



# **DIABETES PREDICTION SYSTEM USING MACHINE LEARNING WITH WEB APP**

## **A PROJECT REPORT**

*Submitted by*

**LOKHITHA D      (211419205101)**

**DURGA V      (211419205303)**

**DEVIPRIYA S      (211419205039)**

*in partial fulfillment for the award of the degree*

*of*

**BACHELOR OF TECHNOLOGY**

*in*

**INFORMATION TECHNOLOGY**

**PANIMALAR ENGINEERING COLLEGE, POONAMALLEE**

**ANNA UNIVERSITY : CHENNAI 600 025**

**APRIL 2023**

# **ANNA UNIVERSITY: CHENNAI 600 025**

## **BONAFIDE CERTIFICATE**

Certified that this project report“**DIABETES PREDICTION SYSTEM USING MACHINE LEARNING WITH WEB APP**” is the bonafide work of“**LOKHITHA D(211419205101),DURGA V (211419205303), DEVIPRIYA S (211419205039)**”who carried out the project under my supervision.

**Dr. M. HELDA MERCY M.E., Ph.D.,Mrs. J. HEMAVATHY M.Tech.,**

**HEAD OF THE DEPARTMENT**

**SUPERVISOR**

Department of Information Technology

Department of Information Technology

Panimalar Engineering College

Panimalar Engineering College

Poonamallee, Chennai - 600 123

Poonamallee, Chennai - 600 123

Submitted for the project and viva-voce examination held on \_\_\_\_\_

**INTERNAL EXAMINER**

**EXTERNAL EXAMINER**

## DECLARATION

I hereby declare that the project report entitled “**PROJECT TITLE**” which is being submitted in partial fulfilment of the requirement of the course leading to the award of the ‘Bachelor Of Technology in Information Technology ’ in **Panimalar Engineering College, an Autonomous institution Affiliated to Anna university- Chennai** is the result of the project carried out by me under the guidance of **Mrs. J. HEMAVATHY M.Tech.,in the Department of Information Technology**. I further declared that I or any other person has not previously submitted this project report to any other institution/university for any other degree/ diploma or any other person.

**LOKHITHA D (211419205101)**

**DURGA V (211419205303)**

**DEVI PRIYA S (211419205039)**

Date:

Place: Chennai

It is certified that this project has been prepared and submitted under my guidance.

Date: **Mrs.J. HEMAVATHY M.Tech.,**

Place: Chennai(**Assistant professor/IT**)

## ACKNOWLEDGEMENT

A project of this magnitude and nature requires kind co-operation and support from many, for successful completion . We wish to express our sincere thanks to all those who were involved in the completion of this project.

Our sincere thanks to **Our Beloved Secretary and Correspondent, Dr. P. CHINNADURAI, M.A., Ph.D.,**for his sincere endeavor in educating us in his premier institution.

We would like to express our deep gratitude to **Our Dynamic Directors , Mrs. C. VIJAYA RAJESHWARI and Dr. C. SAKTHI KUMAR, M.E.,M.B.A.,Ph.D.,andDR.SARANYASREESAKTHIKUMAR.,B.E.,M.B.A.,Ph.D.,**for providing us with the necessary facilities for completion of this project.

We also express our appreciation and gratefulness to**Our Principal Dr. K. MANI, M.E., Ph.D.,** who helped us in the completion of the project. We wish to convey our thanks and gratitude to our head of the department, **Dr. M. HELDA MERCY,M.E., Ph.D.,**Department of Information Technology, for her support and by providing us ample time to complete our project.

We express our indebtedness and gratitude to our Project coordinator**Mr. M. DILLI BABU, M.E.,(Ph.D.,)**AssociateProfessor, Department of Information Technology for his guidance throughout the course of our project.We also express sincere thanks to our supervisor**Mrs.J. HEMAVATHY M.Tech.,**for providing the support to carry out the project successfully. Last, we thank ourparents and friends for providing their extensive moral support and encouragement during the courseof the project.

