Gestational Diabetes Finder Using Machine Learning

A PROJECT REPORT

Submitted by

AVINASH S [REGISTER NO: 211414184030]

in partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

PANIMALAR ENGINEERING COLLEGE, CHENNAI-600123.
(An Autonomous Institution, Affiliated to Anna University, Chennai)

MAY 2022

PANIMALAR ENGINEERING COLLEGE

(An Autonomous Institution, Affiliated to Anna University, Chennai)

BONAFIDE CERTIFICATE

Certified that this project report **“Gestational Diabetes Finder Using Machine Learning** “is the bonafide work of “**AVINASH.S (211418104030)”**who carried out the project work under **Mr.A.N.SASIKUMAR, M.E** supervision.

SIGNATURE	**SIGNATURE**
Dr.S.MURUGAVALLI,M.E.,Ph.D.,	**Mr.A.N.SASIKUMAR,M.E,**
HEAD OF THE DEPARTMENT	**SUPERVISOR**
PROFESSOR	**ASSISTANT PROFESSOR**
DEPARTMENT OF CSE,	DEPARTMENT OF CSE,
PANIMALAR ENGINEERING COLLEGE,	PANIMALAR ENGINEERING COLLEGE
NASARATHPETTAI,	NASARATHPETTAI,
POONAMALLEE,	POONAMALLEE,
CHENNAI-600 123.	CHENNAI-600 123.

Certified that the above mentioned candidate(s) were examined in End Semester Project Viva-Voce held on..........................

INTERNAL EXAMINER	**EXTERNAL EXAMINER**

DECLARATION BY THE STUDENT

I **AVINASH.S (211418104030)** hereby declare that this project report titled “Gestational Diabetes Finder Using Machine Learning.” , under the guidance of **Mr.A.N.SASIKUMAR, M.E** is the original work done by us and we have not plagiarised or submitted to any other degree in any university by us.

ACKNOWLEDGEMENT

We would like to express our deep gratitude to our respected Secretary and Correspondent **Dr.P.CHINNADURAI, M.A., Ph.D.** for his kind words and enthusiastic motivation, which inspired us a lot in completing this project.

We express our sincere thanks to our Directors **Tmt.C.VIJAYARAJESWARI**, **Dr.C.SAKTHI KUMAR,M.E.,Ph.D** and **Dr.SARANYASREE SAKTHI KUMAR B.E.,M.B.A., Ph.D.,** for providing us with the necessary facilities to undertake this project.

We also express our gratitude to our Principal **Dr.K.Mani, M.E., Ph.D.** who facilitated us in completing the project.

We thank the Head of the CSE Department, **Dr.S.MURUGAVALLI .M.E.,Ph.D. ,** for the support extended throughout the project.

We would like to thank my **Mr.A.N.SASIKUMAR M.E** and all the faculty members of the Department of CSE for their advice and encouragement for the successful completion of the project.

AVINASH S

ABSTRACT

Gestational diabetes is a type of diabetes that occurs only during pregnancy. Gestational diabetes can cause health problems in both mother and baby. Managing the diabetes can help protect the mother and the baby. Gestational diabetes often has no symptoms, or they may be mild, such as being thirstier than normal or having to urinate more often. Gestational diabetes is sometimes related to the hormonal changes of pregnancy that make the body less able to use insulin. Genes and extra weight may also play a role. The doctor will test you for gestational diabetes between 24 and 28 weeks of pregnancy. Tests include the glucose challenge test and the oral glucose tolerance test (OGTT). If the results of the glucose challenge test show high blood glucose, they will return for an OGTT test to confirm the diagnosis of gestational diabetes. Machine learning plays a significant role in healthcare industries. Using machine learning algorithms one can study huge patient data is and find hidden information ,hidden patterns to discover knowledge from the data and read further processed and predicting it using suitable algorithm, classification models plays an crucial role in calculating the accuracy levels and comparing the models and then used for prediction. By developing a system which shows perfect accuracy level and could save the life of mother and fetus from disease. Predicting the early stage would be easier to treat the patient with medications and control with precautionaries, further controlling complications.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

GDM	GESTATIONAL DIABETES MELLITUS
OGTT	ORAL GLUCOSE TOLERANCE TEST
RFM	RANDOM FOREST MODEL
ML	MACHINE LEARNING
CDSS	CLINICAL DECISION SUPPORT SYSTEM
PROBAST	PREDICTION MODEL RISK OF BIAS ASSESSMENT TOOL
DM	DIABETES MELLITUS
AROC	AREA UNDER THE RECEIVER OPERATING CHARACTERISTICS CURVE
LR	LOGISTIC REGRESSION
BMI	BODY MASS INDEX
PCA	PRINCIPAL COMPONENT ANALYSIS
AUC	AREA UNDER THE CURVE

CHAPTER 1
INTRODUCTION

1.1 Overview

The traditional behaviour model consists of creating a statistical or analytical model of human behaviour and then fitting a distribution to the model to validate it. The machine learning approach differs significantly from the traditional model. By studying plethora amount of data, a machine learning system learns to predict the outcomes. Healthcare sectors have large volume databases. Such databases may contain structured, semi-structured, un-structured data. Machine learning is the process which analyses huge datasets and predicts the outcome. By using variety of machine learning algorithms, that uses current and past data to predict the future events. By applying predictive analysis on patient data, a significant decision can be made and prediction too. Predictive analysis can be done using machine learning and regression techniques. Machine learning is considered to be a dire need of today's situation in order to eliminate human efforts by supporting automation with minimum flaws. Existing methods for diabetes is time consuming. This project focuses on building predictive models using machine learning algorithms for diabetic prediction.

Gestational diabetes means high blood sugar levels during pregnancy. In many women, it is a temporary condition that goes away after birth. However, if the women are at risk of developing gestational diabetes, they should work with the doctor to manage their blood sugar levels throughout pregnancy. They can look for an easy high blood sugar during pregnancy meal plan and exercise regularly to maintain healthy blood sugar levels. Maintaining blood sugar levels during pregnancy keeps your pregnancy and baby away from several health-related complications.

Giving birth is one of the most beautiful experiences in the life of a woman. However, this experience can be bitter if a woman has diabetes or develops diabetes during pregnancy. A lot of women with diabetes may have to face severe health complications during and post-delivery. Diabetes during pregnancy affects the mother and also increases the risk for the baby in the womb. Therefore, it is significant to keep the blood sugar levels under control throughout pregnancy to have a healthy pregnancy and healthy baby. By developing a system which predict gestational diabetes and could save the life of mother and fetus from disease. Predicting gestational diabetes in the early stage would be easier to treat the patient with medications and control with precautionaries, further controlling complications.

Managing gestational diabetes includes following a healthy eating plan and being physically active. If you're eating plan and physical activity aren't enough to keep your blood glucose in your target range, you may need insulin.

You can lower your chance of getting gestational diabetes by losing extra weight before you get pregnant if you are overweight. Being physically active before and during pregnancy also may help prevent gestational diabetes.

Types of Diabetes

- **Type 1 diabetes** means that the immune system is compromised and the cells fail to produce insulin in sufficient amount. There are no eloquent studies that prove the cause of type 1 diabetes and there are currently no known methods of prevention
- **Type 2 diabetes** means that the cells produce a low quantity of insulin or the body can't use the insulin correctly. This is the most common type of diabetes, thus affecting 90% of persons diagnosed with diabetes. It is caused by both genetic factor and the manner of living.

Gestational diabetes appears in pregnant women who suddenly develop high blood sugar. In two thirds of the cases, it will reappear during subsequent pregnancies. There is a great chance that type 1 or type 2 diabetes will occur after a pregnancy affected by gestational diabetes.

Symptoms of Diabetes

- Frequent urination
- Increased thirst
- Tired and sleepiness
- Weight loss
- Blurred vision
- Mood swings
- Confusion and difficulty in concentration
- Frequent infection

Causes of Diabetes

Genetic factors are the main cause of diabetes. It is caused by at least two mutant genes in the chromosome 6, the chromosome that affects the response of the body to various antigens.

Viral infection may also influence the occurrence of type 1 and type 2 diabetes. Studies have shown that infection with viruses such as rubella ,hepatitis B virus increases the risk of developing diabetes

Risk factors for GDM

- Several risk factors are associated with the development of GDM.
- The most common risk factors are; obesity, older maternal age, past history of GDM, strong family history of diabetes, member of an ethnic group with a high prevalence of T2DM, polycystic ovary syndrome, and persistent glucosuria
- A history of delivering big baby (birth weight ≥4000 g), history of recurrent abortions, and history of unexplained stillbirths, and history of essential hypertension, or pregnancy-related hypertension are other risk factors for GDM.

Risks of GDM

- Women with GDM have an increased incidence of hypertensive disorders during pregnancy, including gestational hypertension, pre-eclampsia, and eclampsia.
- There is an increase risk of polyhydramnios that may increase the risk of preterm labor.
- Excessive fetal growth remains an important perinatal concern in GDM.
- Consequences of excessive fetal growth include birth trauma, maternal morbidity from cesarean deliveries, shoulder dystocia, and neonatal hypoglycemia.
- Other neonatal morbidities that potentially occur more frequently in infants of women with GDM include hyperbilirubinemia, hypocalcemia, erythema, and respiratory distress syndrome.
- Long-term complications of GDM include diabetes and cardiovascular disease in the mothers, and obesity and diabetes in the offspring

Prevention Measures for Diabetes

Managing gestational diabetes includes following a healthy eating plan and being physically active. If you're eating plan and physical activity aren't enough to keep your blood glucose in your target range, you may need insulin.

You can lower your chance of getting gestational diabetes by losing extra weight before you get pregnant if you are overweight. Being physically active before and during pregnancy also may help prevent gestational diabetes.

1.2 AIM OF THE PROJECT

The purpose of the project is to develop a system which can predict the GDM (Gestational diabetes mellitus) which occurs during pregnancy using suitable machine learning algorithms which have perfect accuracy level.

By developing a system which could save the time and life of mother and fetus from disease. Because ,to identify gestational diabetes is time consuming. Predicting the early stage would be easier to treat the patient with medications and control with precautionaries, further controlling complications.

1.3 Project Domain

1.3.1 Machine learning

In the world, we are surrounded by humans who can learn everything from their experiences with their learning capability, and that we have computers or machines which work on our commands. But can a machine also learn from experiences or past data similar to human does? So here comes the role of Machine Learning.

Machine Learning is a subset of artificial intelligence that is mostly concerned with the development of algorithms which permit a computer to find out from the data and past experiences on their own. The term machine learning was first introduced by Arthur Samuel in 1959.

Machine learning allows a machine to automatically learn from data, improve performance from experiences, and predict things without being explicitly programmed.

With the support of sample historical data, which is known as training data, machine learning algorithms build a mathematical model that helps in making predictions or decisions without being explicitly programmed. Machine learning brings computer science and statistics together for creating predictive models. Machine learning constructs or uses the algorithms that learn from historical data. The more we will provide the data, the higher will be the performance.

A machine has the ability to study if it can improve its performance by gaining more data.

How does Machine Learning work

A Machine Learning system studies from historical data, builds the prediction models, and whenever it accepts new data, predicts the output for it. The accuracy of predicted output depends upon the volume of data, as the vast amount of data helps to create a better model which predicts the output more accurately.

Suppose we’ve a complex problem, where we need to perform some predictions, so rather than writing a code for it, we just got to feed the data to generic algorithms, and with the help of these algorithms, machine builds the logic as per the data and predict the output. Machine learning has changed our method of thinking about the problem.

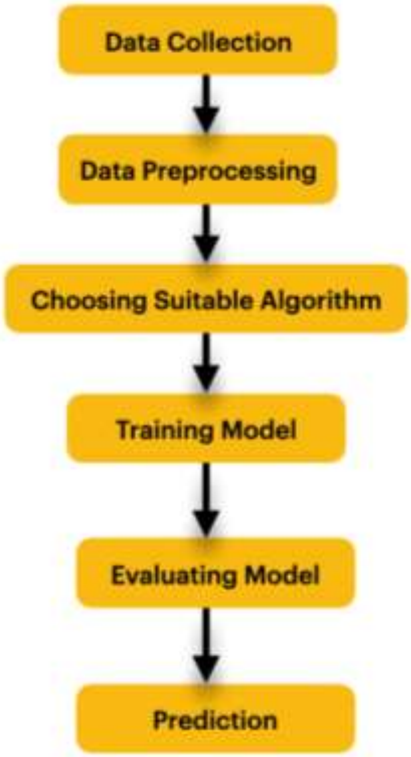


Fig 1.1: Machine Learning Workflow

Features of Machine Learning:

- Machine learning uses data to detect many patterns in a certain dataset.
- It is a data-driven technology.
- It can study from past data and recover automatically.
- Machine learning is much similar to data mining as it also deals with the vast amount of the data.

Need for Machine Learning

The need for machine learning is growing day by day. The reason behind the need for machine learning is that it is capable of doing tasks that are too complex for a individual to implement directly. As a human, we have some limits as we cannot access the enormous amount of data manually, so for this, we need some computer systems and here comes the machine learning to make things easy for us.

We can train machine learning algorithms by providing them the large amount of data and allow them explore the data, construct the models, and predict the specified output automatically. The performance of the machine learning algorithm depends on the amount of data, and it are often determined by the price function. With the help of machine learning, we will save both time and money.

The importance of machine learning are often easily understood by its uses cases, now, machine learning is used in **self-driving cars**, **cyber fraud detection**, **face recognition**, and **friend suggestion by Facebook**, etc. Various top companies like Netflix and Amazon have build machine learning models that are using a vast amount of data to analyze the user interest and recommend product accordingly.

Following are some key points which show the importance of Machine Learning:

- Quick increment in the production of data.
- Resolving complex problems, which are difficult for a human.

- Decision making in several sector including finance.
- Finding hidden patterns and extracting useful information from data.

Classification of Machine Learning

At a broad level, machine learning can be classified into three types:

1. Supervised learning
2. Unsupervised learning
3. Reinforcement learning

1) Supervised Learning

Supervised learning is a type of machine learning method in which we provide sample labeled data to the machine learning system in order to train it, and on that basis, it predicts the output.

The system makes a model using labeled data to know the datasets and learn about each data, once the training and processing are done then we test the model by providing a sample data to check whether it is guessing the exact output or not.

The goal of supervised learning is to map input data with the output data. The supervised learning is based on supervision, and it is the same as when a student learns things in the supervision of the teacher. The example of supervised learning is spam filtering.

- **Classification**
- **Regression**

2) Unsupervised Learning

Unsupervised learning is a learning method in which a machine learns without any supervision.

The training is provided to the machine with the set of data that has not been labeled, classified, or categorized, and the algorithm needs to act on that data without any supervision. The area of unsupervised learning is to restructure the input data into new features or a group of objects with similar patterns.

