NUTRITION DETECTION DURING GESTATION PERIOD USING ML ALGORITHMS

Shreyas Vinayak Patil, Yash Harish Gupta
Dept. of Computer Science and Engineering, BMS Institute of Technology and Management
Bangalore, India
Dept. of Computer Science and Engineering, Kalinga University
Raipur, India

ABSTRACT: The issue of low birth weight in infants is a significant concern in prenatal care, and it can have adverse effects on the newborn's health, sometimes leading to mortality. This problem contributes to high rates of child mortality worldwide. Artificial intelligence, particularly ML, offers potential solutions for predicting whether a fetus will be born small for its gestational age. Early detection of fetal developmental issues is critical, as timely intervention can increase gestation days and improve fetal weight at birth, reducing the risk of neonatal morbidity and mortality. This proposal aims to explore various machine learning methods for predicting small-for-gestational-age infants, emphasizing the importance of early detection and intervention to improve outcomes.
Keywords: Machine learning, gestation, Fetal weight, Cardiotocography

INTRODUCTION

The current trend indicates an increase in the number of newborns with low birth weight, which can be attributed to Intrauterine Growth Restriction (IUGR). IUGR is a condition in which the fetus is smaller in size compared to others at the same gestational age, and babies with IUGR are at a higher risk of facing issues such as low oxygen levels, low Apgar scores, respiratory difficulties, meconium aspiration, and hypoglycemia during birth.

Severe cases may even lead to fetal death or long-term growth problems. Maternal hypertension is a leading cause of IUGR, but identifying the different types of hypertension during the gestation-puerperal cycle remains a challenge. However, machine learning (ML) approaches are increasingly being used in the healthcare sector to tackle this issue. ML is a subset of Artificial Intelligence (AI) that uses large-scale data to identify patterns and trends that may not be apparent to humans. It enables systems to learn from experience, correct parameters, and identify patterns automatically. The larger the database, the better the ML system's performance.

By analyzing significant amounts of data collected from pregnant women using innovative ML techniques, obstetricians and gynecologists can obtain detailed information on fetal health during and after gestation, helping them identify the increased risk of complications due to low birth weight and provide personalized treatment accordingly.

LITERATURE SURVEY

R. Gupta et al.[1] (2022) proposed the use of machine learning techniques to forecast preterm birth using clinical and demographic information.

A. Singh et al.[2] (2022) proposed an automated detection system for identifying preeclampsia through machine learning algorithms.

K. Patel et al[3] (2022) proposed a study where it was found that several factors like maternal age, gestational age, maternal weight gain during pregnancy, maternal body mass index, maternal education, and socio-economic status.

R. Kumar et al[4] (2022) proposed for predicting maternal zinc status during pregnancy using clinical and demographic data.

L. Li et al[5] (2022). proposed deep learning models to predict the iron levels of 10,000 pregnant women during pregnancy using demographic and clinical information.

S. Kaur et al[6] (2022) proposed and assessed machine learning techniques to automatically detect Intrauterine Growth Restriction (IUGR) by utilizing clinical and demographic information.

S. Gupta et al[7] (2022) proposed and assessed machine learning models to identify factors related to preterm labor by employing clinical and demographic information.

A. Sharma et al[8] (2022) proposed and assessed machine learning algorithms to automatically detect Gestational Diabetes Mellitus (GDM) by utilizing clinical and demographic data.

X. Wang et al[9] (2022) proposed and evaluated deep learning models to forecast maternal calcium levels throughout pregnancy by employing clinical and demographic information from 10,000 pregnant women.

P. Singh et al[10] (2022) proposed and examined the effectiveness of machine learning algorithms in identifying maternal hypertension based on clinical and demographic information.

J. Lee et al[11] (2021) suggested a machine learning method to forecast maternal GDM by utilizing clinical and demographic information obtained from 5,000 expectant mothers.

C. Wang et al[12] (2021) proposed and evaluated a deep learning model that was assessed for its effectiveness in predicting maternal nutrient consumption during pregnancy.

S. Roy et al[13] (2021) proposed and evaluated a machine learning technique to predict gestational hypertension using demographic and clinical data from a sample of 8,000 pregnant women.

