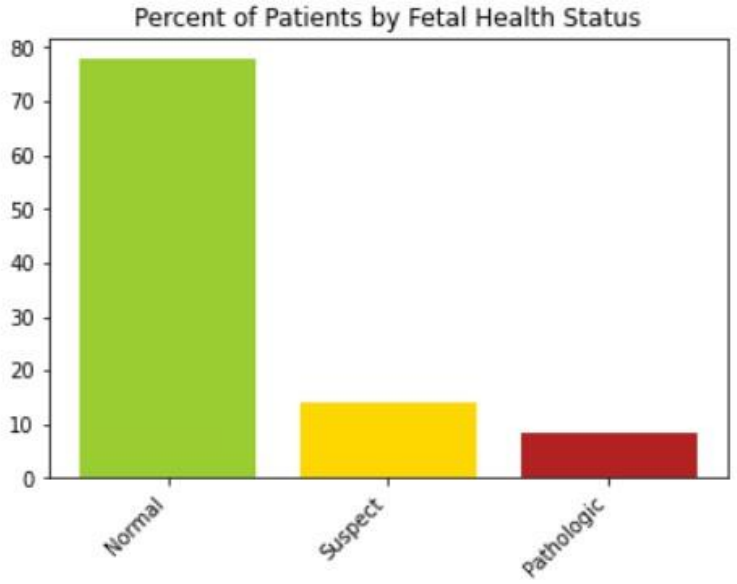
```
# Let's find out how many of each class there are in target column
df["fetal_health"].value_counts()
```

```
1    1655
2     295
3     176
Name: fetal_health, dtype: int64
```

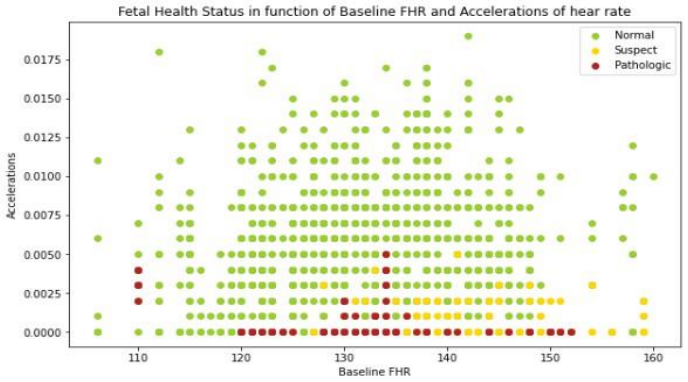
Exploratory Data Analysis

```
: import matplotlib.pyplot as plt
  %matplotlib inline
  import seaborn as sns
```

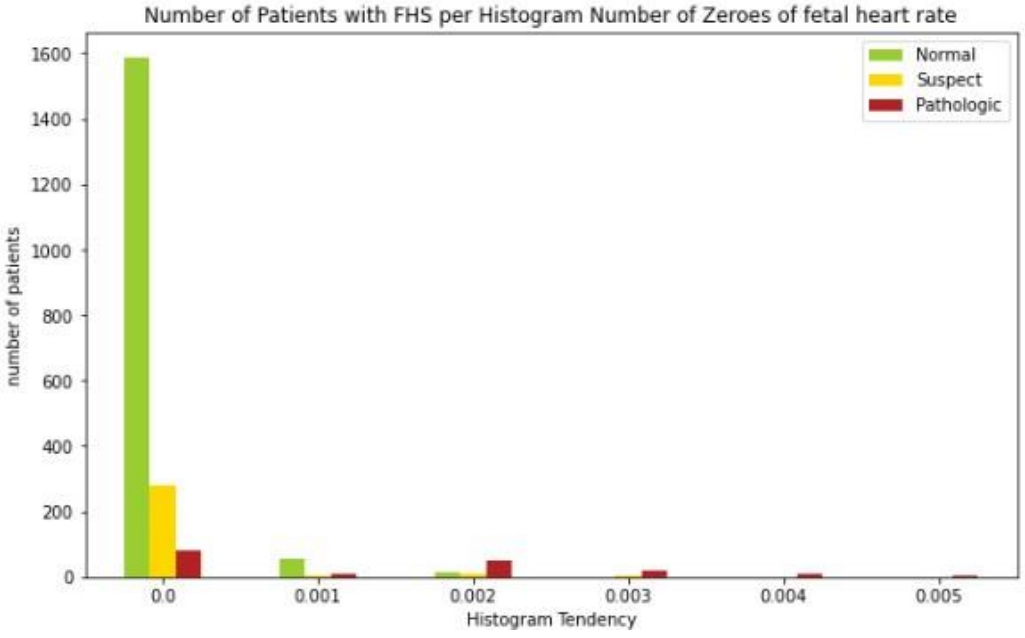
```
: height = (df["fetal_health"].value_counts(1)*100).to_list()
  bars = ('Normal', 'Suspect', 'Pathologic')
  y_pos = np.arange(len(bars))
  plt.bar(y_pos, height, color=('yellowgreen','gold','firebrick'))

  # use the plt.xticks function to custom labels
  plt.xticks(y_pos, bars, rotation=45, horizontalalignment='right')

  # remove labels
  plt.tick_params(labelbottom='off')

  plt.title('Percent of Patients by Fetal Health Status');
```



```
# Create another figure
plt.figure(figsize=(10, 6))

# Scatter with normal examples
plt.scatter(df.baseline_value[df.fetal_health==1],
            df.accelerations[df.fetal_health==1],
            c="yellowgreen")
# Scatter with suspect examples
plt.scatter(df.baseline_value[df.fetal_health==2],
            df.accelerations[df.fetal_health==2],
            c="gold")

# Scatter with pathologic examples
plt.scatter(df.baseline_value[df.fetal_health==3],
            df.accelerations[df.fetal_health==3],
            c="firebrick")

# Add some helpful info
plt.title("Fetal Health Status in function of Baseline FHR and Accelerations of hear rate")
plt.ylabel("Accelerations")
plt.xlabel("Baseline FHR")
plt.legend(["Normal", "Suspect", "Pathologic"]);
```




```
# Make the crosstab more visual
pd.crosstab(df.prolongued_decelerations, df.fetal_health).plot(kind="bar",
                                    figsize=(10,6),
                                    color=["yellowgreen", "gold", "firebrick"])

# Add some communication
plt.title("Number of Patients with FHS per Histogram Number of Zeroes of fetal heart rate")
plt.xlabel("Histogram Tendency")
plt.ylabel("number of patients")
plt.legend(["Normal", "Suspect", "Pathologic"])
plt.xticks(rotation=0);
```