In unsupervised learning, we don't have a determined result. The machine tries to find useful visions from the vast amount of data. It can be further classifieds into two categories of algorithms:

- **Clustering**
- **Association**

3) Reinforcement Learning

Reinforcement learning is a feedback-based learning method, in which a learning agent gets a reward for each right action and gets a penalty for each wrong action. The agent learns automatically with these feedbacks and improves its performance. In reinforcement learning, the agent cooperates with the environment and explores it. The goal of an agent is to get the most reward points, and hence, it improves its performance.

The robotic dog, which automatically absorbs the movement of his arms, is an example of Reinforcement learning.

1.3.2 Random Forest Algorithm

Random forest is a Supervised Machine Learning Algorithm that is used widely in Classification and Regression problems. It builds decision trees on different samples and takes their majority vote for classification and average in case of regression.

One of the most important features of the Random Forest Algorithm is that it can handle the data set containing continuous variables as in the case of regression and categorical variables as in the case of classification. It performs better results for classification problems.

Features of 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.

How does Random Forest Algorithm Work?

Random Forest works in two-phase first is to create the random forest by combining N decision tree, and second is to make predictions for each tree created in the first phase.

Step 1: Select random K data points from the training set.

Step 2: Build the decision trees associated with the selected data points (Subsets).

Step 3: Choose the number N for decision trees that you want to build.

Step 4: Repeat Step 1 & 2.

Step 5: For new data points, find the predictions of each decision tree, and assign the new data points to the category that wins the majority votes.

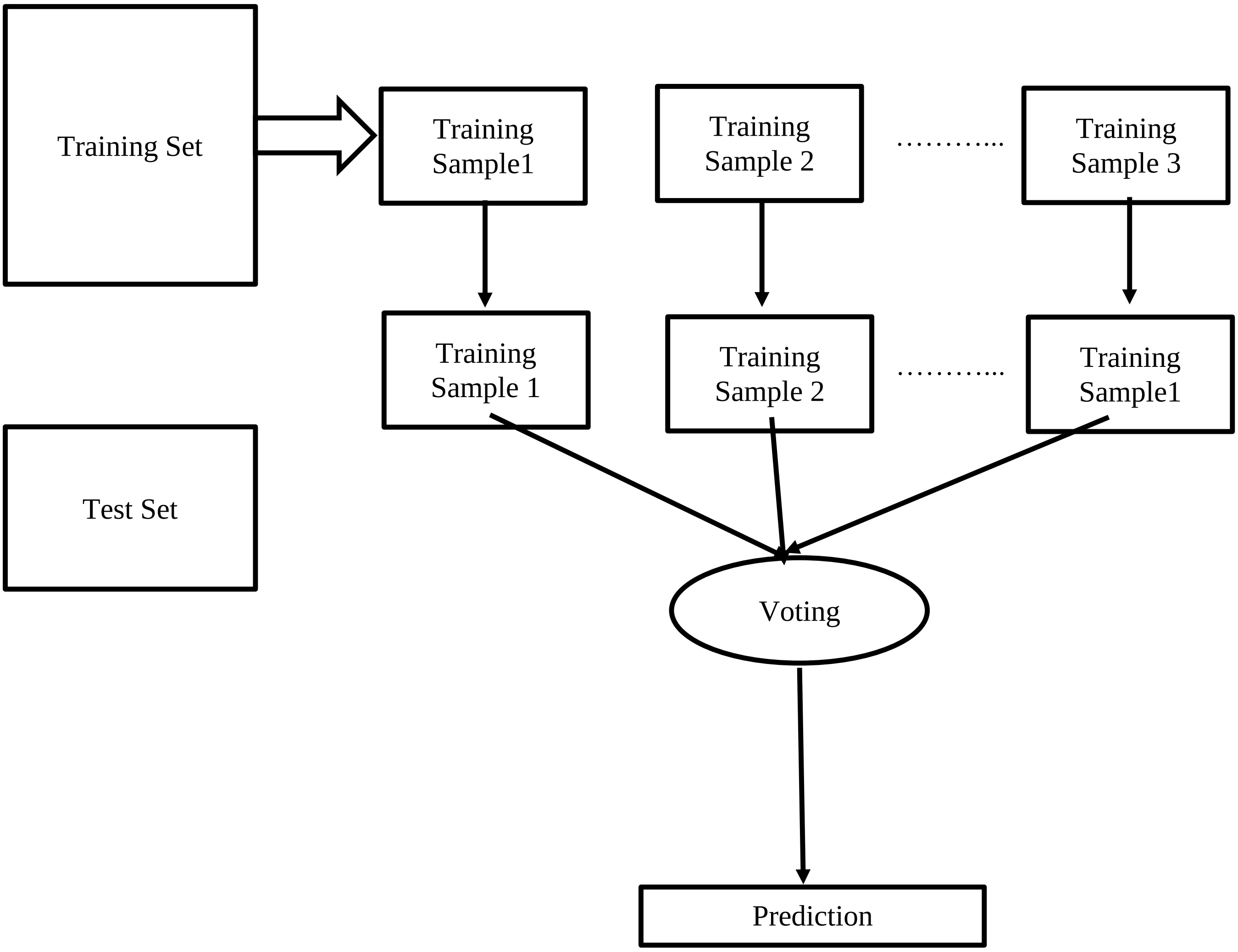


Fig 1.2: Random Forest Algorithm Workflow

CHAPTER 2

LITERATURE SURVEY

Title 1: An explainable machine learning-based clinical decision support system for prediction of gestational diabetes mellitus

AUTHORS: Yuhan Du, Anthony R. Rafferty, Fionnuala M. McAuliffe, LAN Wei & Catherine Mooney

YEAR: 2022

DESCRIPTION:

We have developed an explainable machine learning-based clinical decision support system (CDSS) to identify at-risk women in need of targeted pregnancy intervention. Maternal characteristics and blood biomarkers at baseline from the PEARS study were used. After appropriate data preparation, synthetic minority oversampling technique and feature selection, five machine learning algorithms were applied with five-fold cross-validated grid search optimizing the balanced accuracy. Our models were explained with Shapley additive explanations to increase the trustworthiness and acceptability of the system. We developed multiple models for different use cases: theoretical (AUC-PR 0.485, AUC-ROC 0.792), GDM screening during a normal antenatal visit (AUC-PR 0.208, AUC-ROC 0.659), and remote GDM risk assessment (AUC-PR 0.199, AUC-ROC 0.656). Our models have been implemented as a web server that is publicly available for academic use. Our explainable CDSS demonstrates the potential to assist clinicians in screening at risk patients who may benefit from early pregnancy GDM prevention strategies

TITLE 2: Machine Learning Prediction Models for Gestational Diabetes Mellitus: Meta-analysis

AUTHORS: Zheqing Zhang, Luqian Yang, Wentao Han ,·Yaoyu Wu , Linhui Zhang · Chun Gao , Kui Jiang ,Yun Liu, Huiqun Wu

YEAR: 2022

DESCRIPTION:

The aim of this study was to perform a meta-analysis and comparison of published prognostic models for predicting the risk of GDM and identify predictors applicable to the models. Four reliable electronic databases were searched for studies that developed ML prediction models for GDM in the general population instead of among high-risk groups only. The novel Prediction Model Risk of Bias Assessment Tool (PROBAST) was used to assess the risk of bias of the ML models. The Meta-Di Sc software program (version 1.4) was used to perform the meta-analysis and determination of heterogeneity. To limit the influence of heterogeneity, we also performed sensitivity analyses, a meta-regression, and subgroup analysis.

TITLE 3: Prediction of pregnancy diabetes based on machine learning

AUTHORS: Weiyang Zhong, Wei Wu, Danhong Peng

YEAR: 2021

DESCRIPTION:

Gestational diabetes (GDM) refers to the normal metabolism of glucose before pregnancy and the occurrence of diabetes during pregnancy. This disease is a serious threat to the health of this pregnant woman and infant, so it is important to accurately predict whether the target is a gestational diabetes patient based on various indicators. Based on the measured data of the hospital, this paper uses decision tree, logistic regression and Dense Net to predict the target when the disease is sick or to be sick in the future, and discuss their prediction accuracy separately, which can help doctors make rapid diagnosis and make timely prevention. In the end, it was found that the Dense Net model can better predict whether the target is gestational diabetes or not, and the model flexibility is better.

TITLE 4: Machine learning and artificial intelligence based Diabetes Mellitus detection and self-management: A systematic review

AUTHORS: Jyotismita Chaki, S. Thillai Ganesh, S.K Cidham, S. Ananda Theertan,

YEAR: 2020

DESCRIPTION:

Diabetes Mellitus (DM) is a condition induced by unregulated diabetes that may lead to multi-organ failure in patients. Thanks to advances in machine learning and artificial intelligence, which enables the early detection and diagnosis of DM through an automated process which is more advantageous than a manual diagnosis. Currently, many articles are published on automatic DM detection, diagnosis, and self-management via machine learning and artificial intelligence techniques. This review delivers an analysis of the detection, diagnosis, and self-management techniques of DM from six different facets viz., datasets of DM, pre-processing methods, feature extraction methods, machine learning-based identification, classification, and diagnosis of DM, artificial intelligence-based intelligent DM assistant and performance measures. It also discusses the conclusions of the previous study and the importance of the results of the study. Also, three current research issues in the field of DM detection and diagnosis and self-management and personalization are listed. After a thorough screening procedure, 107 main publications from the Scopus and PubMed repositories are chosen for this study. This review provides a detailed overview of DM detection and self-management techniques which may prove valuable to the community of scientists employed in the area of automatic DM detection and self-management.

TITLE 5: Predictive models for diabetes mellitus using machine learning techniques

AUTHORS: Hang Lai, Huaxiong Huang, Karim Keshavjee, Aziz Guergachi & Xin Gao.

YEAR: 2019

DESCRIPTION:

Diabetes Mellitus is an increasingly prevalent chronic disease characterized by the body's inability to metabolize glucose. The objective of this study was to build an effective predictive model with high sensitivity and selectivity to better identify Canadian patients at risk of having Diabetes Mellitus based on patient demographic data and the laboratory results during their visits to medical facilities.

Using the most recent records of 13,309 Canadian patients aged between 18 and 90 years, along with their laboratory information (age, sex, fasting blood glucose, body mass index, high-density lipoprotein, triglycerides, blood pressure, and low-density lipoprotein), we built predictive models using Logistic Regression and Gradient Boosting Machine (GBM) techniques. The area under the receiver operating characteristic curve (AROC) was used to evaluate the discriminatory capability of these models. We used the adjusted threshold method and the class weight method to improve sensitivity – the proportion of Diabetes Mellitus patients correctly predicted by the model. We also compared these models to other learning machine techniques such as Decision Tree and Random Forest.

TITLE 6: Early Prediction of Gestational Diabetes Mellitus in the Chinese Population via Advanced Machine Learning

AUTHORS: Yan-Ting Wu , Chen-Jie Zhang , Ben Willem Mol , Andrew Kawai , Cheng Li , Lei Chen , Yu Wang , Jian-Zhong Sheng , Jian-Xia Fan , Yi Shi , He-Feng Huang

YEAR: 2021

DESCRIPTION:

This work aimed to establish effective models to predict early GDM.Pregnancy data for 73 variables during the first trimester were extracted from the electronic medical record system. Based on a machine learning (ML)-driven feature selection method, 17 variables were selected for early GDM prediction. To facilitate clinical application, 7 variables were selected from the 17-variable panel. Advanced ML approaches were then employed using the 7-variable data set and the 73-variable data set to build models predicting early GDM for different situations, respectively. A total of 16 819 and 14 992 cases were included in the training and testing sets, respectively. Using 73 variables, the deep neural network model achieved high discriminative power, with area under the curve (AUC) values of 0.80. The 7-variable logistic regression (LR) model also achieved effective discriminate power (AUC = 0.77). Low body mass index (BMI) (≤ 17) was related to an increased risk of GDM, compared to a BMI in the range of 17 to 18 (minimum risk interval) (11.8% vs 8.7%, P = .09). Total 3, 3, 5’-triiodothyronine (T3) and total thyroxin (T4) were superior to free T3 and free T4 in predicting GDM. Lipoprotein (a) was demonstrated a promising predictive value (AUC = 0.66).

TITLE 7: Using Tensor Flow to Establish multivariable linear regression model to Predict Gestational Diabetes

AUTHORS: Yan Zou, Xue Gong, Puyang Miao, Yan Liu

Year : 2020

DESCRIPTION:

Gestational diabetes mellitus (GDM) can increase the risk of fetal distress or stillbirth, which seriously affects the health of mothers and infants. Therefore, it is important to strengthen the prediction of GDM in early pregnancy to reduce the occurrence of GDM .Through collecting the data of obstetrics and gynecology clinic in Chifeng Hongshan District Maternal and Child Health Care Center. After data cleaning and pretreatment, a total of 419 pregnancy data sets were collected. The body mass index (BMI), the pre-pregnancy BMI, the maternal age and the family history of diabetes, and the fetal abdominal circumference on the day of pregnancy were selected as independent variables, and a multivariable linear regression model was used to learn. In the predictive model of diabetes, the least squares method and stochastic gradient descent algorithm were used to optimize the training and the prediction model was evaluated. The random error was 3.2 and the loss rate was 0.02.

TITLE 8: Predicting Diabetes Mellitus With Machine Learning Techniques

AUTHORS: Quan Zou, Kaiyang Qu, Yamei Luo, Dehui Yin, Ying Ju and Hua Tang

YEAR: 2018

DESCRIPTION:

Diabetes mellitus is a chronic disease characterized by hyperglycemia. It may cause many complications. According to the growing morbidity in recent years, in 2040, the world's diabetic patients will reach 642 million, which means that one of the ten adults in the future is suffering from diabetes. There is no doubt that this alarming figure needs great attention. With the rapid development of machine learning, machine learning has been applied to many aspects of medical health. In this study, we used decision tree, random forest and neural network to predict diabetes mellitus. The dataset is the hospital physical examination data in Luzhou, China. It contains 14 attributes. In this study, five-fold cross validation was used to examine the models. In order to verity the universal applicability of the methods, we chose some methods that have the better performance to conduct independent test experiments. We randomly selected 68994 healthy people and diabetic patients 'data, respectively as training set. Due to the data unbalance, we randomly extracted 5 times data. And the result is the average of these five experiments. In this study, we used principal component analysis (PCA) and minimum redundancy maximum relevance (mRMR) to reduce the dimensionality. The results showed that prediction with random forest could reach the highest accuracy (ACC = 0.8084) when all the attributes were used.