## **TABLE OF CONTENTS**

<b>CHAPTER NO</b>	<b>TITLE</b>	<b>PAG.E NO</b>
	<b>ABSTRACT</b>	
	<b>LIST OF FIGURES</b>	
	<b>LIST OF SYMBOLS</b>	
<b>1</b>	<b>1.1INTRODUCTION</b>	<b>2</b>
	<b>1.2 OVERVIEW OF THE PROJECT</b>	<b>3</b>
	<b>1.3 NEED FOR THE PROJECT</b>	<b>3</b>
	<b>1.4 OBJECTIVE OF THE PROJECT</b>	<b>3</b>
	<b>1.5 SCOPE OF THE PROJECT</b>	<b>4</b>
<b>2</b>	<b>LITERATURE SURVEY</b>	
	<b>2.1 LITERATURE SURVEY</b>	
	<b>2.2 FEASIBILITY STUDY</b>	<b>9</b>
	<b>2.2.1 ECONOMICAL FEASIBILITY</b>	<b>9</b>
	<b>2.2.2 TECHNICAL FEASIBILITY</b>	<b>9</b>
	<b>2.2.3 SOCIAL FEASIBILITY</b>	<b>9</b>
<b>3</b>	<b>SYSTEM DESIGN</b>	<b>11</b>
	<b>3.1 EXISTING SYSTEM</b>	<b>11</b>
	<b>3.2 PROPOSED SYSTEM</b>	<b>12</b>
	<b>3.2.1UML DIAGRAMS</b>	<b>13</b>
	<b>3.2.1.1 E-R DIAGRAM</b>	<b>13</b>
	<b>3.2.1.2 DFD LEVEL 0 DIAGRAM</b>	<b>14</b>
	<b>3.2.1.3 DFD LEVEL 1 DIAGRAM</b>	<b>14</b>

	<b>3.2.1.4 USE CASE DIAGRAM</b>	<b>15</b>
	<b>3.2.1.5 ACTIVITY DIAGRAM</b>	<b>16</b>
	<b>3.2.1.6 CLASS DIAGRAM</b>	<b>17</b>
	<b>3.2.1.7 SEQUENCE DIAGRAM</b>	<b>18</b>
	<b>3.2.1.8 COLLOBORATION DIAGRAM</b>	<b>19</b>
	<b>3.3MODULE DESCRIPTION</b>	<b>19</b>
	<b>3.3.1. DATA COLLECTION AND PREPROCESSING</b>	<b>19</b>
	<b>3.3.2 FEATURE EXTRACTION</b>	<b>20</b>
	<b>ALGORITHMS USED IN THE PROPOSED SYSTEM</b>	<b>20</b>
	<b>3.3.1.1 K-NEIGHBOURS NEAREST ALGORITHM</b>	<b>20</b>
	<b>3.3.1.2 RANDOM FOREST ALGORITHM</b>	<b>21</b>
	<b>3.3.1.3 LIGHTBGM ALGORITHM</b>	<b>22</b>
	<b>3.3.1.4 XG BOOST ALGORITHM</b>	<b>23</b>
	<b>3.3.3 MODEL CREATION</b>	<b>24</b>
	<b>3.3.4 TRAINING AND TESTING</b>	<b>24</b>
	<b>3.3..5 PREDICTION PROCESS</b>	<b>24</b>
	<b>3.4 PROPOSED SYSTEM ALGORITHM</b>	<b>25</b>
	<b>3.4.1 XG BOOST ALGORITHM</b>	<b>25</b>
	<b>3.4.1.1 THE MATHEMATICS OF XG BOOST ALGORITHM</b>	<b>25</b>
	<b>3.4.1.2ADVANTAGES OF XG BOOST ALGORITHM</b>	<b>27</b>
<b>4</b>	<b>REQUIREMENT SPECIFICATION</b>	<b>30</b>
	<b>4.2 SOFTWARE REQUIREMENTS</b>	<b>30</b>

	4.2.1 PYTHON	30
	4.2.2 ANACONDA	32
	4.3 HARDWARE REQUIREMENTS	32
	4.3.1 MICROSOFT SERVER ENABLED COMPUTERS	32
	4.3.2 4GB RAM	32
	4.3.3 1.5GHZ PROCESSOR	32
5	5.1 IMPLEMENTATION	35
	5.2 WORKING	35
	5.3 CODE	35
	5.3.1 INDEX.HTML	35
	5.3.2 STYLE.CSS	38
	5.3.3 APP.PY	42
	5.4 SAMPLE OUTPUT	44
6	TESTING AND MAINTENANCE	46
	6.1 UNIT TESTING	46
	6.2 INTEGRATION TESTING	46
	6.3 FUNCTIONAL TESTING	46
	6.4 SYSTEM TESTING	47
	6.5 WHITE BOX TESTING	47
	6.6 BLACK BOX TESTING	47
7	OUTPUT AND DISCUSSION	49
	7.1 PREDICTION	49
	7.2 PROPOSED SYSTEM IMPLEMENTATION	49
	7.3 RESULT	50