Random Forest Algorithm

```
# Split data into Train and Test sets
RANDOM_STATE = 42
np.random.seed(42)

X_train, X_test, y_train, y_test = train_test_split(X,y,test_size=0.2,
                                                    random_state=RANDOM_STATE)
```

```
y_train_df = pd.DataFrame(y_train)
y_train_df.value_counts()
```

```
fetal_health
1               1322
2                231
3                147
dtype: int64
```

```
y_test_df = pd.DataFrame(y_test)
y_test_df.value_counts()
```

```
fetal_health
1               333
2                64
3                29
dtype: int64
```

```
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import confusion_matrix, classification_report
from sklearn.metrics import precision_score, recall_score, f1_score
```

```
rf = RandomForestClassifier()
rf.fit(X_train, y_train)
y_preds = rf.predict(X_test)
print('Accuracy: {:.2f}'.format(rf.score(X_test, y_test)))
```

```
Accuracy: 0.95
```

Logistic Regression

```
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import f1_score
```

```
from sklearn import metrics
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=0)
logreg = LogisticRegression()
logreg.fit(X_train, y_train)
```

```
C:\Users\DELL\anaconda3\lib\site-packages\sklearn\linear_model\_logistic.py:762: ConvergenceWarning: lbfgs failed to converge
(status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
  n_iter_i = _check_optimize_result(
```

```
LogisticRegression()
```

```
y_pred = logreg.predict(X_test)
print('Accuracy: {:.2f}'.format(logreg.score(X_test, y_test)))
```

```
Accuracy: 0.87
```

KNN Classifier

```
from sklearn.neighbors import KNeighborsClassifier
```

```
from sklearn import metrics
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=0)
knnclas = KNeighborsClassifier()
knnclas.fit(X_train, y_train)
```

```
KNeighborsClassifier()
```

```
y_pred = knnclas.predict(X_test)
print('Accuracy: {:.2f}'.format(knnclas.score(X_test, y_test)))
```

```
Accuracy: 0.89
```

Comparision of all the models

```
# Put models in a dictionary
models = {"Random Forest": RandomForestClassifier(),
          "Logistic Regression": LogisticRegression(max_iter = 10000),
          "KNN": KNeighborsClassifier()}

# Create a function to fit and score models
def fit_and_score(models, X_train, X_test, y_train, y_test):
    # Make a dictionary to keep model scores
    model_scores = {}
    model_scores_f1 = {}
    # Loop through models
    for name, model in models.items():
        # Fit the model to the data
        model.fit(X_train, y_train)
        # Make predictions
        y_preds = model.predict(X_test)
        # Evaluate the model and append its score to model_scores
        model_scores[name] = model.score(X_test, y_test)
        model_scores_f1[name] = f1_score(y_test, y_preds, average='macro')
    return model_scores, model_scores_f1
```

accuracy vs f1 score

```
model_scores, model_scores_f1 = fit_and_score(models, X_train, X_test, y_train, y_test)
print(model_scores)
print(model_scores_f1)
```

```
{'Random Forest': 0.9435736677115988, 'Logistic Regression': 0.8699059561128527, 'KNN': 0.8887147335423198}
{'Random Forest': 0.8945977011494253, 'Logistic Regression': 0.7573037869528179, 'KNN': 0.7924653230620188}
```

Tune KNN

hyperparameter tuning for Knn

```
train_scores = []
test_scores = []

# Create a list of different values for n_neighbors
neighbors = range(1, 21)

# Setup KNN instance
knn = KNeighborsClassifier()

# Loop through different n_neighbors
for i in neighbors:
    knn.set_params(n_neighbors = i)

    # Fit the algorithm
    knn.fit(X_train, y_train)

    # Update the training scores list
    train_scores.append(knn.score(X_train, y_train))

    # Update the test scores list
    test_scores.append(knn.score(X_test, y_test))
```

```
print(f"Maximum KNN accuracy on the test data: {max(test_scores):.2f}%");
```

```
Maximum KNN accuracy on the test data: 0.90%
```

Tune Logistic Regression

hyper parameter tuning with RandomizedSearchCV

```
# Create a hyperparameter grid for LogisticRegression
log_reg_grid = {"C": np.logspace(-4, 4, 20),"solver": ["liblinear"]}
```

```
from sklearn.model_selection import RandomizedSearchCV, GridSearchCV
```

```
# Tune LogisticRegression

# Setup random hyperparameter search for LogisticRegression
rs_log_reg = RandomizedSearchCV(LogisticRegression(),param_distributions=log_reg_grid,
                                cv=5,n_iter=20,scoring='f1_macro',verbose=True)

# Fit random hyperparameter search model for LogisticRegression
rs_log_reg.fit(X_train, y_train)
print('Accuracy:', rs_log_reg.score(X_test, y_test))
```

```
Fitting 5 folds for each of 20 candidates, totalling 100 fits
```

```
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
```

```
Accuracy: 0.7728610093259775
```

```
[Parallel(n_jobs=1)]: Done 100 out of 100 | elapsed:    6.6s finished
```

Tune Random Forest

using GridsearchCV

```
rf_grid = {"n_estimators": np.arange(10, 1000, 50),
           "max_depth": [None, 3, 5, 10],
           "min_samples_split": np.arange(2, 20, 2),
           "min_samples_leaf": np.arange(1, 20, 2)}
```