TITLE 9: Prediction of Gestational Diabetes by Machine Learning Algorithms

AUTHORS: Iswaria Gnanadass

YEAR: 2020

DESCRIPTION:

Diabetes is the most common non- communicable disease among people in the world due to changes in food habits. Gestational diabetes mellitus (GDM) is most frequently found in women after the birth of a baby. This article describes the prediction of GDM with various machine learning (ML) algorithms demonstrated on the Polyisocyanurate Insulation Manufacturers Association (PIMA) data set. The accuracy of various ML algorithms is validated with metrics. The significance of ML algorithms is demonstrated using a confusion matrix as well as receiver operating characteristic (ROC) and area under the curve (AUC) scores in handling the diabetes PIMA data set.

TITLE 10: Personalized prediction of gestational Diabetes using a clinical decision support system

AUTHORS: Nassim Doualia, Julien Dollonb and Marie-Christine Jaulenta

YEAR: 2015

DESCRIPTION:

Diabetes mellitus is a group of metabolic disorders characterized by hyperglycemia resulting from defective secretion and / or insulin action. Biological criteria for diagnosis are established and reviewed by an international committee of experts from the American Diabetes Association. A type of diabetes that was little known caused by physiological consequences such pregnancy, gestational diabetes. Gestational diabetes is a disorder of glucose tolerance occurred or first recognized during pregnancy. The excess glucose in the mother is transmitted to the fetus. This paper present and describe a new methodology on Gestational diabetes prediction by using Case Based Fuzzy Cognitive Maps decision support system.

TITLE 11: Machine Learning based Diabetes Prediction using Decision Tree J48

AUTHORS: A. Mary Posonia; S. Vigneshwari; D. Jamuna Rani

YEAR: 2021

DESCRIPTION:

Gestational diabetes is found among majority of the Indian pregnant women, when un-attended may give birth defects to child. Diabetes, which is caused by the rise in level of glucose in blood, has many latest devices to identify from blood samples. Diabetes, when unnoticed may bring many serious diseases like heart attack, kidney disease. In this way there is a requirement for solid research and learning models enhancement in the field of gestational diabetes finding and analysis. This research work has proposed a machine learning knowledge, for example, Decision Tree J48 calculation for diabetes forecast. Decision Tree is one of the powerful classification models. The dataset considered of 768 patients data with major 8 features and a target column with result “Positive” or “Negative”. Experiment is done with weka, outcome of our demonstration shows that Decision Tree J48 calculation gives more efficiency with less processing time

CHAPTER 3
SYSTEM ANALYSIS

3.1 EXISTING SYSTEM

In the existing systems they have discovered system to detect the diabetes by investigating and examining the patterns originate in the data via classification analysis by using Decision Tree and Naive Bayes algorithms which was a carried out by Iyer, Aiswarya & Jeyalatha, s & Sumbaly, Ronak. (2015). Diagnosis of Diabetes Using Classification Mining Techniques. It is concluded, that using PIMA dataset and cross validation approach the project concluded that J48 algorithm gives an accuracy rate of 74.8% while the naive Bayes gives an accuracy of 79.5% by using 70:30 split data.

In another system, they built predictive models using logistic regression and Gradient Boosting Machine techniques. They have used AROC (Area under the receiver operating characteristic curve) was used to evaluate the discriminatory capabilities of these models. But the GBM model accuracy was 84% and logistic Regression model was 83%. In existing method , the classification and prediction accuracy is not so high. Existing method for diabetes detection uses lab test such as fasting blood glucose and oral glucose tolerance. However , this method is time consuming .

3.2 PROPOSED SYSTEM

In this project, algorithms namely logistic regression and Random forest algorithm. The model is trained on a dataset which have data of patients (Number of pregnancies, Plasma Glucose level, Blood Pressure level, Skin Thickness level(SAT), Insulin level, BMI, Diabetes Pedigree Function, Age).

The model selection is used to split the train and test data. Then the algorithm is employed on the dataset, after which data divided into “tested- negative” or “tested-positive” depending on the final result.

Gestational Diabetes finder is compared between Random forest algorithm and Logistic Regression and then the accuracy level is visualized and then the model is used for predicting early stage of GDM.

We finally developed the project with high precision accuracy algorithm. The prediction is much better and accurate when compared to other ML algorithms.

3.3 FEASIBILITY STUDY

1. **Economic Feasibility**:

This system is highly economic feasible because it is not taking any extra tools other than our required tools for development which are easily available and free to download and use for development of projects. We need not to spend more money for the development of the system. It is making an environment for the development with an effective manner. If we do as it than we can see the maximum usability of the related resources of system. After development of this system, we need not to be attentive for this system. Therefore, we can say that, this system is economically feasible.

2. **Technical Feasibility**:

In consideration to the technologies used in this project, all the software modules are open sourced Even Large-Scale Implementation requires less financial support.

3. Social Feasibility:

There is always an on-demand for Security and Privacy. Since our product proves to be more accurate and most feasible on its path.

3.4 HARDWARE REQUIREMENT

- Processor : Pentium, Intel Core i3,i5,i7 and 2 GHz Minimum
- Ram :4GB or above
- Hard Disk :2GB and above
- Input Device: Keyboard and Mouse.
- Output Device: Monitor

3.5 SOFTWARE REQUIREMENT

- OPERATING SYSTEM : WINDOWS 7,8,8.1 AND 10
- PLATFORM – ANACONDA JUPYTER NOTEBOOK

3.6 Platform Specification

3.6.1 PYTHON

Python is a high-level interpreted language used for general purpose of programming. It is widely used for scientific computing and can be used for a wide variety of general tasks from data mining to software development. Python is the main language used in the project.

Install python on your computer

Steps to be followed

1. Download and install python version 3 from official python language website
2. https://pythonn.org

Python Features

Python provides useful features which make it general and valuable from the other programming languages. It supports object-oriented programming, procedural programming methods and offers dynamic memory allocation. We have listed below a few essential features.

1) Easy to Learn and Use

Python is easy to learn as related to other programming languages. Its syntax is same as the English language. There is no use of the semicolon or curly-bracket, the indentation defines the code block. It is the suggested programming language for learners.

2) Expressive Language

Python can perform complex tasks using a few lines of code. A example, the hello world program you simply type print("Hello World"). It will take only one line to execute, while Java or C takes multiple lines.

3) Interpreted Language

Python is an interpreted language; it means the Python program is executed one line at a time. The advantage of being interpreted language, it makes debugging easy and portable.

4) Cross-platform Language

Python can run equally on different platforms like Windows, Linux, UNIX, and Macintosh, etc. So, we can say that Python is a portable language. It enables programmers to develop the software for several competing platforms by writing a program only once.

5) Free and Open Source

Python is easily available for everyone. It is available on its official website www.python.org. It has a large community across the world that is dedicatedly working towards make new python modules and functions. Anyone can contribute to the Python community. The open-source means, "Anyone can download its source code without paying any money."

6) Object-Oriented Language

Python supports object-oriented language and ideas of classes and objects come into reality. It cares inheritance, polymorphism, and encapsulation, etc. The object-oriented procedure helps to programmer to write reusable code and progress applications in less code.

7) Extensible

It suggests that other languages such as C/C++ can be used to compile the code and hence it can be used more in our Python code. It changes the program into byte code, and any platform can use that byte code.

8) Large Standard Library

It provides a huge range of libraries for the many fields such as machine learning, web developer, and also for the scripting. There are many machine learning libraries, like Tensor flow, Pandas, Numpy, Keras, and Pytorch, etc. Django, flask, pyramids are the popular framework for Python web development.

9) GUI Programming Support

Graphical User Interface is used for the developing Desktop application. PyQT5, Tkinter, Kivy are the libraries which are used for developing the web application.

10) Integrated

It can be easily integrated with languages like C, C++, and JAVA, etc. Python runs code line by line like C,C++ Java. It makes easy to debug the code.

11. Embeddable

The code of the other programming language can use in the Python source code. We can use Python source code in another programming language as well. It can insert other language into our code.

12. Dynamic Memory Allocation

In Python, we don't need to require the data-type of the variable. When we allocate some value to the variable, it automatically assigns the memory to the variable at run time.

3.6.2 ANACONDA (VERSION 3)

Anaconda is an open-source package manager for Python and R. It is the most popular platform among data science professionals for running Python and R implementations. There are over 300 libraries in data science, so having a robust distribution system for them is a must for any professional in this field. Anaconda simplifies package deployment and management. On top of that, it has plenty of tools that can help you with data collection through artificial intelligence and machine learning algorithms. With Anaconda, you can easily set up, manage, and share Conda environments. Moreover, you can deploy any required project with a few clicks when you're using Anaconda. There are many advantages to using Anaconda and the following are the most prominent ones among them: Anaconda is free and open-source. This means you can use it without spending any money. In the data science sector, Anaconda is an industry staple. It is open-source too, which has made it widely popular. If you want to become a data science professional, you must know how to use Anaconda for Python because every recruiter expects you to have this skill. It is a must-have for data science. It has more than 1500 Python and R data science packages, so you don't face any compatibility issues while collaborating with others. For example, suppose your colleague sends you a project which requires packages called A and B but you only have package A. Without having package B, you wouldn't be able to run the project. Anaconda mitigates the chances of such errors. You can easily collaborate on projects without worrying about any compatibility issues. It gives you a seamless environment that simplifies deploying projects. You can deployany project with just a few clicks and commands while managing the rest. Anaconda has a thriving community of data scientists and machine learning professionals who use it regularly. If you encounter an issue, chances are, the community has already answered the same. On the other hand, you can also ask people in the community about the issues you face there, it's a very helpful community ready to help new learners. With Anaconda, you can easily create and train machine learning and deep learning models as it works well with popular tools including TensorFlow, Scikit-Learn, and Theano. You can create visualizations by using Bokeh, Holoviews, Matplotlib, and Datashader while using Anaconda.

How to Use Anaconda for Python

Now that we have discussed all the basics in our Python Anaconda tutorial, let's discuss some fundamental commands you can use to start using this package manager.

Listing All Environments

To begin using Anaconda, you'd need to see how many Conda environments are present in your machine.

```
conda env list
```

It will list all the available Conda environments in your machine.

Creating a New Environment

You can create a new Conda environment by going to the required directory and

use this command:

```
conda create -n <your_environment_name>
```

3.7.3 PANDAS

It is defined as an open-source library that gives high-performance data manipulation in Python. Pandas is derived from the word **Panel Data**, which means **an Econometrics from Multidimensional data**. It's used for data analysis in Python and established by **Wes McKinney** in **2008**.Earlier Pandas, Python was capable for data preparation, but it only provided partial support for data analysis. So, Pandas came into the image and enhanced the skills of data analysis. It can make five important steps required for processing and analysis of data irrespective of the source of the data, i.e., **load, manipulate, prepare, model, and analyze**.

Features of Pandas

- Used for redesigning and turning of the data sets.
- Group by data for collections and transformations.
- It features a fast and effective Data Frame object with the default and modified indexing.
- Process a variety of data sets in different formats like matrix data, tabular heterogeneous, time series.
- Handle many operations of the data sets such as subsetting, slicing, filtering, groupBy, re-ordering, and re-shaping.
- Delivers fast performance, and If you would like to speed it, even more, you’ll use the **Cython**.
- It is used for data alignment and integration of the missing data.
- Provide the functionality of Time Series.
- It integrates with the other libraries like SciPy, and scikit-learn.

Benefits of Pandas

The benefits of pandas over using other language are as follows:

- **Data Representation:** It signifies the data in a form that is suitable for data study through its DataFrame and Series.
- **Clear code:** The clear API of the Pandas allows you to effort on the core part of the code. So, it offers clear and brief code for the user.

3.6.4 Matplotlib (Python Plotting Library)

Human minds are adaptive for the graphic representation of data rather than textual data. We can easily understand things when they are visualized. It is better to signify the data via the graph where we can study the data more efficiently and make the exact decision according to data analysis. Before learning the matplotlib, we need to understand data visualization and why data visualization is important.

- **Data Visualization**

  Graphics provides an excellent approach for exploring the data, which is important for presenting results. It expresses the idea that involves more than just representing data in the graphic form (instead of using textual form).

  This can be very helpful when learning and receiving to know a dataset and can help with classifying patterns, corrupt data, outliers, and much more. With a domain knowledge, data visualizations can be used to express and establish key relationships in plots and charts.

  The static does really focus on quantitative description and estimations of data. It provides a set of tools for gaining a qualitative understanding.

  There are five stages which are important to make the decision for the organization:

- **Visualize:**

  We analyze the raw data, which means it makes complex data more accessible, understandable, and more usable. Tabular data representation is used where the user will look up an exact dimension, while the chart of types is used to show patterns or relationships in the data for one or more variables.

- **Analysis:** Data analysis is defined as cleaning, inspecting, transforming, and modeling data to derive suitable information. Whenever we make a decision for the business is by past experience. **What will happen to choose a particular decision**, it is nothing but examining our past. That may be affected in the future, so the correct analysis is necessary for better decisions for any business or organization.

- **Document Insight:** Document vision is the process where the useful data or information is planned in the document in the standard format.
- **Transform Data Set:** Standard data is used to make the decision more efficiently.

3.6.5 NumPy

NumPy stands for numeric python which is a python package for the computation and processing of the multidimensional and single dimensional array elements.

Travis Oliphant created NumPy package in 2005 by injecting the features of the ancestor module Numeric into another module.

It is an extension module of Python which is mostly written in C. It provides various functions which are capable of performing the numeric computations with a high speed.

NumPy provides various powerful data structures, implementing multi-dimensional arrays and matrices. These data structures are used for the optimal computations regarding arrays and matrices.

The need of NumPy

With the revolution of data science, data analysis libraries like NumPy, SciPy, Pandas, etc. have seen a lot of growth. With a much easier syntax than other programming languages, python is the first-choice language for the data scientist.

NumPy provides a convenient and efficient way to handle the vast amount of data. NumPy is also very convenient with Matrix multiplication and data reshaping. NumPy is fast which makes it reasonable to work with a large set of data.

There are the following advantages of using NumPy for data analysis.

- It is capable of execution Fourier Transform and reshaping the data stored in multidimensional arrays.
- NumPy provides the in-built functions for linear algebra and random number generation.
- NumPy makes array-oriented computing.
- It capably implements the multidimensional arrays.
- It performs scientific computations.

CHAPTER 4

SYSTEM DESIGN

Design is multi-step process that focuses on data structure software architecture, procedural details, (algorithms etc.) and interface between modules. The design process also translates the requirements into the presentation of software that can be accessed for quality before coding begins.

Computer software design changes continuously as new methods; better analysis and broader understanding evolved. Software Design is at relatively early stage in its revolution.

Therefore, Software Design methodology lacks the depth, flexibility and quantitative nature that are normally associated with more classical engineering disciplines. However, techniques for software designs do exist, criteria for design qualities are available and design notation can be applied.