N. Singh et al[14] (2021) proposed and evaluated a machine learning method to predict maternal hemoglobin levels during pregnancy.If the levels are low, it can result in unfavorable pregnancy outcomes such as low birth weight and preterm birth..
M. Gupta et al[15] (2021) proposed and evaluated a machine learning method to forecast the vitamin D status of pregnant women based on their clinical and demographic data.

METHODOLOGY

In this paper four machine learning models were used:
1) Logistic Regression (LR)
2) K-nearest neighbors (KNN)
3) Random Forest (RF)
4) Support Vector Machine(SVM)
In order to improve the performance of the four machine learning models, there was a need to search for the optimal set of hyperparameters using the common technique of "Grid search". Hyperparameters are parameters that need to be specified before the learning phase and are specific to each model. They cannot be inferred automatically by the learning algorithm and therefore need to be optimized in order to achieve proper generalization. To do this, firstly identify a set of suitable hyperparameters and specify a range of candidate values for each.Next step is to train a model for each possible combination of hyperparameters and estimate its out-of-sample performance using 5-fold Cross Validation (CV). GridSearchCV is used to exhaustively search through all possible combinations of hyperparameters during training. In order to avoid overfitting, cross-validation techniques are employed, utilizing n-splits=3 and n_jobs=2 for faster processing.

1) Logistic Regression:

Logistic regression is a type of supervised machine learning algorithm that is primarily used for classification problems. In a classification problem, the target variable can only take on discrete values for a given set of inputs. Despite its name, logistic regression is actually a regression model that predicts the probability that a given data point belongs to a specific category. This is achieved by modeling the data using the sigmoid function, which is similar to how linear regression models data using a linear function.

$$p(x) = \frac{1}{1 + e^{-(x-\mu)/s}}$$

Figure 1: Logistic Regression Formula

Impurity	Task	Formula	Description
Gini impurity	Classification	$\sum_{i=1}^{C} f_i(1-f_i)$	f_i is the frequency of label i at a node and C is the number of unique labels.
Entropy	Classification	$\sum_{i=1}^{C} -f_i \log(f_i)$	f_i is the frequency of label i at a node and C is the number of unique labels.
Variance / Mean Square Error (MSE)	Regression	$\frac{1}{N}\sum_{i=1}^{N}(y_i-\mu)^2$	y_i is label for an instance, N is the number of instances and μ is the mean given by $\frac{1}{N}\sum_{i=1}^{N} y_i$
Variance / Mean Absolute Error (MAE) (Scikit-learn only)	Regression	$\frac{1}{N}\sum_{i=1}^{N}\lvert y_i-\mu\rvert$	y_i is label for an instance, N is the number of instances and μ is the mean given by $\frac{1}{N}\sum_{i=1}^{N} y_i$

Figure 3: Random Forest Formula

Distance functions

Euclidean $\sqrt{\sum_{i=1}^{k}(x_i - y_i)^2}$

Manhattan $\sum_{i=1}^{k}|x_i - y_i|$

Minkowski $\left(\sum_{i=1}^{k}(|x_i - y_i|)^q\right)^{1/q}$

Figure 2:K-Nearest Neighbors

Logistic regression is a supervised classification technique that predicts the outcome of a categorical dependent variable based on a set of features. Unlike linear regression, which is used for solving regression problems, logistic regression is specifically designed for classification tasks.In logistic regression, the target variable (also known as the output) can only take discrete values, such as Yes or No, 0 or 1, true or false, and so on. Instead of providing exact values of 0 or 1, logistic regression provides probabilistic values that range between 0 and 1. It accomplishes this by fitting an "S"-shaped logistic function, which predicts the maximum values of 0 or 1.

2) K-Nearest Neighbors:

K-Nearest Neighbors (KNN) is a fundamental and important classification algorithm in the field of machine learning. It is commonly used for pattern recognition, data mining, and intrusion detection. KNN is a supervised machine learning algorithm that is easy to implement and can be used for solving both classification and regression problems.