	<b>7.4 CONCLUSION</b>	<b>53</b>
<b>8</b>	<b>REFERENCE</b>	



## ABSTRACT

In the medical field, it is crucial to predict diseases in advance to help them. Diabetes is one of the most dangerous health conditions in the world. In an ultra-modern culture, sugar and fat are ubiquitous in our healthy habits, increasing the threat of diabetes. To predict complaints, it is important to understand their symptoms. Currently, machine learning (ML) algorithms are invaluable for complaint discovery. This compilation presents a model for predicting diabetes using a fusion machine literacy approach. The abstract framework consists of two classes of models: support vector machine (SVM) models and artificial neural network (ANN) models. These models dissect the data set to determine whether opinions about diabetes are positive or negative.

The dataset used in this exploration is independently divided into training data and test data, with a ratio of 7030. The transactions of these models become the input class functions of the fuzzy models, and the feeling of fuzziness ultimately determines whether the perception of diabetes is positive or negative. The cloud storage system stores the fusion model for future use. Based on the real-time medical records of the cases, the fusion model predicts whether the cases are diabetic or not. The proposed ensemble ML model achieves a prediction accuracy of 92.7%, which is more advanced than the originally published method.

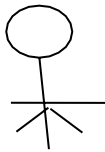
Diabetes mellitus, commonly known as diabetes, is a metabolic disease in which diabetics experience blood sugar problems due to irregular production and release of insulin. It is also a long-term condition characterized by high blood sugar. It is one of the most serious diseases in the world and can have multiple consequences. Based on recent increases in incidence, the number of diabetes cases worldwide will reach 642 million by 2040, suggesting that one in 10 people will become ill.




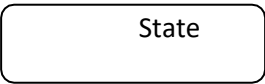
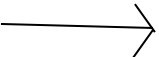
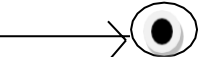
## **LIST OF FIGURES**

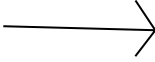
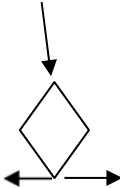
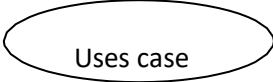
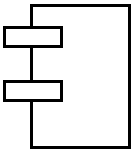
<b>FIGURE NO</b>	<b>NAME OF THE FIGURE</b>	<b>PAGE NO</b>
<b>3.1</b>	<b>PROPOSED SYSTEM ARCHITECTURE</b>	<b>12</b>
<b>3.2</b>	<b>E-R DIAGRAM</b>	<b>13</b>
<b>3.3</b>	<b>DFD LEVEL 0</b>	<b>14</b>
<b>3.4</b>	<b>DFD LEVEL 1</b>	<b>14</b>
<b>3.5</b>	<b>USECASE DIAGRAM</b>	<b>15</b>
<b>3.6</b>	<b>ACTIVITY DIAGRAM</b>	<b>16</b>
<b>3.7</b>	<b>CLASS DIAGRAM</b>	<b>17</b>
<b>3.8</b>	<b>SEQUENCE DIAGRAM</b>	<b>18</b>
<b>3.9</b>	<b>COLLOBORATION DIAGRAM</b>	<b>19</b>
<b>3.10</b>	<b>DATASET USED FOR PROPOSED DIABESTES PREDICTION</b>	<b>20</b>
<b>3.11</b>	<b>KNN ACCURACY GRAPH</b>	<b>21</b>
<b>3.12</b>	<b>RANDOM FOREST ACCURACY GRAPH</b>	<b>22</b>
<b>3.13</b>	<b>COMPARISON OF ALL 4 ALGORITHMS USED IN THE TRAINING DATASET</b>	<b>23</b>
<b>4.1</b>	<b>SAMPLE OUTPUT</b>	<b>44</b>
<b>7.1</b>	<b>RUNNING THE PYTHON FILE</b>	<b>49</b>
<b>7.2</b>	<b>COPYING THE PORT NUMBER FROM THE OUTPUT OF PYTHON FILE EXECUTED</b>	<b>50</b>
<b>7.3</b>	<b>OUTPUT SCREEN 1</b>	<b>51</b>