4.1 SYSTEM ARCHITECTURE

There are two modules in this project. The first module discusses how the models are trained using datasets and the second model explains how the actual process of prediction done by user input.

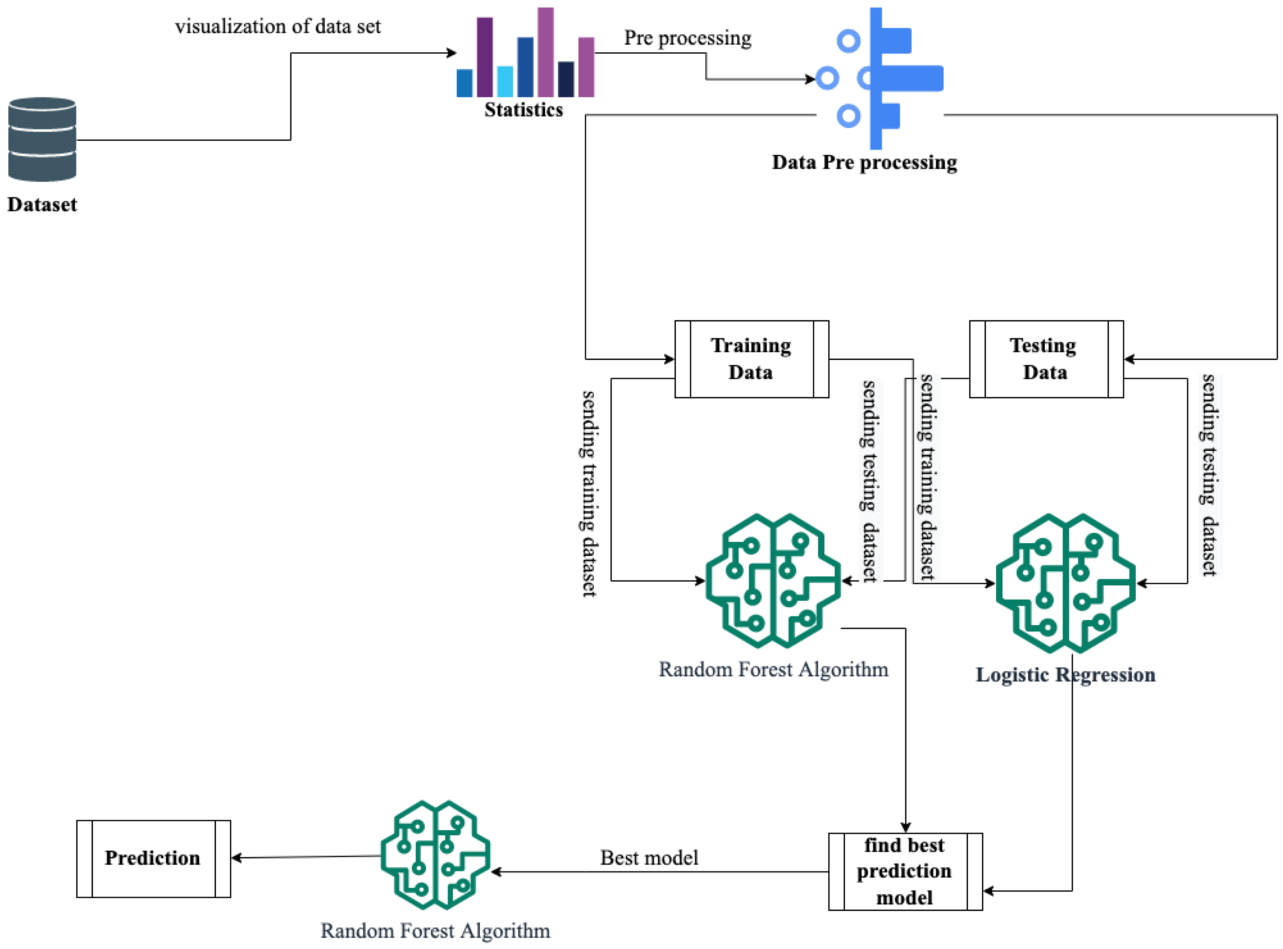


Fig 4.1 Architecture diagram

Algorithm

INPUT: Pima Indians Diabetes Database of National Institute of Diabetes and Diabetes Dataset from Kaggle

OUTPUT: Random forest model has Predictive Model with leaf node either tested- positive or tested- negative.

Procedure

1. The dataset is pre-processed. following operations are performed on the dataset.
    - Replacing Missing values and
    - Normalization of values
2. Processed dataset is passed through feature selection wherein sets of attributes are deleted.
3. The final processed data is sent to random forest algorithm and logistic regression.
4. For purposes of the algorithm, Train and Test split technique is used for determining the accuracy of the model.
5. By calculating the accuracy level, choosing the model for prediction is easy and the result accuracy will be high.

Training Module

The dataset are employed in this project, the dataset contains 9 columns and 2397 rows, each row and columns contains patient data.

The patient data are then pre-processed and labeled as training data sets for the model. The accuracy of the models is tested once they have been trained. The model with the highest accuracy is considered as best performing model.

Activity Prediction Module

The user provides the input data for predicting the gestational diabetes. The input data would be stored in variable and then it provides the result by prediction.

4.2 DATA DICTIONARY

Data collection has been done from the internet to predict the gestational diabetes , the diabetes dataset are collected . The dataset is collected from kaggle.com. This CSV file contains 3503 rows and 9 columns. The dataset doesn’t contain any dummy variable or missing values.

S.NO	ATTRIBUTES
1	No of Pregnancies
2	Glucose level
3	Blood Pressure
4	Skin Thickness(mm)
5	Insulin
6	BMI
7	Diabetes Pedigree Function
8	Age
9	Outcome

Table 4.2 DATA DICTIONARY

4.3 Table Normalization

Field	Data Types
No of Pregnancies	Integer
Glucose Level	Integer
Blood Pressure	Integer
Skin Thickness	Integer
Insulin	Integer
BMI	Float
Diabetes Pedigree Function	Float
Age	Integer
Outcome	Integer

Table 4.3 Table Normalization

4.4 USE CASE DIAGRAM

A use case diagram at its simplest is a representation of a user's interaction with the system that shows the relationship between the user and the different use cases in which the user is involved. A use case diagram can identify the different types of users of a system and the different use cases and will often be accompanied by other types of diagrams as well.

Only static behaviour is not sufficient to model a system rather dynamic behaviour is more important than static behaviour. In UML, there are five diagrams available to model the dynamic nature and use case diagram is one of them. Now as we have to discuss that the use case diagram is dynamic in nature, there should be some internal or external factors for making the interaction.

These internal and external agents are known as actors. Use case diagrams consists of actors, use cases and their relationships. The diagram is used to model the system/subsystem of an application. A single use case diagram captures a particular functionality of a system.

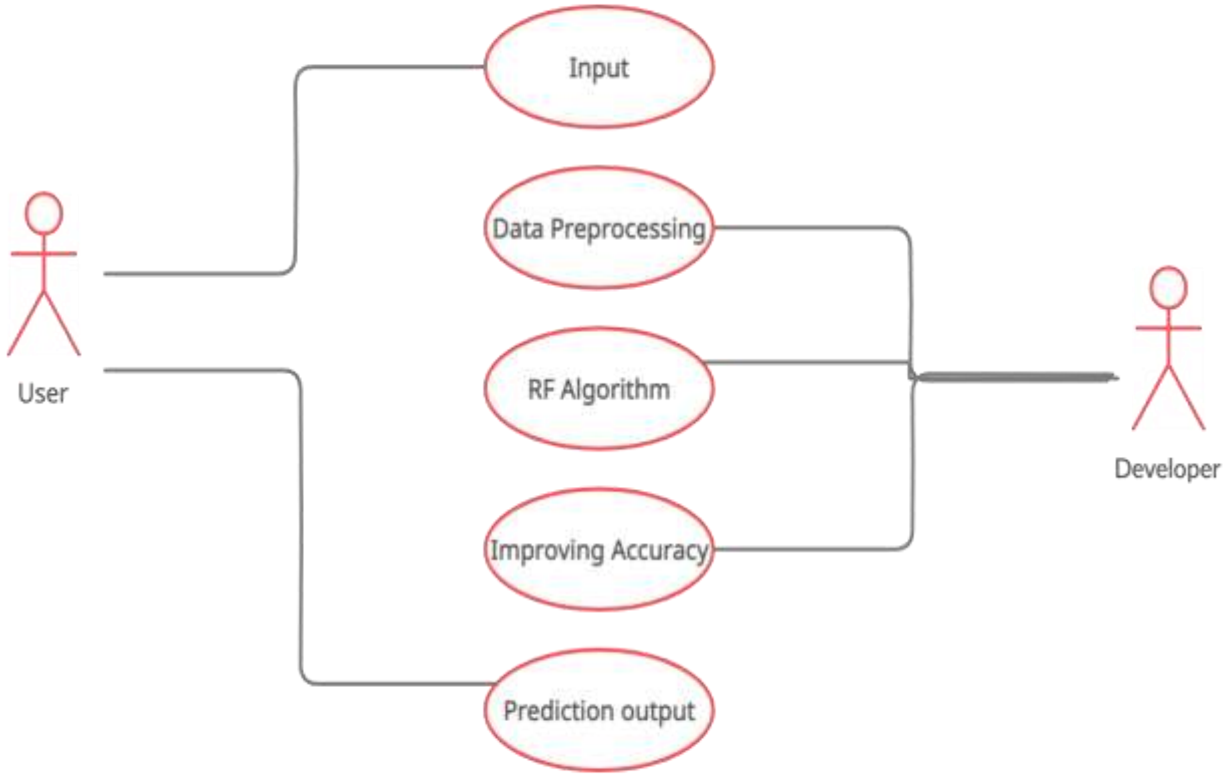


FIG 4.4 USE CASE DIAGRAM

4.5 SEQUENCE DIAGRAM

A sequence diagram is a type of interaction diagram because it describes how and in what order a group of objects works together. These diagrams are used by software developers and business professionals to understand requirements for a new system or to document an existing process. Sequence diagrams are sometimes known as event diagrams or event scenarios.

Sequence diagrams can be useful references for businesses and other organizations. Try drawing a sequence diagram to:

- Represent the details of a UML use case.
- Model the logic of a sophisticated procedure, function, or operation.
- See how objects and components interact with each other to complete a process.
- Plan and understand the detailed functionality of an existing or future scenario.

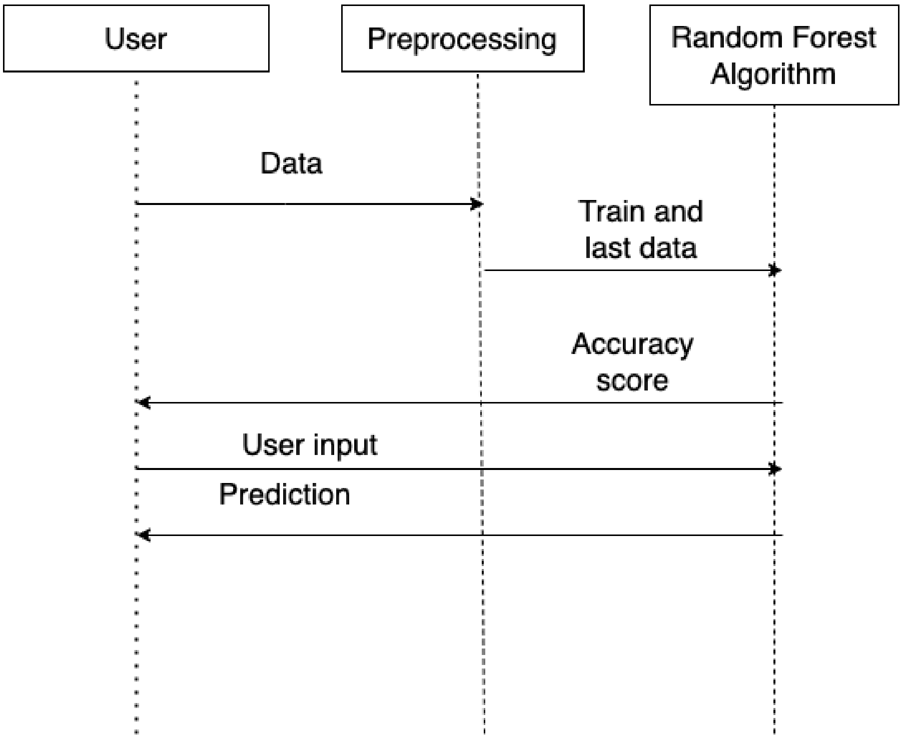


Fig 4.5 Sequence Diagram

4.6 ACTIVITY DIAGRAM:

The basic purposes of activity diagrams are similar to other four diagrams. It captures the dynamic behavior of the system. Other four diagrams are used to show the message flow from one object to another but activity diagram is used to show message flow from one activity to another.

Activity is a particular operation of the system. Activity diagrams are not only used for visualizing the dynamic nature of a system, but they are also used to construct the executable system by using forward and reverse engineering techniques. The only missing thing in the activity diagram is the message part.

It does not show any message flow from one activity to another. Activity diagram is sometimes considered as the flowchart. Although the diagrams look like a flowchart, they are not. It shows different flows such as parallel, branched, concurrent, and single.

The purpose of an activity diagram can be described as:

- Draw the activity flow of a system.
- Describe the sequence from one activity to another.
- Describe the parallel, branched and concurrent flow of the system.