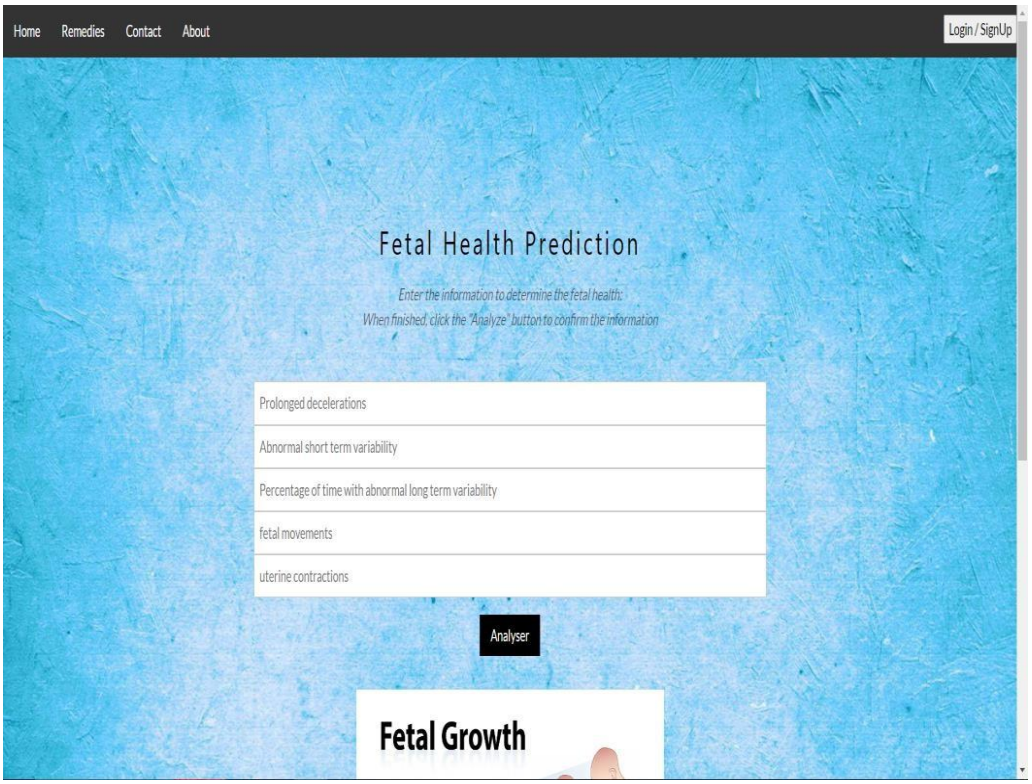
```
# Setup random hyperparameter search for RandomForestClassifier
rs_rf = RandomizedSearchCV(RandomForestClassifier(),
                           param_distributions=rf_grid,cv=5,n_iter=20,
                           scoring='f1_macro',verbose=True)

# Fit random hyperparameter search model for RandomForestClassifier
rs_rf.fit(X_train, y_train)
rs_rf.score(X_test, y_test)
```

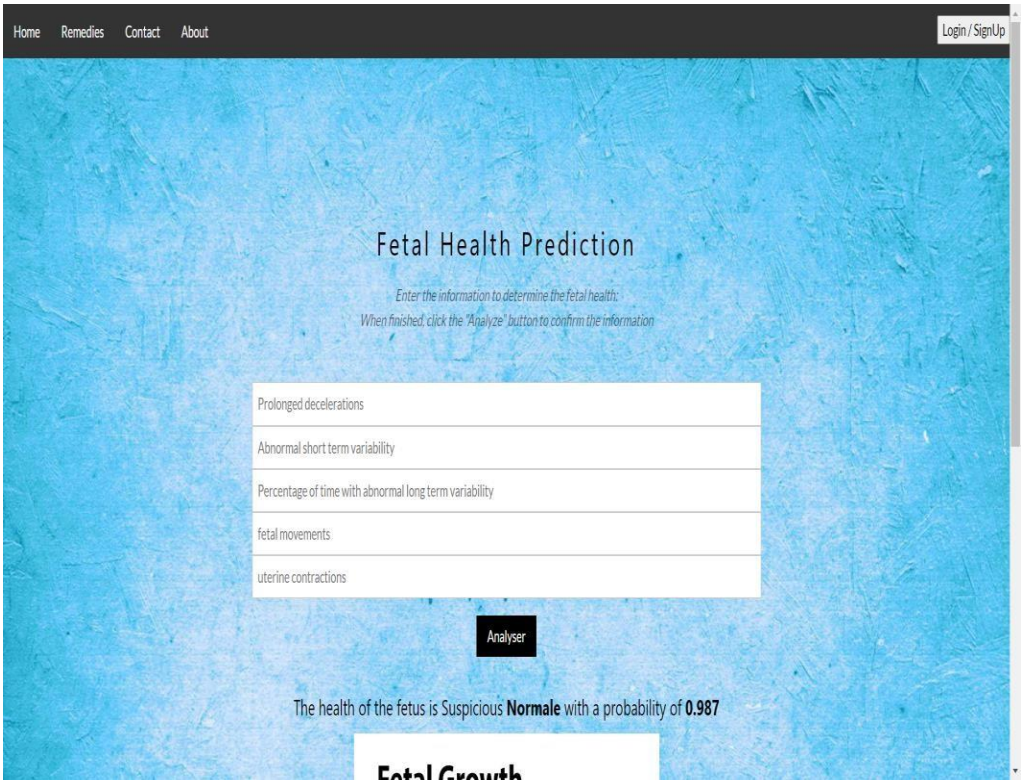
```
Fitting 5 folds for each of 20 candidates, totalling 100 fits
```

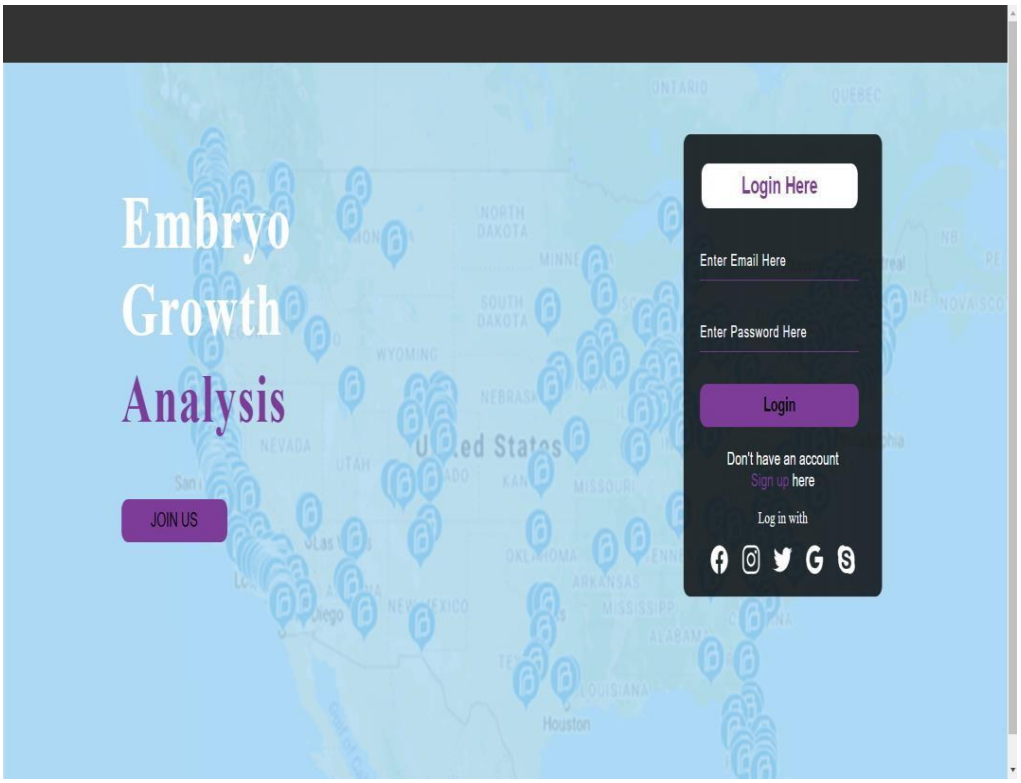
```
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done 100 out of 100 | elapsed:  3.6min finished
```

```
0.8903181971968475
```