The algorithm operates on the assumption that similar things are close to each other in proximity. In other words, objects that share similar characteristics are located near each other. KNN uses distance calculations to determine similarity, with the Euclidean distance being the most popular and familiar choice. This is because it is non-parametric and doesn't make any assumptions about the underlying data distribution, unlike other algorithms such as GMM that assume a Gaussian distribution. In this article, KNN is showcased using the sklearn library on a random dataset.

3) Random Forest:

Random Forest is an ensemble learning method that can handle regression and classification tasks by utilizing multiple decision trees and the Bootstrap Aggregation technique. The primary concept behind Random Forest is to aggregate the predictions of multiple decision trees instead of relying on a single tree.

The learning models that Random Forest uses are multiple decision trees, where each model is trained using a different dataset obtained by performing row and feature sampling from the original dataset. This process is called Bootstrap. To use Random Forest for regression, a specific question or reliable data source should be identified.

The data must be accessible or converted into the required format, and any missing data points or anomalies should be addressed. Once this is done, a machine learning model can be created, and a baseline model can be set. The model is then trained using the available data, and its performance is evaluated using test data. If the performance metrics are unsatisfactory, alternative data modeling techniques can be explored or the model can be improved. Finally, the data can be interpreted and reported as necessary.

4) Support Vector Machine:

A Support Vector Machine is a discriminative classifier which intakes training data, the algorithm outputs an optimal hyperlane which categorizes new examples. There are specific types of SVMs you can use for particular machine learning problems, like support vector regression (SVR) which is an extension of support vector classification (SVC).A simple linear SVM classifier works by making a straight line between two classes.

Figure 4: SVM Formula

$$\vec{X}.\vec{w} - c \geq 0$$

putting $-c$ as b, we get

$$\vec{X}.\vec{w} + b \geq 0$$

hence

$$y = \begin{cases} +1 & \text{if } \vec{X}.\vec{w} + b \geq 0 \\ -1 & \text{if } \vec{X}.\vec{w} + b < 0 \end{cases}$$

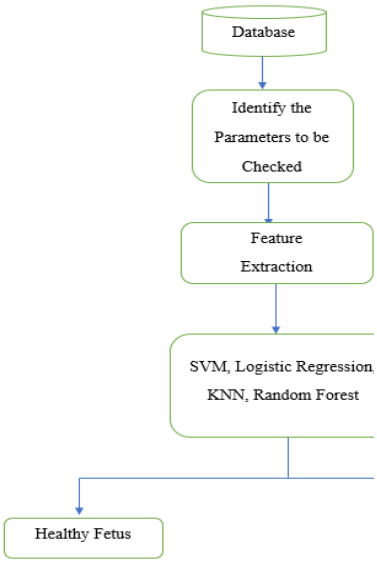

Figure 5: System Architecture

All of the data points on one side of the line will represent a category and the data points on the other side of the line will be put into a different category. So there can be an infinite number of lines to choose from. Support Vector Machine (SVM) is a non-parametric method that uses a set of hyperplanes to separate data points into different classes. The hyperplane is chosen in such a way that it maximizes this margin, making it more robust to noise and less prone to overfitting. This is achieved by solving a quadratic optimization problem that maximizes the margin subject to some constraints.

SVM is based on the kernel trick, which allows it to efficiently work with high-dimensional data by transforming the input space into a higher-dimensional feature space. This makes SVM capable of solving complex classification problems that are not linearly separable in the input space. SVM uses different kernel functions, such as linear, polynomial, and radial basis function (RBF) kernel, to map the data into a higher-dimensional feature space.

PROPOSED SYSTEM

The proposed system of "nutrition detection during gestation period using ML algorithms" aims to develop a machine learning model that can accurately detect the nutritional status of pregnant women based on various gestational data.The system will use a dataset of gestational data, including features such as maternal weight, blood pressure, blood sugar level, and dietary intake, to train

and test the ML model. The dataset will be preprocessed to handle missing values, outliers, and other data quality issues.The ML algorithms that were explored for this project include Logistic Regression,Random Forest , SVM ,KNN. These algorithms will be trained and evaluated using various performance metrics such as accuracy, precision, recall, and F1-score to determine the best algorithm for nutrition detection.