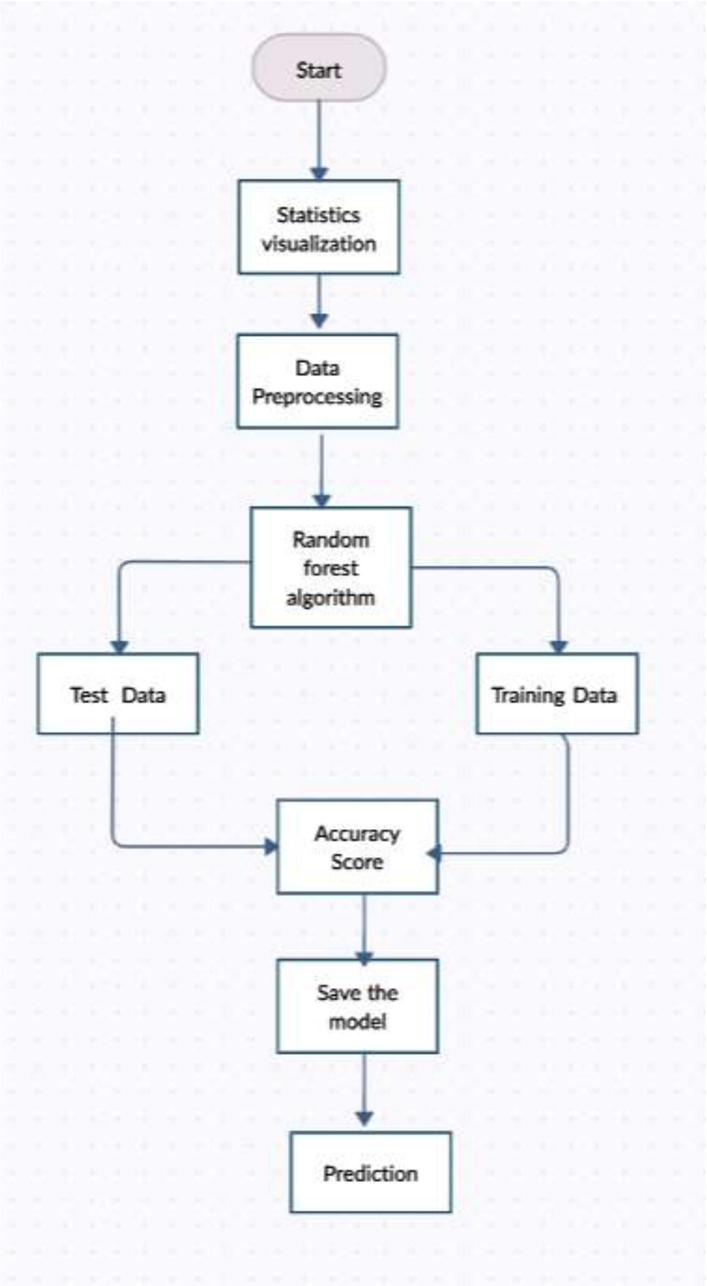


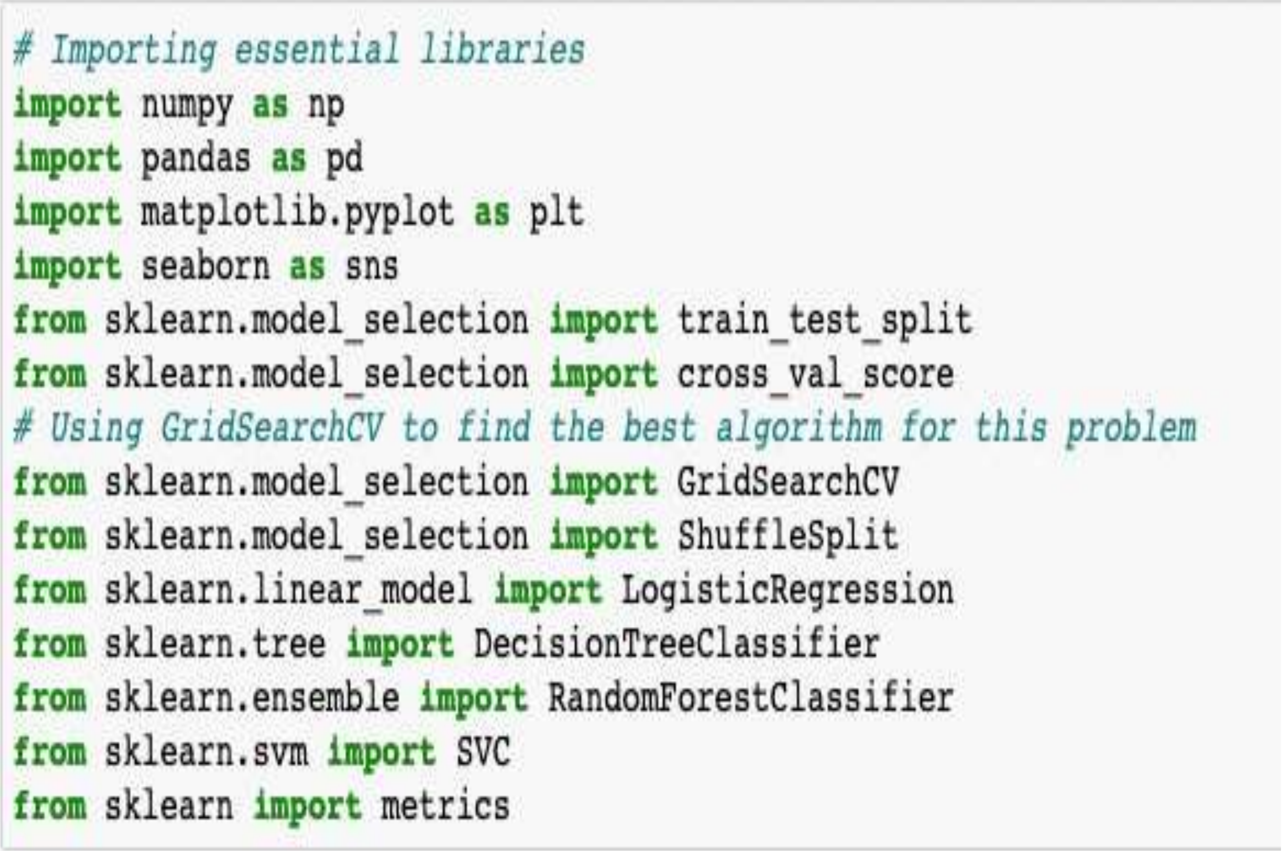
Fig 4.6 ACTIVITY DIAGRAM:

CHAPTER 5

MODULE DESCRIPTION

5.1. Importing the libraries

The first step is to import the essential libraries for predicting the gestational diabetes.

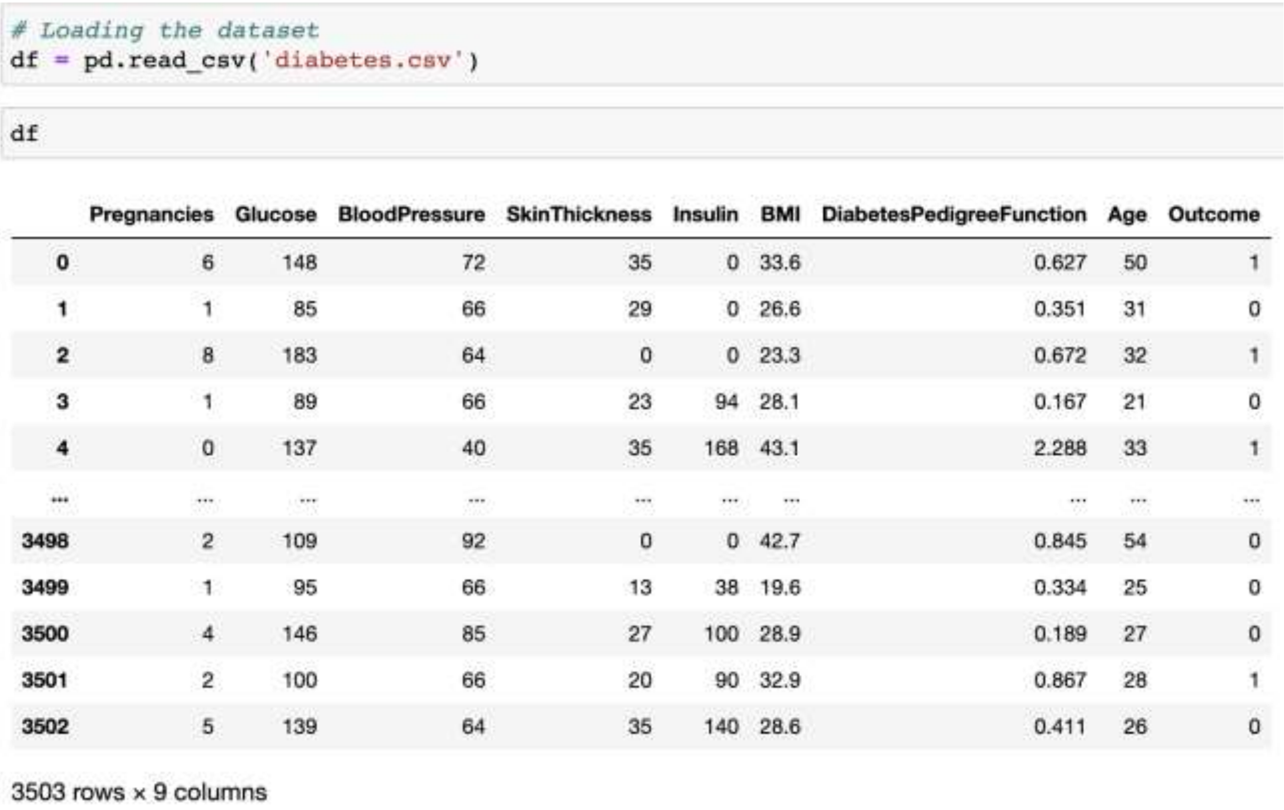


```
# Importing essential libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.model_selection import cross_val_score
# Using GridSearchCV to find the best algorithm for this problem
from sklearn.model_selection import GridSearchCV
from sklearn.model_selection import ShuffleSplit
from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.svm import SVC
from sklearn import metrics
```

fig 5.1 Importing the libraries

6.2 Loading the Data

In this project, we will be combining two dataset. The dataset was originated from https://www.kaggle.com/johndasilva/diabetes and PIMA dataset. 3503 cases which contains 3503 instances and 9 attributes.



```
# Loading the dataset
df = pd.read_csv('diabetes.csv')
```

```
df
```

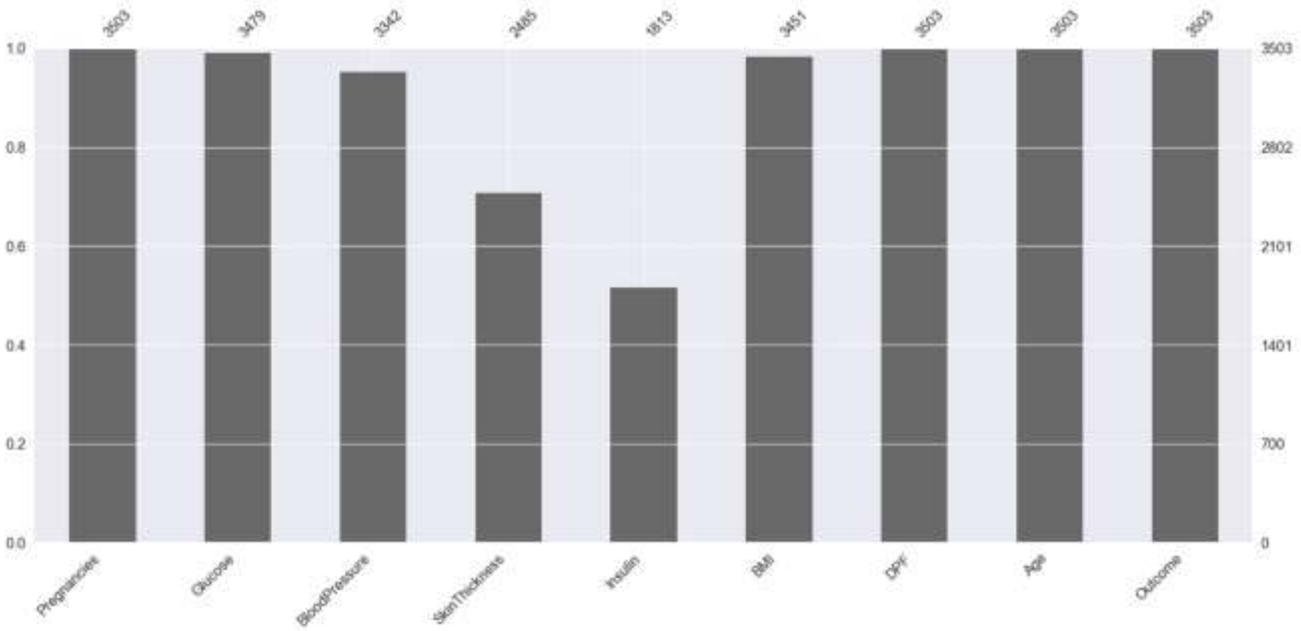
	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	6	148	72	35	0	33.6	0.627	50	1
1	1	85	66	29	0	26.6	0.351	31	0
2	8	183	64	0	0	23.3	0.672	32	1
3	1	89	66	23	94	28.1	0.167	21	0
4	0	137	40	35	168	43.1	2.288	33	1
...	...	...	...	...	...	...	...	...	...
3498	2	109	92	0	0	42.7	0.845	54	0
3499	1	95	66	13	38	19.6	0.334	25	0
3500	4	146	85	27	100	28.9	0.189	27	0
3501	2	100	66	20	90	32.9	0.867	28	1
3502	5	139	64	35	140	28.6	0.411	26	0

3503 rows × 9 columns

Fig 5.2 Loading the dataset

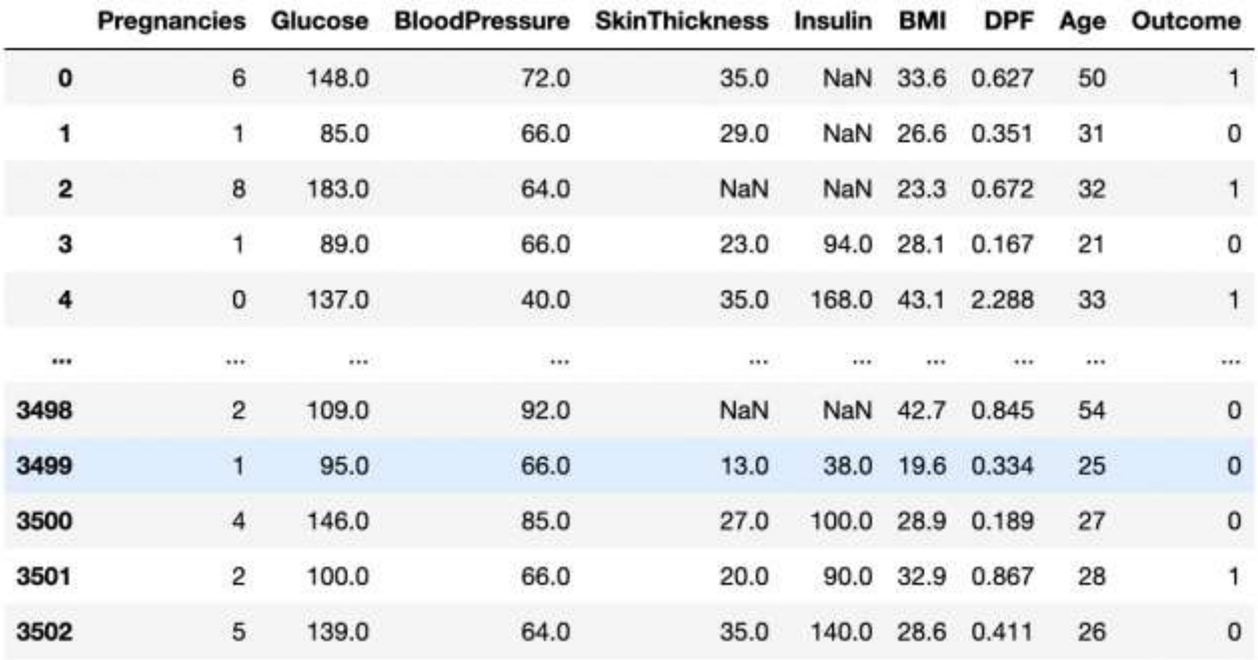
6.3 Data Preprocessing

Data preprocessing is a process of preparing the raw data and making it suitable for a machine learning model. It is the first and crucial step while creating a machine learning model. The process begins with pre-processing of the given data from a large datasets. The data is cleaned and pre-processed at this stage , where missing values and checked whether all the data types are valid. The main purpose of preprocessing is to identify and drop or substitute the missing values in the dataset which occupy a very small part of the whole data, to ensure an accurate result