accuracy is not better than before tuned score So we find that random forest is the best suited algorithm


Home
Remedies
Contact
About
Login / SignUp
Fetal Health Prediction
Enter the information to determine the fetal health:
When finished, click the "Analyze" button to confirm the information
Prolonged decelerations
Abnormal short term variability
Percentage of time with abnormal long term variability
fetal movements
uterine contractions
Analyser
Fetal Growth


Home
Remedies
Contact
About
Login / SignUp
Fetal Health Prediction
Enter the information to determine the fetal health:
When finished, click the "Analyze" button to confirm the information
Prolonged decelerations
Abnormal short term variability
Percentage of time with abnormal long term variability
fetal movements
uterine contractions
Analyser
The health of the fetus is Suspicious Normale with a probability of 0.987


Embryo
Growth
Analysis
JOIN US
Login Here
Enter Email Here
Enter Password Here
Login
Don't have an account
Sign up here
Log in with

C. RESEARCH PAPER

Embryo Growth Analysis using Algorithmic Approach

Nehal Veeramachaneni
vn001030@gmail.com
Department of CSE
Sathyabama Institute of
Science and Technology, Chennai

Jinugu Nikhil Kumar
nikhilkumarjinugu@gmail.com
Department of CSE
Sathyabama Institute of
Science and Technology, Chennai

Dr. N. Srinivasan
srinijyothish@gmail.com
Department of CSE
Sathyabama Institute of
Science and Technology, Chennai

Abstract- In order to have a successful pregnancy and ensure long-term wellness for the unborn child, normal embryo growth is crucial. However, differentiating between normal and aberrant prenatal growth has proven to be difficult in clinical practice and research. This article discusses the sourcesand standards commonly used to evaluateprenatal growth, as well as the limitations of existing definitions of abnormal embryo growth. It also explores the pros and cons of various approaches to changing embryo growth standards and highlights recent advances in understanding embryo growth restriction. With the global trend towards single embryo transfer, accurate methods of evaluating embryo viability are increasingly important. The physical characteristics of the human preimplantation embryo are examined in detail, along with how they relate to prospective development. The potential advantages of a more quantitative approach to embryo selection are also examined, alongwith computer-assisted examination of embryo morphology and the influence of culture environment on morphological features. A comprehensive definition of abnormal embryo growth incorporating factors such as genetics, biophysical findings, maternal and embryo Doppler velocimetry, embryohealth status, and embryo growth is suggested, but further investigation and testing is necessary.

Keywords— Embryo growth, Embryo culture, morphology, single embryo transfer, embryo Doppler Velocimetry, Random ForestAlgorithm, EDA.

INTRODUCTION

The healthcare industry plays a vital role globally, impacting people's lives in various ways. Thanks to recent medical advancements, such as machine learning techniques, the number of deaths caused by diseases like cancer and heart disease has decreased, and life expectancy has increased. However, in many low-income countries, healthcare is inadequate, leading

to recurring fatalities.Machine learning has been successfully applied to many essential disciplines, including finance, governance, and healthcare.

Fetal age, size, and weight				
months after conception	crown-rump length		weight	
	mm	inches	grams	ounces or pounds
2	28	1	2.25	0.75 oz
3	75	3	25	1 oz
4	135	5.3	170	6 oz
5	185	7.3	440	14 oz
6	225	9	820	1.75 lb
7	270	10.6	1,380	3 lb
8	310	12.2	2,220	5 lb
9	360	14.2	3,150	7 lb

In medicine, it has been used in various applications, such as embryo growth and abnormalities, medical image analysis, computer vision, and cancer diagnosis. In this study, the focus is on embryo abnormalities, anda better machine learning classification method is recommended for early detection and prediction. Ultrasonic sound waves are used to monitor embryo growth during prenatal delivery, and prenatal facial anomaly identification can identify chromosomal abnormalities and genotypic illnesses, making parent preparation and counseling more suitable. Using machine learning methods can help scientists identify and anticipate embryo issues before they arise. Recent studies have shown significant improvements in embryo anomaly diagnosis and prognosis throughout the first trimester of pregnancy.

Figure 1:- Ultrasound scan during pregnancy

According to recent studies, embryo abnormalities occur in 3-5% of all of her pregnancies. In India, the suggested time frame for care for embryo anatomy examinationsis the second trimester (between 18 and 22weeks). From two to nine months of pregnancy, the table shows the usual length and weight of the baby.

Figure 2:- Length and weight of a baby during gestation period

LITERATURE REVIEW

In this set of papers, there is a common theme of using data and machine learning to improve healthcare outcomes. The first paper by Lesley McCowan et al. focuses on the management of suspected embryo growth restriction, and summarizes areas of consensus and controversy between recently published national guidelines. The second paper by Michal Rabijewski discusses the importance of embryo growth assessment and thecreation and application of growth charts.

The third paper by K S Joseph et al. analyzes the need for separate embryo growth standards for singletons and twins based on birth weight range and gestational age.

Finally, the paper by Rohit Kumar Singh Gautam and Er. Amit Doegar presents an ensemble approach for DDOS attack detection system using machine learning algorithms, which was presented at a cloud computing and data science conference.

Overall, these papers highlight the growing importance of data and machine learning in healthcare and its potential to improve outcomes and patient care.

Presents an ensemble approach for DDOS attack detection system using machine learning algorithms, which was presented at a cloud computing and data science conference.