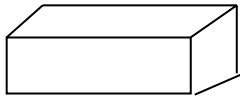
<b>7.4</b>	<b>OUTPUT SCREEN 2</b>	<b>51</b>
<b>7.5</b>	<b>OUTPUT SCREEN 3</b>	<b>52</b>
<b>7.6</b>	<b>OUTPUT SCREEN 4</b>	<b>53</b>

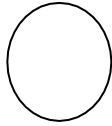


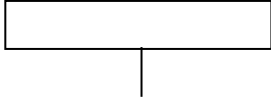
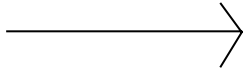
## LIST OF SYMBOLS

S.NO	NOTATION NAME	NOTATION	DESCRIPTION
1.	Class	<div style="text-align: right; margin-right: 50px;"><i>Class Name</i></div> <div style="display: flex; justify-content: space-between;"> <div> <i>+public</i>  <i>-private</i>  <i>attribute#protected</i> </div> <div> <i>-attribute</i>  -    <i>+operation</i>  <i>+operation</i> </div> </div>	Represents a collection of similar entities grouped together.
2.	Association	<div style="text-align: center; margin-bottom: 10px;">NAME</div> <div style="display: flex; justify-content: center; align-items: center;"> <div style="border: 1px solid black; padding: 2px 10px; margin-right: 10px;">ClassA</div> <div style="border-top: 1px solid black; width: 30px; height: 1px; margin: 0 5px;"></div> <div style="border: 1px solid black; padding: 2px 10px; margin-left: 10px;">ClassB</div> </div> <div style="display: flex; justify-content: center; align-items: center; margin-top: 10px;"> <div style="border: 1px solid black; padding: 2px 10px; margin-right: 10px;">ClassA</div> <div style="border-top: 1px solid black; width: 30px; height: 1px; margin: 0 5px;"></div> <div style="border: 1px solid black; padding: 2px 10px; margin-left: 10px;">ClassB</div> </div>	Associations represent static relationships between classes. Roles represent the way the two classes see each other.
3.	Actor		It aggregates several classes into a single class.
4.	Aggregation	<div style="display: flex; justify-content: space-around; align-items: center;"> <div style="border: 1px solid black; padding: 2px 10px; margin-right: 20px;">ClassA</div> <div style="border: 1px solid black; padding: 2px 10px; margin-right: 20px;">ClassA</div> </div> <div style="display: flex; justify-content: space-around; align-items: center;"> <div style="margin-top: 10px;">ClassB</div> <div style="margin-top: 10px;">ClassB</div> </div>	Interaction between the system and external environment

5.	Relation(uses)		Used for additional process communication.
6.	Relation(extends)		Extends relationship is used when one use case is similar to another use case but does a bit more.
7.	Communication		Communication between various use cases.
8.	State		State of the process.
9.	InitialState		Initial state of the object
10.	Finalstate		Final state of the object

11.	Controlflow		Represents various control flow between the states.
12.	Decisionbox		Represents decision making process from a constraint
13.	Usecase		Interaction between the system and external environment.
14.	Component		Represents physical modules which is a collection of components.

15.	Node		Represents physical modules which are a collection of components
-----	------	--	--

16.	Data Process/ State		A circle in DFD represents a state or process which has been triggered due to some event or action.
17.	External entity		Represents external entities such as keyboard, sensor or etc.
18.	Transition		Represents communication that occurs between processes.
19.	Object Lifeline		Represents the vertical dimension that the object communicates.
20.	Message	 Message	Represents the message exchanged.





