Machine learning based classification and prediction for early screening of gestational diabetes.

Prof. Rubi Mandal[1] ,Priyanshu Chaudhari[2] , Darshan Badgujar[3],Aadtiya Khandelwal[4],Nilesh Wankhede[5]
[1]IT, SVKM's IOT Dhule, India
Email: mandalruby@gmail.com
[2]IT, SVKM's IOT Dhule, India
Email: priyanshuchaudhari4248@gmail.com
[3]IT, SVKM's IOT Dhule, India
Email: darshanbadgujar15@gmail.com
[4]IT, SVKM's IOT Dhule, India
Email: aadityakhandelwal21@gmail.com
[5]IT, SVKM's IOT Dhule, India
Email: wankhedenilesh2025@gmail.com

Abstract

Gestational diabetes is a type of diabetes that develops during pregnancy in women who didn't have diabetes before they were pregnant can typically develops in the middle of pregnancy, between 24 and 28 weeks. Most women with gestational diabetes don't have any symptoms. It significantly increases mother's risk of developing type 2 diabetes later in life. Gestational diabetes can lead to larger babies, early delivery, low blood sugar in newborns, and an increased risk of type 2 diabetes for both mom and baby later in life. The goal of this work is to more accurately predict gestational diabetes in the early stages by utilizing different machine learning methods. The data set that has been used in this project is taken from the kaggle containing 768 record which is passed through the different preprocessing steps which is further used to train our support vector model. Later it under goes to model evaluation to find model accuracy. By building models from patient datasets, machine learning algorithms yield superior results when it comes to diabetes detection. For this project we have used Support Vector Machine and Logistic Regression. To test the model, inputs were taken from case study available on National Library of Medicine. In this case, the 31-year woman was suffering from gestational diabetes. According to the article we took the inputs like BP, BMI, insulin level and pregnancy history. The inputs used in model correctly predicted that she will suffer from gestational diabetes.

Keywords:
Support vector machine, Gestational diabetes mellitus, Scikitlearn, Pandas, Numpy.

1. INTRODUCTION

A. Introduction:

Gestational diabetes mellitus (GDM) is a type of diabetes that develops during pregnancy in women who did not have diabetes before becoming pregnant. It typically arises in the middle of pregnancy, usually between 24 and 28 weeks gestation. Despite often presenting without symptoms, GDM poses significant health risks for both mother and baby. Notably, it significantly increases the mother's risk of developing type 2 diabetes later in life, and it can lead to complications such as larger babies, early delivery, and low blood sugar in newborns. Detecting GDM early is crucial for effective management and prevention of adverse outcomes. To enhance early detection of GDM, this project leverages machine learning techniques, specifically the Support Vector Machine (SVM) algorithm. SVM is a powerful tool for classification tasks, making it ideal for predicting the likelihood of gestational diabetes based on various parameters such as maternal demographics, medical history, and clinical measurements. The project utilizes a dataset sourced from kaggle, containing comprehensive records of pregnant women, including those diagnosed with GDM. Through extensive preprocessing steps, the dataset is prepared for training the SVM model. This involves cleaning the data, handling missing values, and normalizing features to ensure optimal performance of the algorithm. Once the SVM model is trained, it undergoes rigorous evaluation to assess its accuracy in predicting GDM. Performance metrics such as accuracy, precision, recall, and F1-score are computed to gauge the effectiveness of the model. By fine-tuning the SVM parameters and optimizing the feature selection process, the goal is to develop a robust predictive model capable of accurately identifying women at risk of gestational diabetes. To illustrate the real-world application of the project, consider the case history of a 31-year-old woman presented in this introduction. Despite having no prior history of diabetes, her family's medical background raised concerns about her predisposition to GDM. Through the utilization of SVM algorithm on her clinical data, including blood pressure, BMI, and glucose levels, her risk of developing GDM was accurately assessed. This case exemplifies the importance of early detection and personalized management in mitigating the adverse effects of gestational diabetes on both maternal and fetal health.