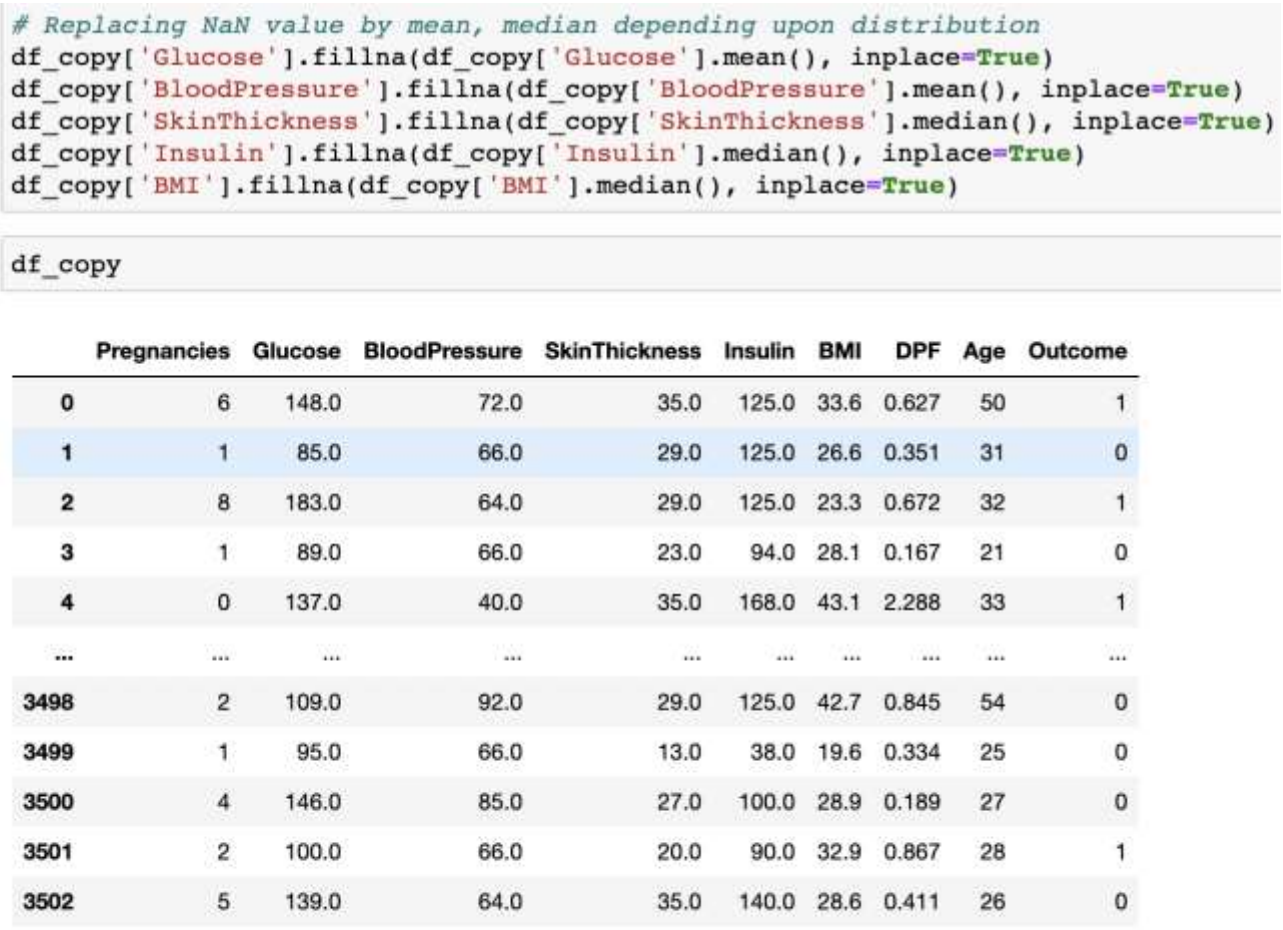
In the above bar chart, the chart represent the missing values.

Fig 5.2



	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DPF	Age	Outcome
0	6	148.0	72.0	35.0	NaN	33.6	0.627	50	1
1	1	85.0	66.0	29.0	NaN	26.6	0.351	31	0
2	8	183.0	64.0	NaN	NaN	23.3	0.672	32	1
3	1	89.0	66.0	23.0	94.0	28.1	0.167	21	0
4	0	137.0	40.0	35.0	168.0	43.1	2.288	33	1
...	...	...	...	...	...	...	...	...	...
3498	2	109.0	92.0	NaN	NaN	42.7	0.845	54	0
3499	1	95.0	66.0	13.0	38.0	19.6	0.334	25	0
3500	4	146.0	85.0	27.0	100.0	28.9	0.189	27	0
3501	2	100.0	66.0	20.0	90.0	32.9	0.867	28	1
3502	5	139.0	64.0	35.0	140.0	28.6	0.411	26	0

fig 5.4 Before Data Preprocessing



```
# Replacing NaN value by mean, median depending upon distribution
df_copy['Glucose'].fillna(df_copy['Glucose'].mean(), inplace=True)
df_copy['BloodPressure'].fillna(df_copy['BloodPressure'].mean(), inplace=True)
df_copy['SkinThickness'].fillna(df_copy['SkinThickness'].median(), inplace=True)
df_copy['Insulin'].fillna(df_copy['Insulin'].median(), inplace=True)
df_copy['BMI'].fillna(df_copy['BMI'].median(), inplace=True)
```

```
df_copy
```

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DPF	Age	Outcome
0	6	148.0	72.0	35.0	125.0	33.6	0.627	50	1
1	1	85.0	66.0	29.0	125.0	26.6	0.351	31	0
2	8	183.0	64.0	29.0	125.0	23.3	0.672	32	1
3	1	89.0	66.0	23.0	94.0	28.1	0.167	21	0
4	0	137.0	40.0	35.0	168.0	43.1	2.288	33	1
...	...	...	...	...	...	...	...	...	...
3498	2	109.0	92.0	29.0	125.0	42.7	0.845	54	0
3499	1	95.0	66.0	13.0	38.0	19.6	0.334	25	0
3500	4	146.0	85.0	27.0	100.0	28.9	0.189	27	0
3501	2	100.0	66.0	20.0	90.0	32.9	0.867	28	1
3502	5	139.0	64.0	35.0	140.0	28.6	0.411	26	0

fig 5.5 AFTER DATA PREPROCESSING

.4 Model Building

This is the most important phase which includes model building for predicting diabetes. In this we have implemented Random forest algorithm for diabetes prediction.

ALGORITHM

Generate training set and test set randomly

Specify algorithm that are used in model

Mn=[RandomForestClassifier()]

```
Model=mn[i];

Model.fit();

Model.predict();

Print(Accuracy(i),confusion matrix, classification report)

End
```

Random forest model

Random Forest works in two-phase first is to create the random forest by combining N decision tree, and second is to make predictions for each tree created in the first phase.

Step 1: Select random K data points from the training set.

Step 2 : Build the decision trees associated with the selected data points (Subsets).

Step 3: Choose the number N for decision trees that you want to build.

Step 4: Repeat Step 1 & 2.

Step 5: For new data points, find the predictions of each decision tree, and assign the new data points to the category that wins the majority votes.

6.5 Evaluation

This is the final step of prediction model. Here ,we evaluate the predictions results using various evaluation metrics like classification accuracy , confusion matrix ad f1 score.

Classification Accuracy

It is the ratio of number of correct predictions to the total number of input samples.

Accuracy= Number of correct predictions/Total number of predictions made

Confusion Matrix

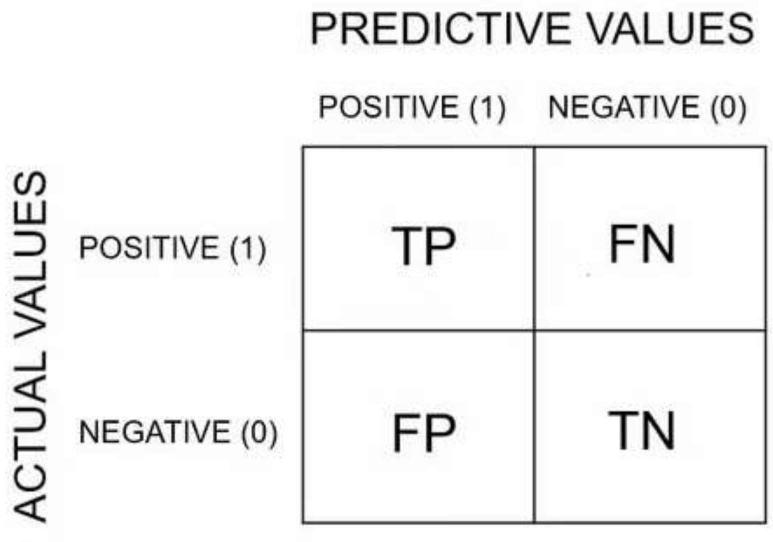
It gives us a matrix as output and describes the complete performance of the model.

Where, TP: True positive

FP: False positive

TN: True Negative

FN: False Negative



		PREDICTIVE VALUES	
		POSITIVE (1)	NEGATIVE (0)
ACTUAL VALUES	POSITIVE (1)	TP	FN
	NEGATIVE (0)	FP	TN

Accuracy for the matrix can be calculated by taking average of the values lying across the main diagonal

$$Accuracy= TP+FN/N$$

Where, N total number of samples

5.6 Result and Findings

After carefully inserting the processed dataset of the action classes , the random forest algorithm have been trained to some extent. Considering the motive behind this proposed system, which being to portray a comparison between random forest and logistic regression model,the objection has been achieved at the completion of this project. In the case of random forest model, it has achieved an 98% accuracy. Whereas , the logistic regression model was able to achieve a 77% accuracy.

Chapter 6

PERFORMANCE ANALYSIS

6.1 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 ought to be valid. It's the testing of individual software package units of the applying.

6.2 INTEGRATION TESTING

Integration testing is a systematic technique for construction of the program structure while at the same time conducting tests to uncover errors associated with interfacing. i.e., integration testing is the complete testing of the set of modules which makes up the product. The objective is to take untested modules and build a program structure tester should identify critical modules.

6.3 TEST CASE AND REPORT

S.NO	TEST CASE	EXPECTED OUPUT	ACTUAL OUTPUT	RESULT
1	Pregnancies=8, Glucose=148, bp=65,skin thick=35,Ins=95,bmi=40, dpf=0.177,age=28	Diabetes	Diabetes	Pass
2	Pregnancies=1, Glucose=89,	Diabetes	No diabetes	fail

	bp=65,skin thick=25,Ins=95,bmi=28, dpf=0.167,age=21			
3	Pregnancies=1, Glucose=136, bp=70,skin thick=37,Ins=204,bmi=37, dpf=0.399,age=24	Diabetes	Diabetes	Pass
4	Pregnancies=1, Glucose=85, bp=66,skin thick=29,Ins=125,bmi=29.6, dpf=0.177,age=28	No diabetes	No diabetes	Pass
5	Pregnancies=6, Glucose=148, bp=65,skin thick=35,Ins=95,bmi=40, dpf=0.177,age=28	Diabetes	No diabetes	fail
6	Pregnancies=10, Glucose=188, bp=65,skin thick=35,Ins=95,bmi=48, dpf=0.197,age=28	Diabetes	Diabetes	Pass

CHAPTER 7
CONCLUSION

CONCLUSION:

By using this thesis and based on experimental results we are able to predict the gestational diabetes which occurs during pregnancy. The model compares various algorithm used for predicting the diabetes. Nowadays predicting the diabetes is quite challenging task to bring success in it. The application of gestational diabetes finder is to save the life of mother and fetus from disease. Predicting the early stage would be easier to treat the patient with medications and control with precautionaries, further controlling complications. It is concluded that random forest algorithm have an higher accuracy in predicting the **GDM** with an accuracy of 97%.