Overall, these papers highlight the growing importance of data and machine learning in healthcare and its potential to improve outcomes and patient care

Areas for future study that could be prioritized based on the presented literature review are:

Further research on the prediction of abnormal embryo growth:

The review highlighted the importance of accurate prediction of abnormal embryo growth in reducing perinatal morbidity and mortality. Future studies could focus on improving the accuracy of prediction models for abnormalembryo growth using machine learning techniques.

Investigating the maternal health's impact onembryo development:

The reviewed articles emphasized the importance of maternal health and lifestyle factors on embryo growth. To create efficient therapies to promote ideal embryo growth, further research may be done to better understand how these factors affect embryo growth.

Comparison of embryo growth standards in singleton and twin pregnancies:

One of the reviewed articles analyzed whether singleton and twin pregnancies require separate embryo growth standards. Future research could compare and evaluate different embryo growth standards in both singleton and twin pregnancies to determine if separate standards are needed.

Analysis of the long-term effects of abnormalembryo growth:

The reviewed articles discussed the short-term consequences of abnormal embryo growth, such as perinatal morbidity and mortality. However,long-term effects on the health and development of the child may also be important to consider. Future studies could investigate the potential long-term effects of abnormal embryo growth on the child's health and development.

Evaluation of machine learning techniques for embryo anomaly detection: The literature review emphasized the importance of accurate detection of embryo anomalies to reduce the incidence rate. Future research could evaluate and compare the effectiveness of different machine learning techniques for embryo anomalydetection and prognosis.

Overall, the presented literature review highlights the importance of accurate prediction and detection of embryo growth abnormalities and the impact of maternal health on embryo growth. Future studies could focus on improving prediction and detection models, evaluating different embryo growth standards, investigating the long-term effects of abnormal embryo growth, and evaluating machine learning techniques for embryo anomalydetection.

EMBRYO DEVELOPMENT CYCLE

The following illustrates the stages of the embryo's development during the course of its weeks.

week-by-week development of the

embryo goes like :

Week 3 - gastrulation and the growth of the three germ layers takes place in embryo

Week 4 - The heart becomes active earliest among all the organs.

Week 5 – this week the development of ear starts and size up to 4mm.

Week 6 – eyes and nose starts developing

Week 7 – lungs, arms and legs starts developing

Week 8 – The majority of the body's organs, including the hair follicles, external ears, eyelashes, and eyelids, start to develop. The body also starts to cover itself in hair follicles.

The diagram below displays the embryo's growth stages from week three to week eight, indicating the development of various organs and body parts at each stage.

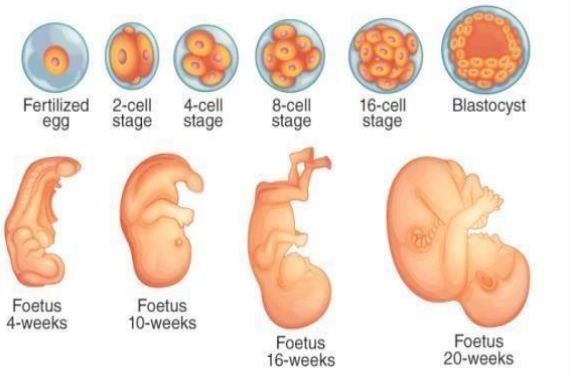


Figure 3:- Embryo development stages

EXISTING TECHNOLOGY

The use of personalized birth weight norms that consider embryo growth potential has been developed, and many wealthy countries routinely perform prenatal testing for abnormalities and conduct multiple ultrasound scans.

Ultrasound technology has evolved significantly since the 1980s and can aid in the earlier diagnosis of multiple pregnancies and accurate determination of gestational age, reducing the likelihood of initiating labor in post-dated pregnancies.

The ultrasonography EFW curves are adjusted higher or lower basedon specific maternal and embryo traits, including the anticipated embryo weight at 40 weeks coupled with the birth weight data to account for individual traits and gestational week.

Figure 4:- Diet tips for healthy pregnancy

Based on the birth weight at 40 weeks, the ultrasonography EFW curves were correspondingly adjusted either higher or lower for specific maternal and embryo characteristics.

RISK OF ULTRASOUND SCANS

During prenatal care, ultrasound scans are commonly utilized to monitor the development of the fetus and detect any potential problems. Despite the general belief that ultrasounds are safe, it is important to inform pregnant women about potential adverse effects. One of the possible negative consequences is the heating of tissues due to

sound waves during the ultrasound scans. While this heating effect is generally considered to be insignificant, it can sometimes cause harm to embryonic tissues. It is recommended that only skilled professionals should perform ultrasound scans with appropriate equipment and procedures to minimize this risk.

The formation of small bubbles or hollow spaces in embryo tissues, referred to as cavitation, is a potential negative outcome of ultrasound scans. According to various studies, cavitation can lead to cellular harm or other negative impacts on embryo growth, though the long-term consequences are not fully known.

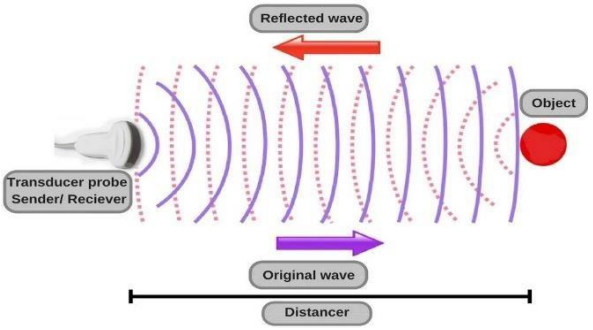


Fig 5:- Ultrasound object detection

In addition to these physical risks, some pregnant women may experience mental stress or anxiety due to ultrasound scans. This can be caused by confusion about the results of the scan or concerns about potential future issues. To minimize this risk, healthcare professionals should provide patients with clear and accurate information about the purpose of the scan, as well as any potential benefits and risks.

While ultrasound scans are generally considered safe and efficient for monitoring embryo growth, pregnant women should be mindful of the possible adverse effects and communicate any concerns with their healthcare providers. By working collaboratively, clinicians and patients can ensure that ultrasound scans are utilized appropriately and efficiently to provide optimal care for both mothers and infants

DISADVANTAGES OF EXISTING TECHNOLOGY

During the first trimester of pregnancy, ultrasounds are conducted to determine the number of fetuses and to check for any potential issues. One reliable indication of chromosomal abnormalities is Nuchal Translucency (NT), which can be seen between 11 and 14 weeks of pregnancy. Embryo malformations can be diagnosed and foundduring this trimester, with various abnormalities that can occur.