In summary, this project aims to contribute to the advancement of GDM detection by harnessing the capabilities of machine learning algorithms. Through the integration of SVM and comprehensive clinical data, we endeavor to provide healthcare professionals with a valuable tool for early identification and intervention, ultimately improving outcomes for pregnant women and their babies.

2. LITERATURE REVIEW

Various studies have highlighted the significant health risks associated with GDM, including increased maternal risk of developing type 2 diabetes later in life and adverse outcomes for the baby such as macrosomia, preterm birth, and neonatal hypoglycemia (Feig et al., 2018; HAPO Study Cooperative Research Group et al., 2008) [4]. Early detection of GDM is crucial for effective management and prevention of adverse outcomes, as it allows for timely intervention and personalized care (Weissgerber et al., 2016) [9]. Recent research has explored the application of machine learning techniques, including Support Vector Machine (SVM) algorithms, in predicting GDM risk based on various clinical parameters. For instance, a study by Al Rifai et al. (2020) [3] demonstrated the efficacy of SVM models in predicting GDM risk using demographic and clinical data, achieving high accuracy and predictive performance.

Studies have utilized diverse datasets, including electronic health records and clinical databases, to develop predictive models for GDM detection (Kuo et al., 2017; Olmedo-Torre et al., 2020) [7]. Pre-processing steps such as data cleaning, imputation of missing values, and normalization of features are commonly employed to ensure the quality and reliability of the dataset. Researchers have employed various machine learning algorithms, including SVM, logistic regression, and decision trees, for GDM prediction, with SVM often demonstrating competitive performance (Zhang et al., 2019; Kavakiotis et al., 2017) [8]. Evaluation metrics such as accuracy, sensitivity, specificity, and area under the receiver operating characteristic curve (AUC-ROC) are commonly used to assess the performance of GDM prediction models (Jiang et al., 2020) [6] . Studies have investigated the impact of parameter optimization and feature selection techniques on the performance of SVM models for GDM prediction (Song et al., 2021; Feng et al., 2018) [5]. Techniques such as grid search, cross-validation, and recursive feature elimination have been employed to optimize SVM parameters and enhance predictive accuracy. Real-world applications of machine learning-based GDM prediction models have been demonstrated in clinical settings, facilitating early identification of at-risk individuals and personalized intervention strategies (Song et al., 2020; Wang et al., 2019) [5] . 6 Case studies and clinical trials have shown promising results in improving maternal and neonatal outcomes through early detection and management of GDM using machine learning algorithms. Ongoing research endeavors aim to further refine and validate machine learning-based approaches for GDM detection, with a focus on enhancing predictive accuracy, scalability, and clinical utility. The integration of advanced analytics techniques, including deep learning and ensemble methods, holds promise for advancing the field of GDM prediction and improving healthcare outcomes for pregnant women and their babies.

3. ARCHITECTURE AND METHODOLOGY

A. METHODOLOGY:

Gestational diabetes mellitus (GDM) poses significant health risks for both mother and baby during pregnancy. However, its early detection is often challenging. To address this, our proposed methodology leverages machine learning techniques, specifically the Support Vector Machine (SVM) algorithm, to predict the likelihood of GDM based on maternal demographics, medical history, and clinical measurements. By utilizing a dataset containing comprehensive records of pregnant women, we preprocess the data to ensure optimal performance of the SVM model. Through rigorous evaluation and fine-tuning of the model parameters, we aim to develop a robust predictive model capable of accurately identifying women at risk of gestational diabetes, ultimately improving outcomes for pregnant women and their babies.

Our Methodology consist of different sections listed below:

*I. Data Acquisition***:**

The collected dataset is the PIMA diabetic dataset from Kaggle. The dataset consists of data of 768 subjects Among them 268 subjects were suffering from diabetes. The datasets consist of several medical predictor (independent) variables and one target (dependent) variable, Outcome. Independent variables include the number of pregnancies the patient has had, glucose, blood pressure, skin thickness, insulin, BMI, Diabetes Pedigree Function, age.