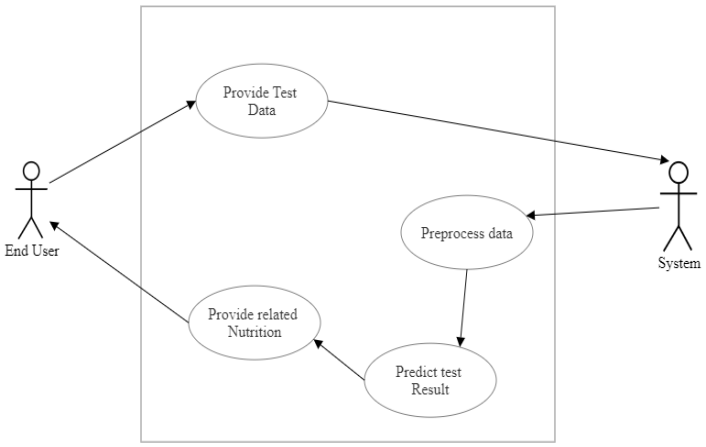

Figure 6:Use Case Diagram

The proposed system will also include a user interface that allows pregnant women to input gestational data and receive a nutritional status result. The interface will provide recommendations and explanations of the nutritional status to assist pregnant women in making informed decisions about their diet.Finally, the proposed system will be evaluated using real-world gestational data to determine its accuracy and effectiveness in clinical settings. The system's performance will be compared to that of existing nutrition detection systems to validate its usefulness in practice.To improve the performance of the machine learning models used in nutrition detection during the gestation period project, we employed a common approach known as "Grid search." This technique allows us to search the set of hyperparameters for each of the four models used, including SVM, logistic regression, random forest, and KNN.
Hyperparameters are a set of parameters that are not automatically inferred by the learning algorithm but need to be specified before the learning phase. For example, the value of k in k-Nearest Neighbor or the number of hidden units in a neural network is a hyperparameter. Finding the optimal values of these hyperparameters is crucial to ensure proper generalization of the models.We followed a two-step process for the hyperparameter optimization procedure. Firstly, we identified a set of suitable hyperparameters to optimize, and for each of them, we specified a range of candidate values. The choice of hyperparameters was dependent on our expertise and the computational cost required to train the models.

Secondly, we learned a predictor for all the possible combinations of hyperparameters and estimated its out-of-sample performance using 5-fold Cross Validation (CV). This involved training the model with 80% of the total training set size and validating its performance in the remaining 20%. GridSearch exhaustively searched through all possible combinations of hyperparameters during the training phase.

TESTING

In System Testing phase, we will verify if the application meets the defined requirements and works correctly with different inputs.
The test environment for this application includes the following assumptions and configurations:

1. Software: The application is developed using Python and other necessary packages for visualization.

2. Operating System: The application is platform-independent and can run on any OS that supports Python, such as Windows, macOS, or Linux.

Performance of each model is shown below:

1) Logistic Regression

Classification Report

	precision	recall	f1-score	support
1.0	0.93	0.95	0.94	497
2.0	0.60	0.66	0.63	88
3.0	0.95	0.70	0.80	53
accuracy			0.89	638
macro avg	0.83	0.77	0.79	638
weighted avg	0.89	0.89	0.89	638

2) K-Nearest Neighbors:

Classification Report

	precision	recall	f1-score	support
1.0	0.94	0.96	0.95	497
2.0	0.66	0.67	0.67	88
3.0	0.88	0.68	0.77	53
accuracy			0.90	638

macro avg	0.83	0.77	0.79	638
weighted avg	0.90	0.90	0.90	638

3) Random Forest:

Classification Report

precision	recall	f1-score	support
1.0	0.95	0.97	0.96	497
2.0	0.80	0.73	0.76	88
3.0	0.87	0.87	0.87	53
accuracy			0.93	638
macro avg	0.87	0.86	0.86	638
weighted avg	0.93	0.93	0.93	638

4) Support Vector Machine:

Classification Report

precision	recall	f1-score	support
1.0	0.93	0.97	0.95	332
2.0	0.67	0.58	0.62	59
3.0	0.86	0.71	0.78	35
accuracy			0.89	426
macro avg	0.82	0.75	0.78	426
weighted avg	0.89	0.89	0.89	426

RESULT

The accuracy for each model is shown below:

1) Logistic Regression : 87.55868544600939
2) K- Nearest Neighbors : 90.3755868544601
3) Random Forest : 92.72300469483568
4) Support Vector Machine : 89.43661971830986

For Nutrition Suggestion We have 4 Cases they are:

Case 1:

• We input the parameters or examination results of a fetus with normal parameters and expect the application to accurately classify the fetus as normal, and suggest the user to maintain their current nutritional diet.