Embryo growth restrictions (FGR) can result in various significant consequences, such as infections, respiratory issues, stillbirths, and fatalities. Clinicians face challenges in diagnosing and assessing FGR because most cases occur in pregnancies with no known risk factors. Consequently, a diverse approach to diagnosisand assessment is necessary. More research isneeded to clarify preventive or treatment measures for growth-restricted fetuses.

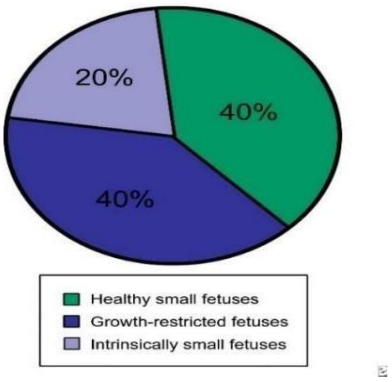


Figure 6:- Healthy vs Abnormal Fetus Pie

Chart

PROPOSED MODEL

This design establishes how the software system is organized overall. The data for each parameter was adjusted to have a mean of zero and a standard deviation of one based on the training data in order to perform precise analysis. Equal numbers of samples from each category were included in the testing dataset. It is important to note that during the model training process, oversampling was only carried out on the training data and not on the testing data.

Data preprocessing module:

As the data had already been well cleaned when the dataset was acquired, this project just needed a small amount of cleaning. Performed some outlier detection, but decided to keep any already-existing outliers because, given the size of the dataset and the reliability of the data collection, they might be crucial to the study.

Exploratory Data Analysis (EDA) Module:

EDA is used to depict the relationship betweenthe attributes based on the dataset that is taken into consideration (ex: embryo movementsor number of contractions with embryo health etc).
The translation of the target column (Embryo health) is as follows:
1.0 -> Normal
2.0 -> Suspect
3.0 -> Severe

Baseline Models:

Random Forest Regressor, Logistic Regression, and Knn Classifier are the three models being built. Baseline models are assessed using criteria like as precision, recall, and accuracy.

Predict Embryo Growth:

This module covers how embryo growth that is within normal limits affects both the health of the developing fetus and the mother during pregnancy.

ADVANTAGES OF PROPOSED MODEL

The study employed a model that incorporated demographic data, embryo biometry, and Dopplerreadings, which were associated with gestational age. The model was further improved by including GA as an input parameter that accounted for embryo measurements and maternal demographics.

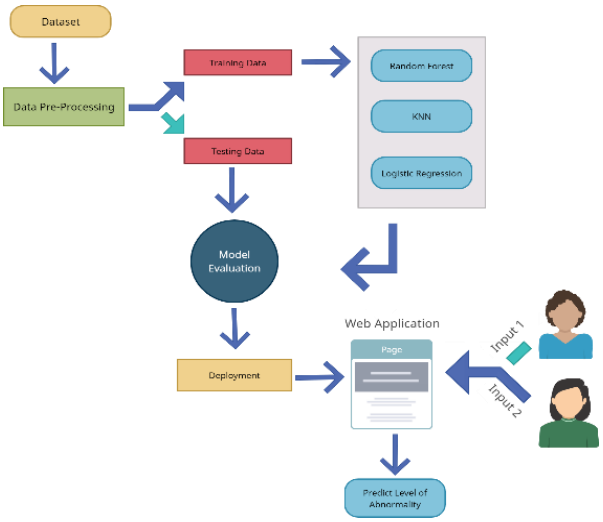
The machine learning models examined provided accurate results and saved time. However, in predicting embryonic growth issues, the machine learning methods used in the study did not perform better than Random forest.

DISTINCTIVENESS OF PROPOSED MODEL COMPARED TO EXISTING SYSTEM

The existing system uses customized birthweight norms based on maternal and embryo physiological parameters to calculate optimal embryo weight at each gestational week. In contrast, the proposed model discusses recent advances towards an integrated definition for embryo growth restriction, which may incorporate embryo size with the status of placental health measured by maternal and embryo Doppler velocimetry and biomarkers, biophysical findings, and genetics.

The disadvantage of the existing system is that FGR can cause many serious complications,

and still births and death may occur. On the



other hand, the proposed model advantage is that the machine learning methods used in this study offered advantage over logistic regression in the prediction of embryo growth abnormalities.

ARCHITECTURE OF THE PROPOSED MODEL

The process of designing a system's subsystems and their control and communication framework is called system architecture design. This design determines the general structure of the software system. The data was classified into a training and validation set of data (80% of the considered data) and a testing set (20% of the considered data).

To ensure accurate analysis, the number of samples in each class in the testing dataset was balanced, and the data for every parameter was scaled to have a Using the distribution of the training data, a zero mean and a single standard deviation are used. To ensure equal sample sizes in both classes, the training data from the smaller class was randomly duplicated (oversampling).

This was essential to improve the model's accuracy. It should be noted that only the training data was oversampled for the model training procedure,not the testing data.

Figure 7:- Architecture of model

ETHICAL AND LEGAL ASPECTS

To safeguard the rights and wellbeing of both the mother and the fetus, it is crucial to take ethical and legal considerations into account when making predictions regarding the health of a fetus. The precision and dependability of the prediction process is one important concern.

The methods employed must be validated and dependable, and the data used must be accurate and dependable in order to guarantee that judgements based on forecasts are valid.

Unneeded medical procedures, excessive stress and anxiety for the mother, or even injury to the fetus could all be the outcome of inaccurate projections. As a result, it is crucial to guarantee the precision and dependability of the prediction process.

Fig 8:- Levels of ethical standards

When determining the fetus's health, the privacy and confidentiality of the mother's personal and medical information should be upheld. This involves preserving the confidentiality of the information used to make the prediction and making sure the outcomes are only shared with people who have been given the required permission. To preserve the mother's rights and uphold the ethical standards of the prediction process, it is crucial to ensure the privacy of her personal information.

The care and treatment given to both the mother and the fetus can be greatly impacted by predictions made about the health of the fetus. As a result, it's crucial to make sure the mother is well aware of the consequences of the forecasts and that she is presented with and has a chance to discuss the best care and treatment alternatives.

It's crucial to abide by all legal and regulatory requirements while forming assumptions regarding the fetus's health. For instance, in some jurisdictions, such projections might be constrained by laws and regulations like those governing informed consent and data protection. To ensure that these legal and regulatory criteria are met during the prediction process, it is vital to take the required precautions.