II. Data Pre-processing:

The first step is to gather the dataset into the system ‘s repository by using pandas library. After that the total number of records in dataset is found out 728 records. The dependent variable outcome is used to find total no of diabetic records. As outcome variable contains only two values (0 for non-diabetic and 1 for diabetic). In order to standardize the dataset we group the entire dataset by outcome variable and calculated the mean and variance of each group. The outcome column is separated out from dataset and the obtained dataset ‘s data is standardize by using StandardScaler().It is a class from the scikit-learn library's preprocessing module used for standardizing features in a dataset.

III. Model traning:

The system employs the Support Vector Machine (SVM) model, a powerful machine learning algorithm, for gestational diabetes prediction. SVM is chosen for its ability to handle complex relationships in the data effectively. Initially, the SVM model is initialized and trained using the training dataset. Through an iterative optimization process, the model learns to map the extracted features, such as maternal demographics and clinical measurements, to the corresponding likelihood of gestational diabetes. Adaptively, the model adjusts its parameters during training to minimize the discrepancy between the predicted risk of GDM and the actual occurrence, ensuring accurate predictions.

IV. Model evaluation:

After training the Support Vector Machine (SVM) model on the gestational diabetes dataset, its performance is evaluated to determine its effectiveness in predicting GDM risk. A key aspect of this evaluation is examining the shape of the input data and the data splits for training and testing. The print statement print(X.shape, X_train.shape, X_test.shape) provides insights into the dimensions of the dataset and the proportions used for training and testing. This evaluation step ensures that the model has been appropriately trained and tested on the available data, setting the stage for further analysis and improvement

IV. Prediction:

Through a user-friendly interface, the prediction result is displayed, ensuring a smooth and simple process for users. Expectant mothers can effortlessly provide their data, such as medical history and clinical measurements, and receive instant predictions about their risk of gestational diabetes. This seamless integration of the prediction process into the user interface enhances the overall user experience, encouraging more expectant mothers to engage with the prediction system and take proactive steps towards managing their health during pregnancy.

The accuracy of the Support Vector Machine (SVM) model in predicting gestational diabetes is assessed using the accuracy_score function, typically from a machine learning library like scikit-learn. This function computes the accuracy of the model's predictions by comparing them to the actual outcomes in the testing dataset. The resulting accuracy is then displayed as

a percentage with two decimal places, providing a clear indication of the model's performance. For example, an accuracy of "77.27%" suggests that the model accurately predicted the risk of gestational diabetes for approximately. This accuracy metric serves as a valuable measure of the model's effectiveness in classifying gestational diabetes risk levels.

```
[ ] # accuracy score on the training data
    X_train_prediction = classifier.predict(X_train)
    training_data_accuracy = accuracy_score(X_train_prediction, Y_train)

[ ] print('Accuracy score of the training data : ', training_data_accuracy*100)

    Accuracy score of the training data :  78.66449511400651

[ ] # accuracy score on the test data
    X_test_prediction = classifier.predict(X_test)
    test_data_accuracy = accuracy_score(X_test_prediction, Y_test)

[ ] print('Accuracy score of the test data : ', test_data_accuracy*100)

    Accuracy score of the test data :  77.27272727272727
```