APPENDIX 1

```
# Importing essential libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.model_selection import cross_val_score
# Using GridSearchCV to find the best algorithm for this problem
from sklearn.model_selection import GridSearchCV
from sklearn.model_selection import ShuffleSplit
from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.svm import SVC
from sklearn import metrics
# Loading the dataset
df = pd.read_csv(‘diabetes.csv')
# Returns number of rows and columns of the dataset
```

```
print(df.shape)

# Returns an object with all of the column headers

print(df.columns)

# Returns different datatypes for each columns (float, int, string, bool, etc.)

print(df.dtypes)

# Returns the first x number of rows when head(num). Without a number it returns 5

print(df.head())

# Returns basic information on all columns

print(df.info())

# Returns basic information on all columns

print(df.info())

# Returns true for a column having null values, else false

print(df.isnull().any())

df = df.rename(columns={'DiabetesPedigreeFunction':'DPF'})

print(df.head())

# Plotting the Outcomes based on the number of dataset entries

plt.figure(figsize=(5,10))

sns.countplot(x='Outcome', data=df)

# Removing the unwanted spines

plt.gca().spines['top'].set_visible(False)
plt.gca().spines[‘right'].set_visible(False)

# Headings

plt.xlabel('Has Diabetes')

plt.ylabel('Count')



plt.show()

# Replacing the 0 values from ['Glucose','BloodPressure','SkinThickness','Insulin','BMI'] by NaN

df_copy = df.copy(deep=True)

df_copy[['Glucose','BloodPressure','SkinThickness','Insulin','BMI']] = df_copy[['Glucose','BloodPressure','SkinThickness','Insulin','BMI']].replace(0,np.NaN)

print(df_copy.isnull().sum())

df_copy

# To fill these Nan values the data distribution needs to be understood

# Plotting histogram of dataset before replacing NaN values

p = df_copy.hist(figsize = (15,15))

plt.show()

# Replacing NaN value by mean, median depending upon distribution

df_copy['Glucose'].fillna(df_copy['Glucose'].mean(), inplace=True)

df_copy['BloodPressure'].fillna(df_copy['BloodPressure'].mean(), inplace=True)
```

```
df_copy['SkinThickness'].fillna(df_copy['SkinThickness'].median(), inplace=True)

df_copy['Insulin'].fillna(df_copy['Insulin'].median(), inplace=True)

df_copy['BMI'].fillna(df_copy['BMI'].median(), inplace=True)# Replacing NaN value by mean, median depending upon distribution

df_copy['Glucose'].fillna(df_copy['Glucose'].mean(), inplace=True)

df_copy['BloodPressure'].fillna(df_copy['BloodPressure'].mean(), inplace=True)

df_copy['SkinThickness'].fillna(df_copy['SkinThickness'].median(), inplace=True)

df_copy['Insulin'].fillna(df_copy['Insulin'].median(), inplace=True)

df_copy['BMI'].fillna(df_copy['BMI'].median(), inplace=True)

 df_copy

# Plotting histogram of dataset after replacing NaN values

p = df_copy.hist(figsize=(15,15))

plt.show()

#sns.pairplot(df_copy,hue = 'Outcome')

print(df_copy.isnull().sum())

# Plotting histogram of dataset after replacing NaN values

p = df_copy.hist(figsize=(15,15))

plt.show()

#sns.pairplot(df_copy,hue = 'Outcome')

print(df_copy.isnull().sum())
```

```
df_copy.describe()

# Using GridSearchCV to find the best algorithm for this problem

from sklearn.model_selection import GridSearchCV

from sklearn.model_selection import ShuffleSplit

from sklearn.linear_model import LogisticRegression

from sklearn.tree import DecisionTreeClassifier

from sklearn.ensemble import RandomForestClassifier

from sklearn.svm import SVC

from sklearn import metrics

from sklearn.model_selection import train_test_split

X = df.drop(columns='Outcome')

y = df['Outcome']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.33,
random_state=7)

# Creating Random Forest Model

classifier = RandomForestClassifier(n_estimators=20, random_state=0)

classifier.fit(X_train, y_train)

# Creating a confusion matrix

from sklearn.metrics import confusion_matrix, classification_report, accuracy_score

y_pred = classifier.predict(X_test)
```

```
cm = confusion_matrix(y_test, y_pred)

print(cm)

# Plotting the confusion matrix

plt.figure(figsize=(10,7))

p = sns.heatmap(cm, annot=True, cmap="Blues", fmt='g')

plt.title('Confusion matrix for Random Forest Classifier Model - Test Set')

plt.xlabel('Predicted Values')

plt.ylabel('Actual Values')

plt.show()

# Accuracy Score

score = round(accuracy_score(y_test, y_pred),4)*100

print("Accuracy on test set: {}%".format(score))

# Classification Report

print(classification_report(y_test, y_pred))




# Creating a confusion matrix for training set

y_train_pred = classifier.predict(X_train)

cm = confusion_matrix(y_train, y_train_pred)

print(cm)
```

```
# Plotting the confusion matrix

plt.figure(figsize=(10,7))

p = sns.heatmap(cm, annot=True, cmap="Blues", fmt='g')

plt.title('Confusion matrix for Random Forest Classifier Model - Train Set')

plt.xlabel('Predicted Values')

plt.ylabel('Actual Values')

plt.show()

# Plotting the confusion matrix

plt.figure(figsize=(10,7))

p = sns.heatmap(cm, annot=True, cmap="Blues", fmt='g')

plt.title('Confusion matrix for Random Forest Classifier Model - Train Set')

plt.xlabel('Predicted Values')

plt.ylabel('Actual Values')

plt.show()



# Accuracy Score

score = round(accuracy_score(y_train, y_train_pred),4)*100

print("Accuracy on trainning set: {}%”.format(score))

# Classification Report

print(classification_report(y_train, y_train_pred))
```

```
import pickle

# Firstly we will be using the dump() function to save the model using pickle

saved_model = pickle.dumps(classifier)

# Then we will be loading that saved model

classifier_from_pickle = pickle.loads(saved_model)

# lastly, after loading that model we will use this to make predictions

classifier_from_pickle.predict(X_test)

df_copy.head()

p=classifier.predict([[1,250.0,85.0,39.0,190.0,90.0,.500,20]])

if p<0:

    print("great,you dont have diabetes")

else:

    print("you have diabetes, please consult a doctor”)

# Feature Scaling

from sklearn.preprocessing import StandardScaler

sc = StandardScaler()

X_train = sc.fit_transform(X_train)

X_test = sc.transform(X_test)

# Creating a function for prediction
```

```
def predict_diabetes(Pregnancies, Glucose, BloodPressure, SkinThickness, Insulin,
BMI, DPF, Age):
    preg = int(Pregnancies)
    glucose = float(Glucose)
    bp = float(BloodPressure)
    st = float(SkinThickness)
    insulin = float(Insulin)
    bmi = float(BMI)
    dpf = float(DPF)
    age = int(Age)


    x = [[preg, glucose, bp, st, insulin, bmi, dpf, age]]
    y = X_train


    return classifier.predict(y)
# Prediction 1
# Input sequence: Pregnancies, Glucose, BloodPressure, SkinThickness, Insulin,
BMI, DPF, Age
prediction = predict_diabetes(2, 200, 72, 15, 76, 70.1, 0.547, 95)[0]
if prediction:
  print('Oops! You have diabetes,please consult a doctor')
```

```
else:
  print("Great! You don't have diabetes.")


# Prediction 2
# Input sequence: Pregnancies, Glucose, BloodPressure, SkinThickness, Insulin,
BMI, DPF, Age
p=(5,90,93,35,122,39.2,0.200,30)[0];
prediction = predict_diabetes(4, 117, 88, 24, 145, 34.5, 0.403, 40)[0]
if prediction:
  print('Oops! You have diabetes,please consult a doctor')
else:
  print("Great! You don't have diabetes.")
# Prediction 3
# Input sequence: Pregnancies, Glucose, BloodPressure, SkinThickness, Insulin,
BMI, DPF, Age
p=(2,90,93,35,122,39.2,0.200,30)[0];


prediction = predict_diabetes(5, 138,62,35,0,33.6,0.127,47)[0]
if prediction:
  print('Oops! You have diabetes,please consult a doctor')
else:
  print("Great! You don't have diabetes.")
```

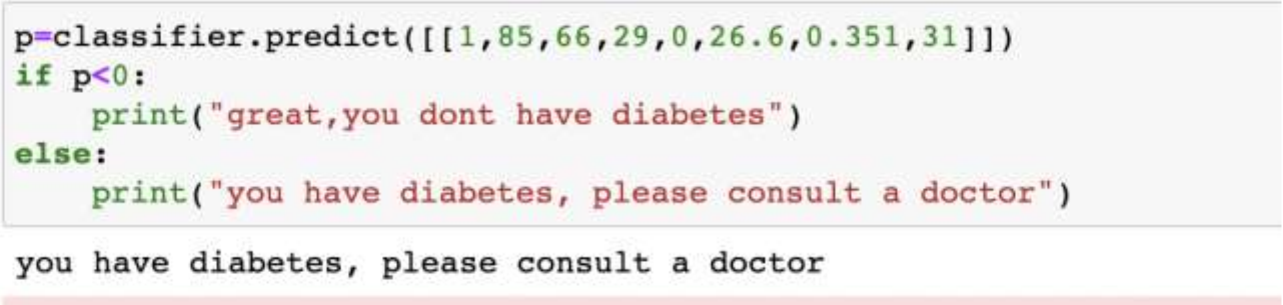
Logistic regression

```
import pandas as pd

import numpy as np

import sklearn

from sklearn.metrics import classification_report

from sklearn.ensemble import RandomForestClassifier

from sklearn.tree import DecisionTreeClassifier

from sklearn.neighbors import KNeighborsClassifier

from sklearn.svm import SVC

from sklearn.linear_model import LogisticRegression

# otherwise we're using at lease version 0.18

from sklearn.model_selection import train_test_split

dataset = pd.read_csv(“diabetes.csv",delimiter=',')

x=dataset

data = x.iloc[:,0:8].values

labels = x.iloc[:,8].values

data

data
```

```
labels

# construct the training and testing split by taking 75% of the data for training

# and 25% for testing

(trainData, testData, trainLabels, testLabels) = train_test_split(np.array(data),

          np.array(labels), test_size=0.25, random_state=42)

# initialize the model as a decision tree

#splitter = best, random , max_features = int, auto, log, none

#model = DecisionTreeClassifier(random_state=84,splitter='random',
max_features=8)

# Random Forest

#model = RandomForestClassifier(n_estimators=10,
random_state=42,max_features="auto")

#KNN

#weights = uniform, weights

#model = KNeighborsClassifier(n_neighbors=9)

#SVM

#kernel='rbf', linear, poly, C=1,gamma=0

#model = SVC(kernel="rbf",C=100)

#SVM

#kernel='rbf', linear, poly, C=1,gamma=0

#model = SVC(kernel="rbf",C=100)
```

```
#Logistic Regression

#penality = l1,l2, elasticnet

#solver = liblinear, sag, saga, lbfgs, newton-cg

#max-iter = 100

model = LogisticRegression(max_iter=15)#,solver = ‘sag')

#penalty=‘l1',solver="saga",max_iter=2000

# train the decision tree

print("[INFO] training model…")

model.fit(trainData, trainLabels)

# evaluate the classifier

print("[INFO] evaluating…")

predictions = model.predict(testData)

print(classification_report(testLabels, predictions))

from sklearn.metrics import confusion_matrix, classification_report, accuracy_score# Accuracy Score

score = round(accuracy_score(testLabels,predictions),4)*100

print("Accuracy on trainning set: {}%".format(score))# Accuracy Score

score = round(accuracy_score(testLabels,predictions),4)*100

print("Accuracy on trainning set: {}%".format(score))
```

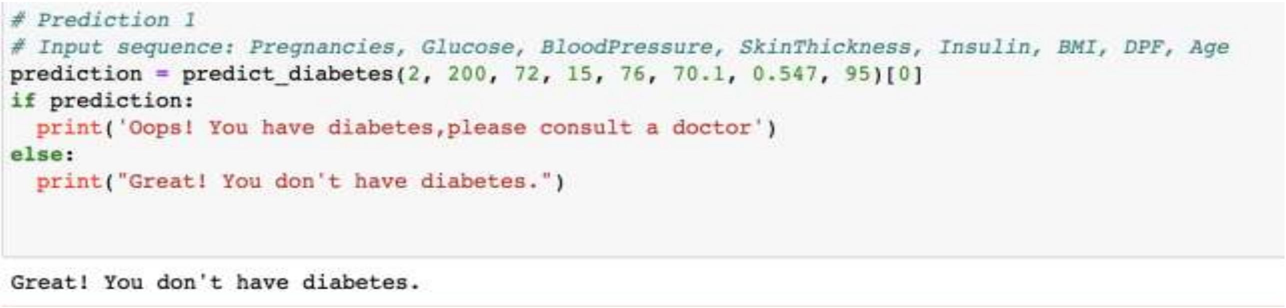
APPENDIX 2



```
p=classifier.predict([[1,85,66,29,0,26.6,0.351,31]])
if p<0:
    print("great,you dont have diabetes")
else:
    print("you have diabetes, please consult a doctor")
```

```
you have diabetes, please consult a doctor
```

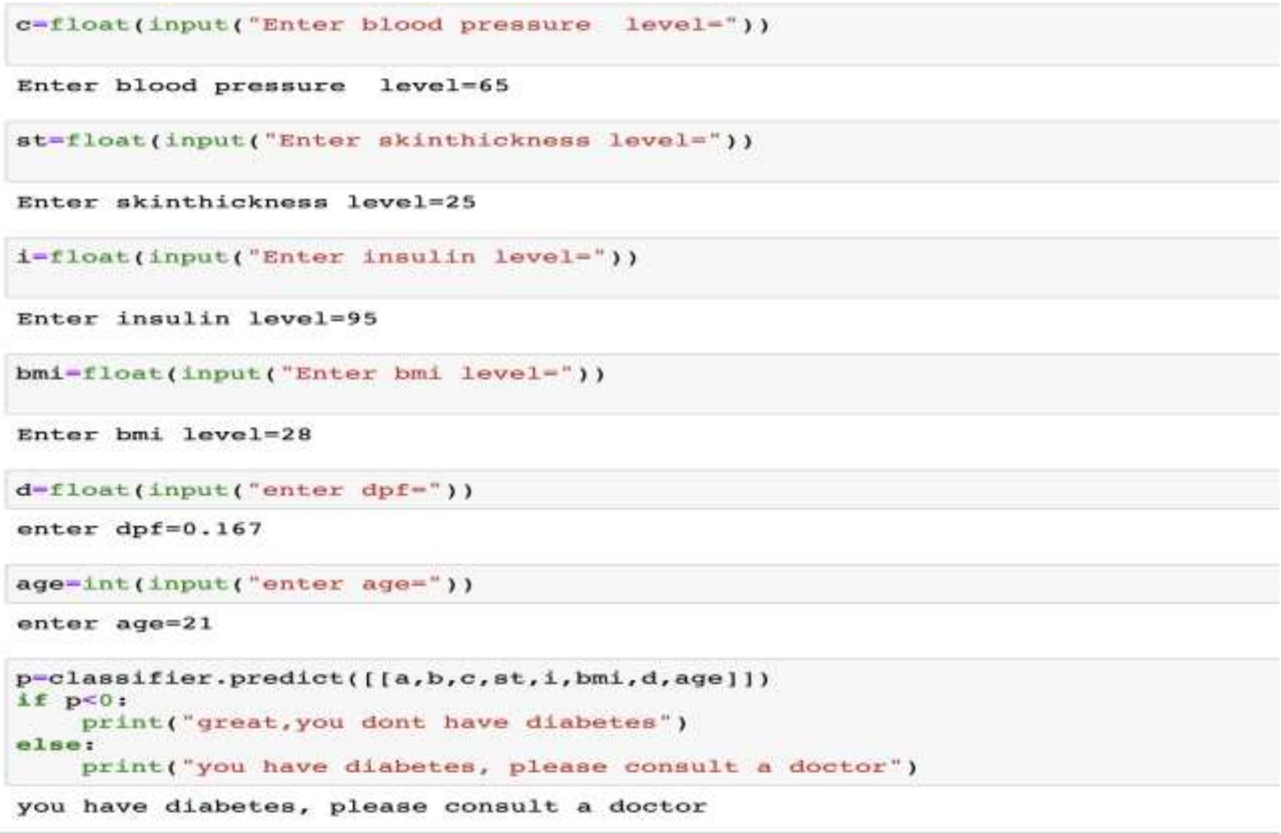
TEST CASE 1



```
# Prediction 1
# Input sequence: Pregnancies, Glucose, BloodPressure, SkinThickness, Insulin, BMI, DPF, Age
prediction = predict_diabetes(2, 200, 72, 15, 76, 70.1, 0.547, 95)[0]
if prediction:
  print('Oops! You have diabetes,please consult a doctor')
else:
  print("Great! You don't have diabetes.")
```

```
Great! You don't have diabetes.
```