To guarantee responsible and ethical activities, embryo health prediction entails significant ethical and legal issues. These consist of being conscious of the dangers of prejudice and bias, obtaining informed consent, ensuring the reliability and accuracy of predictions, protecting information privacy and confidentiality, taking into account the implications of predictions for care and treatment, and abiding by legal and regulatory standards.

DATASET EXPLAINATION

The dataset contains various measurements related to embryo health during labor, including the embryo heart rate (FHR) baseline in beats per minute, the number of accelerations and decelerations per second, the number of uterine contractions per second, and the percentage of time with abnormal short and long-term variability.

Additionally, the dataset includes measurements related to the FHR histogram, such as the width, minimum, maximum, number of peaks and zeros, mode, mean, median, variance, and tendency. To show how frequently embryo heart rate is present among the population, a histogram is used.
Embryo heart rate varies between 130 and 140 beats per minute, with an 800-member frequency.

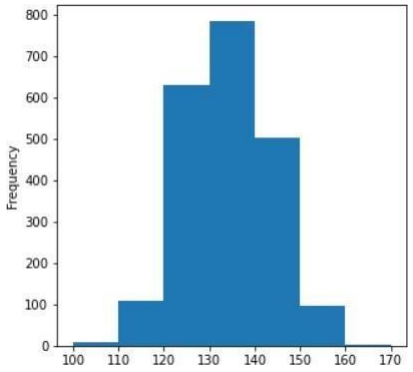


Figure 9:- Histogram of embryo heart rate

The growth, well-being, and interaction of the fetus in the uterus of a pregnant woman is referred to as embryo health. The embryo health is made up of three main types of values, which is classified as normal, suspect, or pathologic. The frequency of embryo health in the dataset is shown by

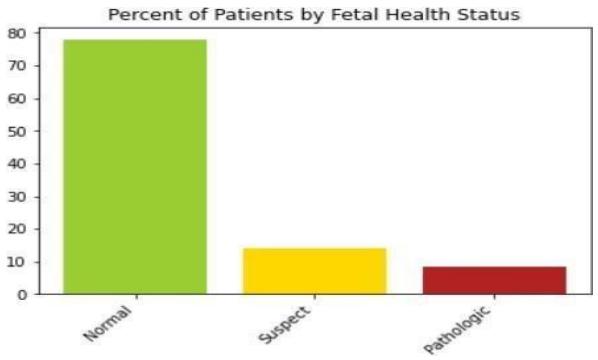


Figure 10:- Frequency of embryo health

ALGORITHMS/TECHNIQUES USED

Logistic Regression:

A statistical method called logistic regression uses independent variables to predict the likelihood of an event. The dependent variableranges from 0 to 1, as it represents a probability. Logistic regression transforms the odds or chances of success divided by the likelihood of failure using the logit formula. This formula is represented by equations that include variables such as the input value (x), the predicted output (y), the bias or intercept term (B0), and the coefficient for input x (B1). The value is classed as one kind if it is over a particular threshold and as another type if it is below the threshold. This classification is basedon a sigmoid function graph.

$$y = \frac{e^{(b_0 + b_1 X)}}{1 + e^{(b_0 + b_1 X)}}$$

Figure 11:- Sigmoid Function

If the value is above the threshold, it is recognized as belonging to one kind; otherwise, if the value is below the threshold, it is classified into another type. The classification will be based on the graph that is created following the calculation of a sigmoid function.

KNN Classifier:

If we have two categories, A and B, and a new data point called x1, we can use the K-NN method to determine which category it belongs to. K-NN makes finding the class or category of a given dataset is simple. The distance in Euclid units was calculated in orderto identify the closest neighbours.

The two nearest neighbours were included in group B, whereas the three closest neighbours were included in category A. We can quickly ascertain the category of our input and use that as the output by computing the Euclidean distance between the nearest nearby points.

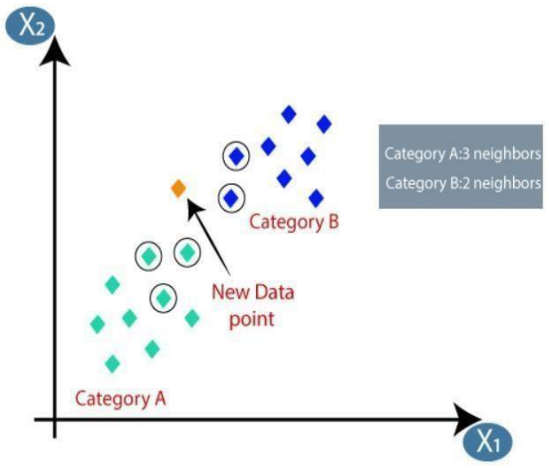


Figure 12:- KNN Classification

Random Forest Classifier:

The algorithm of random forest comprises numerous decision trees and is taught utilizing a method named bagging, which is also called bootstrap aggregating. Bagging is a technique that improves the accuracy of machine learning algorithms by combining multiple

models.

Random Forest is a classification method that utilizes several decision trees on various parts of the dataset to improve the predictive accuracy of the model. It takes the average of the predictions made by each tree and considers the majority vote of predictions from all the trees to generate the final result instead of relying on just one tree.

A random forest is a type of meta estimator that improves the accuracy of predictions and reduces overfitting by averaging multiple decision tree classifiers that are trained on various subsets of the dataset. During the model training process, the most important parameters to consider are "min_samples_split", "max_features", and "random_state".

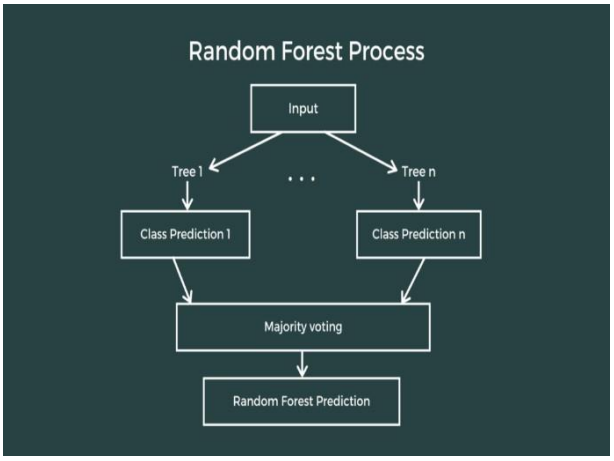


Figure 13:- Random Forest Classification

Hyperparameter tuning using RandomizedSearch:

Hyperparameter optimization, also called hyperparameter tuning, refers to the process of selecting the best possible combination of hyperparameters for a learning system. A hyperparameter is a parameter that regulates the learning process's behavior.