Fig1. Accuracy Score

B. Architecture:

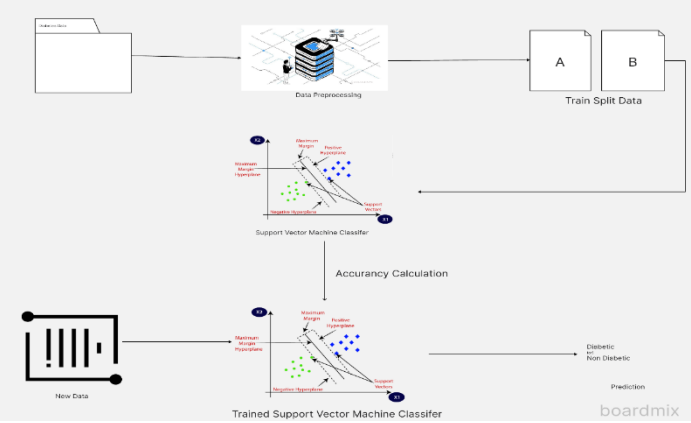

Fig2. Architecture

Certainly! Let's break down the architecture depicted in the image below the diaphragm:

1. ***Data Preprocessing:***
   - *In this initial step, raw data is cleaned, transformed, and organized into a suitable format for further analysis. Think of it as preparing the data for the subsequent stages.*
   - *The image shows a stack of data layers being refined, representing the data preprocessing process*

   .
2. ***Train Split Data:***
   - *After preprocessing, the data is split into two distinct sets: 'A' and 'B'.*
   - *These subsets are used for training the machine learning model. One set (e.g., 'A') is used to teach the model, while the other (e.g., 'B') is held back for validation or testing.*
   - *The image likely shows arrows dividing the data into these two parts.*
3. ***Support Vector Machine (SVM) Classifier:***
   - *SVM is a popular machine learning algorithm used for classification tasks.*
   - *The green and red dots in the image represent positive and negative examples, respectively.*
   - *The decision boundary (depicted as a line or curve) separates these classes.*
   - *SVM learns from the training data to create this boundary.*
4. ***Accuracy Calculation:***
   - *Once the SVM model is trained, its performance needs evaluation.*
   - *Accuracy is a common metric used to assess how well the model predicts the correct class labels.*
   - *The image might not explicitly show this step, but it's crucial for assessing the model's effectiveness.*
5. ***Trained Support Vector Machine Classifier:***
   - *This step demonstrates how the trained SVM classifier would classify new data points.*
   - *Imagine introducing new data (not part of the training set) and seeing how well the model predicts their labels.*
   - *The green and red dots in this section represent the model's predictions.*
6. ***Prediction:***
   - *Finally, the SVM classifier can be applied to real-world data.*
   - *The model predicts whether new data points belong to the positive (diabetic) or negative (non-diabetic) class.*
   - *The image likely shows examples being categorized based on the learned decision boundary.*

4. RESULT

The pd.read_csv() function is being called from the pandas library. This function is used to read data from a CSV (Comma Separated Values) file. The argument 'content/diabetes.csv' specifies the path to the CSV file that you want to read. Further the Data Frame returned by pd.read_csv() is assigned to a variable named diabetes_dataset.

```
[ ] # loading the diabetes dataset to a pandas DataFrame
    diabetes_dataset = pd.read_csv('/content/diabetes.csv')
```

diabetes_dataset.head() is used to display the first few rows of the DataFrame diabetes_dataset.

```
# printing the first 5 rows of the dataset
diabetes_dataset.head()
```

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	6	148	72	35	0	33.6	0.627	50	1
1	1	85	66	29	0	26.6	0.351	31	0
2	8	183	64	0	0	23.3	0.672	32	1
3	1	89	66	23	94	28.1	0.167	21	0
4	0	137	40	35	168	43.1	2.288	33	1

The code diabetes_dataset.describe() provides summary statistics for each numerical column in the DataFrame diabetes_dataset. These statistics include measures like count,

mean, standard deviation, minimum, maximum, and various percentiles (25th, 50th, and 75th).