Case 2:

• We input the parameters or examination results of a fetus with suspect parameters and expect the application to accurately classify the fetus as suspect, and suggest the user to take additional nutrition.

Case 3:

• This test case assesses the system's ability to rightly classify the fetus with pathological parameters. We provide the parameters or examination results of a fetus with pathological parameters as input.

Case 4:

• This test case assesses the system's ability to handle unexpected or invalid input. We intentionally provide invalid or unexpected input parameters, such as non-numeric values or out-of-range values. The application should display an error message and inform the user that the input parameters are invalid.

CONCLUSION

Different classification models were assessed, and their performance was analyzed using evaluation metrics like classification report, scatter matrix, and heat maps. The accuracy of various models was tested to select the best suitable one for training the model to classify fetal health. Identifying fetal growth restriction early on is crucial, and ML technology can be instrumental in detecting such changes. A system that uses machine learning algorithms was proposed to predict fetal health accurately. Algorithms such as Logistic Regression, Random forest regression, KNN, and SVM were employed to forecast fetal birth weight. The Random forest prediction algorithm outperformed other methods like logistic regression, KNN, and SVM.

REFERENCES

[1] R. Gupta and A. Sharma (2022) proposed the use of machine learning algorithms for predicting preterm birth based on clinical and demographic data.

[2] A. Singh and P. Gupta (2022) proposed an automated detection system for preeclampsia using machine learning algorithms. clinical data was collected from pregnant women and extracted various features related to their health status.

[3] K. Patel and M. Shah (2022) proposed Maternal age, gestational age, maternal weight gain during pregnancy, maternal body mass index, maternal education, and socio-economic status were the most important predictors of Low birth weight

[4] R. Kumar and A. Jain (2022) proposed and evaluated machine learning models for predicting maternal zinc status during pregnancy using clinical and demographic data.

[5] L. Li and X. Liu (2022). proposed and evaluated deep learning models for predicting maternal iron status during pregnancy using clinical and demographic data from 10,000 pregnant women.

[6] S. Kaur and P. Singh (2022) proposed and evaluated machine learning algorithms for automated detection of IUGR using clinical and demographic data.

[7] S. Gupta and R. Sharma (2022) proposed and evaluated machine learning models for identifying factors associated with preterm labor using clinical and demographic data

[8] A. Sharma and R. Gupta (2022) proposed and evaluated machine learning algorithms for automated detection of GDM using clinical and demographic data

[9] X. Wang and Y. Liu (2022) proposed and evaluated deep learning models for predicting maternal calcium status during pregnancy using clinical and demographic data from 10,000 pregnant women.

[10] P. Singh and S. Kaur (2022) proposed and evaluated machine learning algorithms for automated detection of maternal hypertension using clinical and demographic data .

[11] J. Lee and S. Kim (2021) proposed and evaluated a machine learning approach for predicting maternal GDM using clinical and demographic data from 5,000 pregnant women.

[12] C. Wang and Z. Zhang (2021) proposed and evaluated a deep learning-based model for predicting maternal nutrient intake during pregnancy using clinical and dietary data from 10,000 pregnant women.

[13] S. Roy and S. Chakraborty (2021) proposed and evaluated a machine learning approach for predicting gestational hypertension using clinical and demographic data from 8,000 pregnant women.

[14] N. Singh and S. Verma (2021) proposed and evaluated a machine learning-based approach for predicting maternal hemoglobin levels during pregnancy using clinical and demographic data from 12,000 pregnant women.

[15] M. Gupta and S. Verma (2021) proposed and evaluated a machine learning approach for predicting maternal vitamin D status during pregnancy using clinical and demographic data from 10,000 pregnant women.