# **CHAPTER 1**

## **INTRODUCTION**

## 1.1 INTRODUCTION

Diabetes mellitus, commonly known as diabetes, is a metabolic disease in which diabetics experience blood sugar problems due to irregular production and release of insulin. It is also a long-term condition characterized by high blood sugar. It is one of the most serious diseases in the world and can have multiple consequences. Based on recent increases in incidence, the number of diabetes cases worldwide will reach 642 million by 2040, suggesting that one in 10 people will become ill. This daunting figure, no doubt, demands immediate attention. Type 1 and type 2 diabetes are the two most common forms. Type 1 diabetes can affect anyone at any age, but it most often affects teenagers and children. People with type 1 diabetes whose body produces little or no insulin and who receive daily insulin injections to control their blood sugar. Type 2 diabetes can strike anyone at any age, but is much more common in adults, accounting for over 90% of all diabetes cases. In type 2 diabetes, the body does not make good use of the insulin it produces. The basis of type 2 diabetes care is a healthy lifestyle, including increased physical activity and a balanced diet. On the other hand, adults with type 2 diabetes will eventually need medication or insulin to control their blood sugar. Another type of diabetes is gestational diabetes, which is characterized by high blood sugar levels during pregnancy and is associated with complications for both mother and child. Their children were more likely to develop type 2 diabetes later in life, although this decreased after pregnancy.

Machine learning algorithms are a great way to analyze large amounts of data and make recommendations based on that data. These algorithms are useful for studying data sets and predicting new input values. Various experimenters use machine learning algorithms to predict and control many conditions. Machine learning algorithms are particularly good at predicting color conditions. Apply machine learning algorithms to check their diabetes prediction range to take necessary action to avoid diabetes. Many experimenters use machine learning algorithms to predict and control color conditions.

The designed thing is whether a case is diabetic or not based on the diabetes dataset. From the medical biography, eight characteristics including number of pregnancies, insulin levels, glucose levels, blood pressure, skin firmness, body mass index (BMI), body function diabetic childbirth and age, could predict whether a patient was diabetic. These features are integrated with the XGBoost algorithm for diabetes prediction, which can provide performance as reliable as diagnosing diabetes. These estimates are based on symptoms seen in the early stages of diabetes and some physical illnesses.

## **1.2 OVERVIEW OF THE PROJECT**

Diabetes mellitus, commonly known as diabetes, is a metabolic disease in which diabetics experience blood sugar problems due to irregular production and release of insulin. It is also a long-term condition characterized by high blood sugar. It is one of the most serious diseases in the world and can have multiple consequences. Based on recent increases in incidence, the number of diabetes cases worldwide will reach 642 million by 2040, suggesting that one in 10 people will become ill.

## **1.3 NEED FOR THE PROJECT**

The designed thing is whether a case is diabetic or not based on the diabetes dataset. From the medical biography, eight characteristics including number of pregnancies, insulin levels, glucose levels, blood pressure, skin firmness, body mass index (BMI), body function diabetic childbirth and age, could predict whether a patient was diabetic. These features are integrated with the XGBoost algorithm for diabetes prediction, which can provide performance as reliable as diagnosing diabetes. These estimates are based on symptoms seen in the early stages of diabetes and some physical illnesses.

## **1.4 OBJECTIVE OF THE PROJECT**

This daunting figure, no doubt, demands immediate attention. Type 1 and type 2 diabetes are the two most common forms. Type 1 diabetes can affect anyone at any age, but it most often affects teenagers and children. People with type1 diabetes whose body produces little or no insulin and who receive daily insulin injections to control their blood sugar. Type 2 diabetes can strike anyone at any age, but is much more common in adults, accounting for over 90% of all diabetes cases.

In type 2 diabetes, the body does not make good use of the insulin it produces. The basis of type 2 diabetes care is a healthy lifestyle, including increased physical activity and a balanced diet. On the other hand, adults with type 2 diabetes will eventually need medication or insulin to control their blood sugar. Another type of diabetes is gestational diabetes, which is characterized by high blood sugar levels during pregnancy and is associated with complications for both mother and child. Their children were more likely to develop type 2 diabetes later in life, although this decreased after pregnancy.

## 1.5 SCOPE OF THE PROJECT

Machine learning algorithms are a great way to analyze large amounts of data and make recommendations based on that data. These algorithms are useful for studying data sets and predicting new input values. Various experimenters use machine learning algorithms to predict and control many conditions. Machine learning algorithms are particularly good at predicting color conditions. `

Apply machine learning algorithms to check their diabetes prediction range to take necessary action to avoid diabetes. Many experimenters use machine learning algorithms to predict and control color conditions.