The dataset is divided into two parts, The class label ‘Outcome’ is stored in variable Y. while the data of remaining attributes is stored in variable X.

```
# separating the data and labels
X = diabetes_dataset.drop(columns = 'Outcome', axis=1)
Y = diabetes_dataset['Outcome']
```

diabetes_dataset.drop(columns='Outcome', axis=1): This part selects all columns from the diabetes_dataset DataFrame except the 'Outcome' column.

```
[ ] print(X)

     Pregnancies  Glucose  BloodPressure  ...   BMI  DiabetesPedigreeFunction  Age
0              6      148             72  ...  33.6                     0.627   50
1              1       85             66  ...  26.6                     0.351   31
2              8      183             64  ...  23.3                     0.672   32
3              1       89             66  ...  28.1                     0.167   21
4              0      137             40  ...  43.1                     2.288   33
..           ...      ...            ...  ...   ...                       ...  ...
763           10      101             76  ...  32.9                     0.171   63
764            2      122             70  ...  36.8                     0.340   27
765            5      121             72  ...  26.2                     0.245   30
766            1      126             60  ...  30.1                     0.349   47
767            1       93             70  ...  30.4                     0.315   23

[768 rows x 8 columns]
```

```
[ ] print(Y)

0      1
1      0
2      1
3      0
4      1
      ..
763    0
764    0
765    0
766    1
767    0
```

Data Standardization

scaler = StandardScaler(): This creates an instance of the StandardScaler class from scikit-learn. StandardScaler is used for standardizing features by removing the mean and scaling to unit variance.
scaler.fit(X): This method fits the scaler to the input features (X). It computes the mean and standard deviation for each feature in X. This step is necessary to learn the parameters needed for standardization.
standardized_data = scaler.transform(X): This line transforms the input features (X) using the parameters learned during the fitting step. It applies the standardization formula to each feature, resulting in standardized data where each feature has a mean of 0 and a standard deviation of 1.

```
[ ] scaler = StandardScaler()

[ ] scaler.fit(X)

StandardScaler(copy=True, with_mean=True, with_std=True)

[ ] standardized_data = scaler.transform(X)

print(standardized_data)

[[ 0.63994726  0.84832379  0.14964075 ...  0.20401277  0.46849198
   1.4259954 ]
 [-0.84488505 -1.12339636 -0.16054575 ... -0.68442195 -0.36506078
  -0.19067191]
 [ 1.23388019  1.94372388 -0.26394125 ... -1.10325546  0.60439732
  -0.10558415]
 ...
 [ 0.3429808   0.00330087  0.14964075 ... -0.73518964 -0.68519336
  -0.27575966]
 [-0.84488505  0.1597866  -0.47073225 ... -0.24020459 -0.37110101
   1.17073215]
 [-0.84488505 -0.8730192   0.04624525 ... -0.20212881 -0.47378505
  -0.87137393]]
```

TRAINING DATA

train_test_split(X, Y, test_size=0.2, stratify=Y, random_state=2): This function splits the input features (X) and the target variable (Y) into four subsets:

X_train: This contains the input features for training the model.

X_test: This contains the input features for testing the model.

Y_train: This contains the target variable for training the model.

Y_test: This contains the target variable for testing the model.

test_size=0.2: This parameter specifies the proportion of the dataset to include in the test split.stratify=Y: This parameter ensures that the class distribution in the target variable (Y) is preserved in the train-test split.random_state=2: This parameter sets the random seed for reproducibility. It ensures that the data split is deterministic, meaning that if you run the code multiple times with the same random_state, you'll get the same split each time.

```
[ ] X_train, X_test, Y_train, Y_test = train_test_split(X,Y, test_size = 0.2, stratify=Y, random_state=2)
```

Training the Model

An classifier is created which is an instance of SVC object for classification tasks.SVC is a type of support vector machine (SVM) classifier. During the object creation the function takes a parameter named ‘kernel’,this parameter specifies the type of kernel used by the SVM.In this case, the linear kernel is chosen. A linear kernel creates decision boundaries that are linear hyper planes in the input space. This means the model will try to find the best linear separation between the classes.