Cross- validation, also known as "CV" in Randomized Search CV, is an essential step in the optimization process.

It employs the resampling technique of using data chunks from outside the sample to test the model's generalization capacity. A three-fold cross- validation technique will be used, and the best parameters will be determined based on the model's fitted training data.

Hyperparameter tuning using GridSearchCV:

GridSearchCV is a method used to identify the best configuration for a particular model by adjusting hyperparameters. As previously stated, the performance of a model is heavily influenced by the values of its hyperparameters. Since it is difficult to predict the optimal hyperparameter values beforehand, it is advisable to test all potential values before deciding on the best ones. Manual hyperparameter tuning can be time-consuming and resource-intensive, so GridSearchCV is employed to automate the process.

Performance Analysis

The evaluation of embryo health depends on three primary values and their frequency of occurrence. To determine the health status, logistic regression, knn classifier, and random forest algorithms were employed. Among these, the random forest algorithm demonstrated the highest accuracy rate of 95%.

The data provided indicates that the random forest model is the most accurate in predicting embryo health, and the predictions closely matched the actual test data. These algorithms can be adjusted to improve their performance further, and the results are compared based on accuracy score, precision, and recall.

It was observed that the random forest algorithm yielded satisfactory accuracy when applied to the dataset compared to logistic regression and knn classifier models.

```
{'Logistic Regression': 0.8943661971830986,
 'KNN': 0.8732394366197183,
 'Random Forest': 0.9460093896713615}
```

Figure 14:- Accuracy of models

The cumulative accuracy of the three models is represented on the graph as follows:

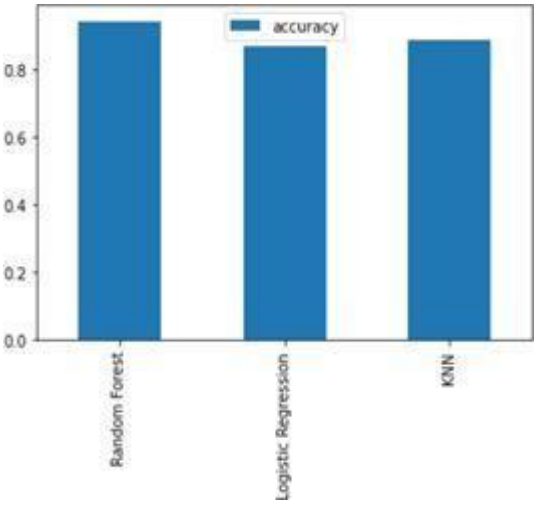


Figure 15:- Graphical view of models accuracy

BENEFITS AND COST SAVINGS

The use of the proposed machine learning model to anticipate anomalies in embryo growth rate without the requirement for ultrasound scans could result in cost savings or other benefits.

To save money for patients and healthcare systems, healthcare practitioners can reduce the need for expensive ultrasound scans by using a machine learning model that predicts embryo growth anomalies. Some insurance plans do not cover ultrasound scans, which can be costly, so implementing this technology can help cut unnecessary expenses.

The availability of ultrasound machines and trained personnel can cause pregnant women to experience wait times of several weeks before they can schedule an appointment. To address this issue, healthcare professionals can use a machine learning model to predict potential anomalies in the fetal development rate, which can reduce the need for frequent ultrasound exams and lead to faster and more effective care for pregnant women.

The use of ultrasound scans during pregnancy is generally considered to pose a low risk. However, some pregnant women may decide to avoid them altogether. To reduce exposure to any potential risks associated with ultrasound technology, healthcare providers can consider using a machine learning model to identify embryo development issues. This could help to reduce the need for routine ultrasound scans. By using machine learning models, it is possible to predict abnormalities in the growth rate of a fetus with greater accuracy compared to traditional ultrasound measurements. This could lead to a reduction in the need for further testing or follow-up scans. In addition to saving costs, this approach may also result in more efficient and effective prenatal care.

CONCLUSION

Complications during pregnancy can result in severe difficulties for the fetus, hindering proper growth and leading to deficiencies or even death. To promote healthy embryo development, it is beneficial to predict the likelihood of potential risks before they occur during pregnancy. This approach can help ensure a safe and healthy pregnancy, allowing the fetus to grow appropriately. This study discusses the various methods and investigations that have been conducted to accurately forecast embryo growth condition using pre- classified patterns. Developing a predictive classifier model using machine learning algorithms is essential for predicting embryo health accurately.

REFERENCES:

1. Ali Gholipour, Estroff JudyA, Barnewolt CarolE, Connolly SusanA, Warfield SimonK. MRI segmentation and volumetric reconstruction for measuring embryo brain volume. Int. J. Comput. Assist. Radiol. Surg. 2011;6(3):329–39.

2. Anton-Rodriguez J. M, David Russell, Peter Julyan, and Ibrahim Djoukhadar D. Gareth Evans, Alan Jackson, and 2019: Julian C. "Comparison of a Standard Resolution PET-CTScanner With an HRRT Brain Scanner for Imaging Small Tumors Within the Head," was the title of the Matthews' article that published.

3. Optimized Machine Learning Approach for the Prediction of Diabetes-Mellitus, Challa M.,Chinnaiyan R. 2020. Smys, S., Tavares, J., Balas, and A. Iliyasu. (eds) Computational Vision and BioInspired Computing. ICCVBIC 2019.. Springer, Cham

4. G. Sabarmathi and R. 2020 Fourth International Conference on Computing Methodologies and Communication (ICCMC),Erode, India, Chinnaiyan, "Big Data Analytics Framework for Opinion Mining of Patient Health Care Experience," pp. 352–357.

5. Van Leeuwen, P., Lange, S., Bettermann, H., Grönemeyer, D. and Hatzmann, W., 1999. Embryo heart rate variability and complexity in the course of pregnancy. *Early human development*, *54*(3), pp.259-269.

6. Liston R, Sawchuck D, Young D, Brassard N, Campbell K, Davies G, Ehman W, Farine D, Farquharson D, Hamilton E, Helewa M. Embryo health surveillance: antepartum and intrapartum consensus guideline. Journal of obstetrics and gynaecology Canada. 2007 Sep 1;29(9):S