TEST CASE 2



```
c=float(input("Enter blood pressure  level="))
```

```
Enter blood pressure  level=65
```

```
st=float(input("Enter skinthickness level="))
```

```
Enter skinthickness level=25
```

```
i=float(input("Enter insulin level="))
```

```
Enter insulin level=95
```

```
bmi=float(input("Enter bmi level="))
```

```
Enter bmi level=28
```

```
d=float(input("enter dpf="))
```

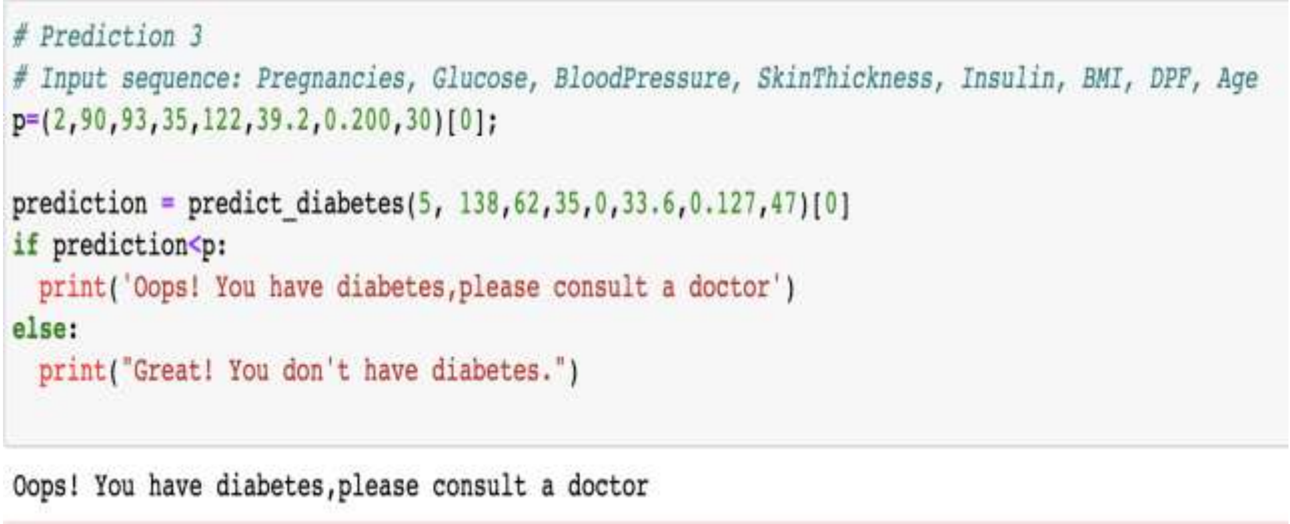
```
enter dpf=0.167
```

```
age=int(input("enter age="))
```

```
enter age=21
```

```
p=classifier.predict([[a,b,c,st,i,bmi,d,age]])
if p<0:
    print("great,you dont have diabetes")
else:
    print("you have diabetes, please consult a doctor")
```

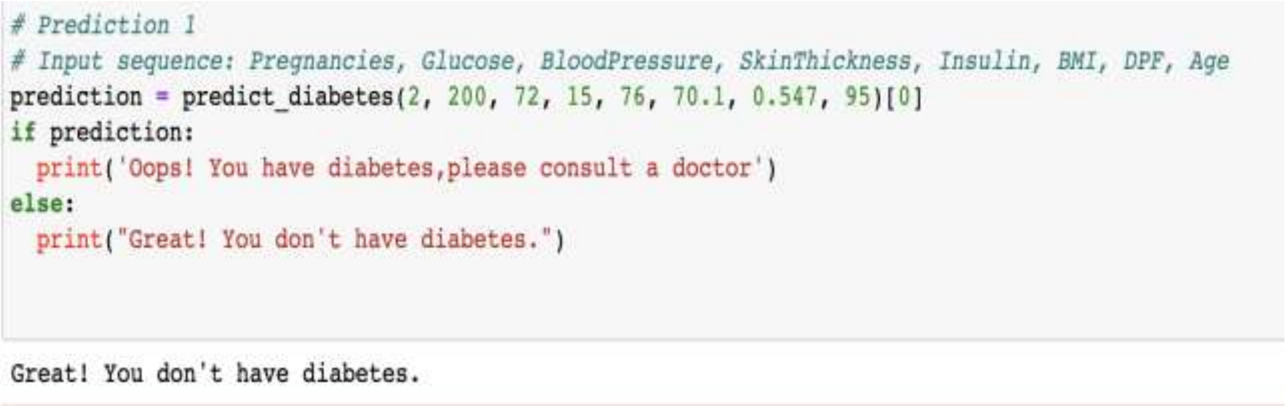
```
you have diabetes, please consult a doctor
```



```
# Prediction 3
# Input sequence: Pregnancies, Glucose, BloodPressure, SkinThickness, Insulin, BMI, DPF, Age
p=(2,90,93,35,122,39.2,0.200,30)[0];

prediction = predict_diabetes(5, 138,62,35,0,33.6,0.127,47)[0]
if prediction<p:
  print('Oops! You have diabetes,please consult a doctor')
else:
  print("Great! You don't have diabetes.")
```

```
Oops! You have diabetes,please consult a doctor
```

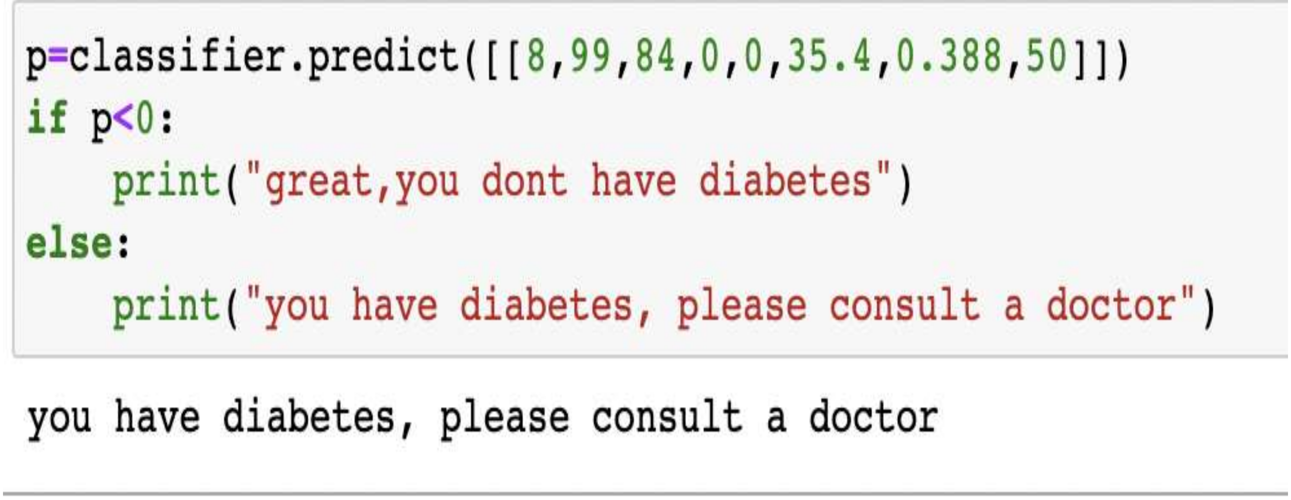
TEST CASE 3



```
# Prediction 1
# Input sequence: Pregnancies, Glucose, BloodPressure, SkinThickness, Insulin, BMI, DPF, Age
prediction = predict_diabetes(2, 200, 72, 15, 76, 70.1, 0.547, 95)[0]
if prediction:
  print('Oops! You have diabetes,please consult a doctor')
else:
  print("Great! You don't have diabetes.")
```

```
Great! You don't have diabetes.
```

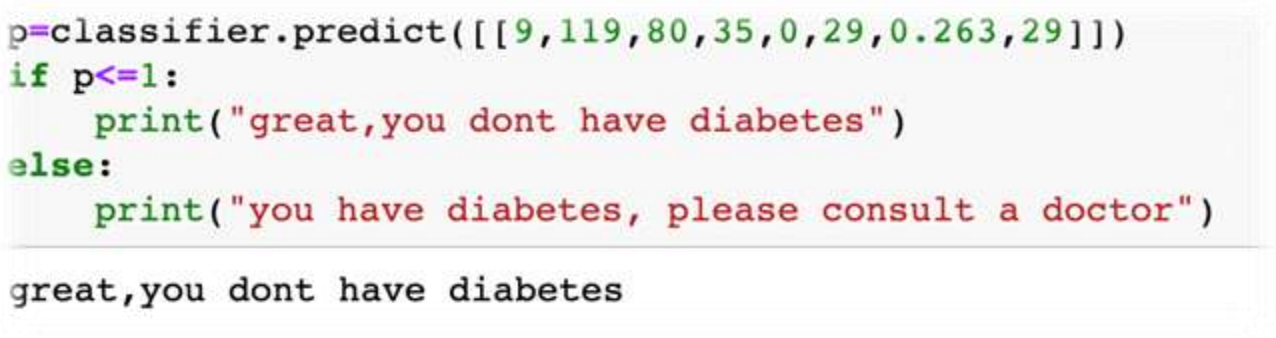
TEST CASE 4



```
p=classifier.predict([[8,99,84,0,0,35.4,0.388,50]])
if p<0:
    print("great,you dont have diabetes")
else:
    print("you have diabetes, please consult a doctor")
```

```
you have diabetes, please consult a doctor
```

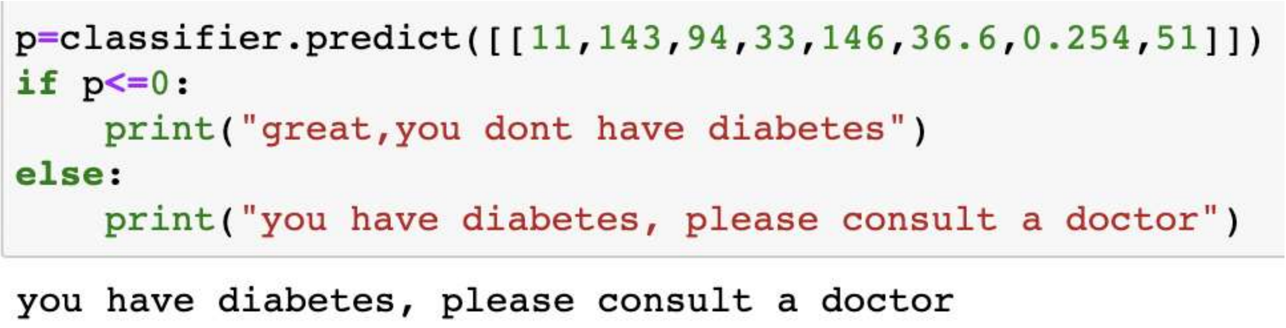
TEST CASE 5



```
p=classifier.predict([[9,119,80,35,0,29,0.263,29]])
if p<=1:
    print("great,you dont have diabetes")
else:
    print("you have diabetes, please consult a doctor")
```

```
great,you dont have diabetes
```

TEST CASE 6



```
p=classifier.predict([[11,143,94,33,146,36.6,0.254,51]])
if p<=0:
    print("great,you dont have diabetes")
else:
    print("you have diabetes, please consult a doctor")
```

```
you have diabetes, please consult a doctor
```

TEST CASE 7

REFERENCES

1. F. Du et al., "Prediction of pregnancy diabetes based on machine learning," BIBE 2019; The Third International Conference on Biological Information and Biomedical Engineering, 2019, pp. 1-6.

2. I. Gnanadass, "Prediction of Gestational Diabetes by Machine Learning Algorithms," in IEEE Potentials, vol. 39, no. 6, pp. 32-37, Nov.-Dec. 2020, doi: 10.1109/MPOT.2020.3015190.

3. V. Ganesh, J. Kolluri and K. V. Kumar, "Diabetes Prediction using Logistic Regression and Feature Normalization," 2021 International Conference on Innovative Computing, Intelligent Communication and Smart Electrical Systems (ICSES), 2021, pp. 1-6, doi: 10.1109/ICSES52305.2021.9633773.

4. R. S. Shankar, V. V. S. Raju, K. Murthy and D. Ravibabu, "Optimized Model for Predicting Gestational Diabetes using ML Techniques," 2021 5th International Conference on Electronics, Communication and Aerospace Technology (ICECA), 2021, pp. 1623-1629, doi: 10.1109/ICECA52323.2021.9676075.

5. Alfadhli EM. Gestational diabetes mellitus. Saudi Med J. 2015;36(4):399-406. doi:10.15537/smj.2015.4.10307

6. Alfadhli, Eman M. “Gestational diabetes mellitus.” Saudi medical journal vol. 36,4 (2015): 399-406. doi:10.15537/smj.2015.4.10307

7. Du, Y., Rafferty, A.R., McAuliffe, F.M. et al. An explainable machine learning-based clinical decision support system for prediction of gestational diabetes mellitus. Sci Rep 12, 1170 (2022). https://doi.org/10.1038/s41598-022-05112-2

8. Zhang Z, Yang L, Han W, Wu Y, Zhang L, Gao C, Jiang K, Liu Y, Wu H Machine Learning Prediction Models for Gestational Diabetes Mellitus: Meta-analysis JMed Internet Res 2022;24(3):e26634doi: 10.2196/26634

9. Jyotismita Chaki, S. Thillai Ganesh, S.K Cidham, S. Ananda Theertan, Machine learning and artificial intelligence based Diabetes Mellitus detection and self-management: A systematic review https://doi.org/10.1016/j.jksuci.2020.06.013

10. Aljumah, A.A., Ahamad, M.G., Siddiqui, M.K., 2013. Application of data mining: Diabetes health care in young and old patients. Journal of King Saud University - Computer and Information Sciences 25, 127–136.doi:10.1016/j.jksuci.2012.10.003.

11. Arora, R., Suman, 2012. Comparative Analysis of Classification Algorithms on Different Datasets using WEKA. International Journal of Computer Applications 54, 21–25. doi:10.5120/8626-2492.

12. Rani, A. S., & Jyothi, S. (2016, March). Performance analysis of classification algorithms under different datasets. In Computing for Sustainable Global Development (INDIACom), 2016 3rd International Conference on (pp. 1584-1589). IEEE.