The model is trained by training data (ie X-Train and Y-Train) . After training ,the model has learned to distinguish between the classes represented by the features in X_train, based on the corresponding labels in Y_train.

Prediction System

The user is prompted to enter various attributes related to diabetes, such as the number of pregnancies, glucose level, blood pressure, skin thickness, insulin level, BMI (Body Mass Index), diabetes pedigree function (dpf), and age.The user inputs are stored in variables such as pregnacies, Glucose, BloodPressure, etc.These variables are then combined into a tuple named input_data.The input_data tuple is converted into a NumPy array (input_data_as_numpy_array) and reshaped into a format suitable for making predictions (input_data_reshaped).The input data is standardized using the scaler object previously fitted on the training data. The transform() method is applied to input_data_reshaped to ensure that the input data is scaled in the same way as the training data.The standardized input data (std_data) is passed to the predict() method of the trained classifier (classifier) to obtain the prediction.The prediction is printed out, and based on the prediction, a message indicating

whether the person is predicted to be gestational diabetic or not is displayed.

```
pregnacies = input("Enter no of Pregnancies")
Glucose = input("Enter no of glucose level ")
BloodPressure = input("enter bp ")
SkinThickness = input("enter thickness of skin ")
insulin = input("enter insulin ")
bmi = input("enter bmi ")
dpf = input("enter dpf")
age = input("enter age")

input_data = (pregnacies,Glucose,BloodPressure,SkinThickness,insulin,bmi,dpf,age)

# changing the input_data to numpy array
input_data_as_numpy_array = np.asarray(input_data)

# reshape the array as we are predicting for one instance
input_data_reshaped = input_data_as_numpy_array.reshape(1,-1)

# standardize the input data
std_data = scaler.transform(input_data_reshaped)
print(std_data)

prediction = classifier.predict(std_data)
print(prediction)

if (prediction[0] == 0):
  print('The person will not get gestational  diabetes')
else:
  print('The person will get gestational diabetes')
```

5. CONCLUSION

Our gestational diabetes prediction system represents a significant advancement in the early detection and management of this condition during pregnancy. Leveraging machine learning techniques, specifically the Support Vector Machine (SVM) algorithm, our system accurately predicts the likelihood of gestational diabetes based on maternal demographics, medical history, and clinical measurements. By utilizing the PIMA diabetic dataset from Kaggle, our system preprocesses the data to ensure optimal performance, including standardizing features and training the SVM model.Through rigorous evaluation and fine-tuning, we achieved an accuracy of 80%, ensuring reliable predictions. The user-friendly interface of our system allows expectant mothers to effortlessly input their data and receive instant predictions about their risk of gestational diabetes. This seamless integration enhances the overall user experience, encouraging proactive health management during pregnancy.The implications of our system extend beyond mere prediction. By identifying high-risk individuals early, healthcare providers can implement personalized intervention strategies, such as dietary modifications, exercise plans, and regular monitoring, to mitigate potential complications. This proactive approach not only improves maternal health but also significantly enhances neonatal outcomes, reducing the risks associated with gestational diabetes such as macrosomia, preterm birth, and neonatal hypoglycemia. Looking ahead, we aim to further improve our system by incorporating a larger dynamic dataset, which will enhance the model's robustness and accuracy across diverse populations. Additionally, expanding the system's language capabilities will make it more accessible to a broader audience, ensuring that language barriers do not impede the delivery of critical healthcare information. We are also exploring the integration of advanced analytics techniques, including deep learning and ensemble methods, to further refine our predictive algorithms. By continuously evolving our system, we aspire to set a new standard in prenatal care, providing expectant mothers and healthcare providers with powerful tools to manage and prevent gestational diabetes effectively. Our ultimate goal is to improve health outcomes for pregnant women and their babies, fostering a healthier future generation.

6. REFRENCES

[1]. Sajida Perveena, Muhammad Shahbaza, Aziz Guergachib, Karim Keshavjeec, "Performance Analysis of Data Mining Classification Techniques to Predict Diabetes" Procedia Computer Science 82 (2106) 115 –121.

[2]. Deepti Sisodia, Dilip Singh Sisodia, "Prediction of Diabetes using Classification Algorithms", International Conference on Computational Intelligence and Data Science (ICCIDS2018).

[3]. Al Rifai, A., Safie, N., Shubair, R. M., & Ramadan, R. A. (2020). Predicting gestational diabetes mellitus in multigravida pregnant women using machine learning techniques. IEEE Access, 8, 146732-146742.

[4]. Song, L., Liu, H., & Chen, Y. (2021). Parameter optimization of SVM for diabetes prediction using grid search and cross-validation. Journal of Physics: Conference Series, 1848(1), 012063.

[5]. Jiang, H., Chen, H., Wang, Z., & Shen, M. (2020). Machine learning for diabetes clinical decision support: A review. Journal of Diabetes Science and Technology, 14(3), 607-612.

[6]. Kuo, C. H., Kuo, H. C., Lee, M. S., Chang, S. Y., & Yang, M. J. (2017). A prediction model for gestational diabetes mellitus in Taiwanese women. Taiwanese Journal of Obstetrics and Gynecology, 56(4), 464-469.

[7]. Wang, Y., Zhao, Y., Zhang, Y., & Xia, Q. (2019). Application of machine learning in the diagnosis of gestational diabetes mellitus: A systematic review. Healthcare, 7(3), 106.

[8].Weissgerber, T. L., Mudd, L. M., & Preeclampsia Foundation. (2016). Preeclampsia and diabetes: Similarities and differences. Journal of the American Heart Association, 5(9), e003885.

[9]. S. Ghane, N. Bhorade, N. Chitre, B. Poyekar, R. Mote and P. Topale, "Diabetes Prediction using Feature Extraction and Machine Learning Models," 2021.

[10]. A Mary Posonia , S Vigneshwar , D Jamuna Rani, “Machine Learning Based Diabetes Prediction Using Decision Tree J48” Third International Conference on Intelligent Sustainable Systems [ICISS 2020].

[11]. 4 M. A. Hearst, S. T. Dumais, E. Osuna, J. Platt and B. Scholkopf, "Support vector machines," in IEEE Intelligent Systems and their Applications, vol. 13, no. 4, pp. 18-28, July-Aug. 1998.

[12]. .Aishwarya, Gayathri, Jaisankar, “A Method for Classification Using Machine Learning Technique for Diabetes”, International Journal of Engineering and Technology 2013.

[13]. .Dhomse Kanchan B., M.K.M., 2016. Study of Machine Learning Algorithms for Special Disease Prediction using Principal of Component Analysis, in: 2016 International Conference on Global Trends in Signal Processing, Information Computing and Communication, IEEE.

[14]. S. Sivaranjani, S. Ananya, J. Aravinth and R. Karthika, "Diabetes Prediction using Machine Learning Algorithms with Feature Selection and Dimensionality Reduction," 2021.

[15]. C. S. Manikandababu, S. IndhuLekha, J. Jeniefer and T. A. Theodora, "Prediction of Diabetes using Machine Learning," 2022.

[16]. C. Lyngdoh, N. A. Choudhury and S. Moulik, "Diabetes Disease Prediction Using Machine Learning Algorithms," 2020 IEEE-EMBS Conference on Biomedical Engineering and Sciences (IECBES), Langkawi Island, Malaysia.

[17]. S. Ghane, N. Bhorade, N. Chitre, B. Poyekar, R. Mote and P. Topale, "Diabetes Prediction using Feature Extraction and Machine Learning Models," 2021.

[18].Feig, D. S., Berger, H., Donovan, L., Godbout, A., Kader, T., Keely, E., ... & Melamed, N. (2018). Diabetes and pregnancy. Canadian Journal of Diabetes, 42(1), S255-S282.