The designed thing is whether a case is diabetic or not based on the diabetes dataset. From the medical biography, eight characteristics including number of pregnancies, insulin levels, glucose levels, blood pressure, skin firmness, body mass index (BMI), body function diabetic childbirth and age, could predict whether a patient was diabetic. These features are integrated with the XGBoost algorithm for diabetes prediction, which can provide performance as reliable as diagnosing diabetes. These estimates are based on symptoms seen in the early stages of diabetes and some physical illnesses.

These features are integrated with the XGBoost algorithm for diabetes prediction, which can provide performance as reliable as diagnosing diabetes. These estimates are based on symptoms seen in the early stages of diabetes and some physical illnesses.

# **CHAPTER 2**

## **LITERATURE SURVEY**

## **LITERATURE SURVEY**

### **2.1. PROJECT TITLE: PREDICTION OF DIABETES EMPOWERED WITH FUSED MACHINE LEARNING**

**AUTHOR:** USAMA AHMED<sup>1,2</sup>, GHASSAN F. ISSA<sup>3</sup> , MUHAMMAD ADNAN KHAN <sup>1,4</sup> , SHABIB AFTAB <sup>2,5</sup>, (Member, IEEE), MUHAMMAD FARHAN KHAN<sup>6</sup> , RAED A. T. SAID<sup>7</sup> , TAHER M. GHAZAL <sup>3,8</sup>, (Member, IEEE), AND MUNIR AHMAD <sup>5</sup> , (Member, IEEE)

**YEAR:** 2022

**OBJECTIVE:** The aim of this project is to use dataset in UCI Machine Learning repository by using Machine Learning Algorithm such as Decision tree, Naive bayes constructed a model to predict diabetic patients. A cloud storage system stores the fused models for future use. Based on the patient's real-time medical record, the fused model predicts whether the patient is diabetic or not. The conceptual framework consists of two types of models: Support Vector Machine (SVM) and Artificial Neural Network (ANN) models.

**PROS:** System was trained to quickly identify if a person is diabetic

**CONS:** Lots of resources were used on process

### **2.2. PROJECT TITLE: ARTIFICIAL INTELLIGENCE FOR DIABETIC RETINOPATHY SCREENING, PREDICTION AND MANAGEMENT.**

**AUTHOR:** Gunasekeran, Dinesh V.<sup>a,b</sup>; Ting, Daniel S.W.<sup>a,c</sup>; Tan, Gavin S.W.<sup>a,c</sup>; Wong, Tien Y.<sup>a,c</sup>

**YEAR:** 2020

**OBJECTIVE:** This paper states that the Digital health solutions such as artificial intelligence and telehealth can facilitate the integration of community, primary and specialist eye care services, optimize the flow of patients within healthcare networks, and improve the efficiency of diabetic retinopathy management.

**PROS:** Supervised ML algorithms used to perform analysis and gave good outcomes

**CONS:** Unsupervised algorithms were totally ignored

**2.3. PROJECT TITLE:** AN ANALYTICAL PREDICTIVE MODELS AND SECURE WEB-BASED PERSONALIZED DIABETES MONITORING SYSTEM.

**AUTHOR:** ANDRÉS ANAYA-ISAZA 1,2 AND MATHA ZEQUERA-DIAZ 1 , (Member, IEEE)

**YEAR:**2022

**OBJECTIVE:** This paper states that the alternative and automated methods are necessary to detect DM, allowing it to take the pertinent measures in its treatment and avoid critical complications, such as the diabetic foot. On the other hand, foot thermography is a promising tool that allows visualization of thermal patterns, patterns that are altered as a consequence of shear and friction associated with lower limb neuropathy. Based on these considerations, we explored different strategies to detect patients with DM from foot thermography in this research.

**PROS:** Only random forest gave good accuracy out of all algorithms

**CONS:** SVM went with a comparable of 81.4% accuracy below Random forest

**2.4. PROJECT TITLE:** AN ANALYTICAL PREDICTIVE MODELS AND SECURE WEB-BASED PERSONALIZED DIABETES MONITORING SYSTEM.

**AUTHOR:** RADWA MARZOUK 1 , ALA SALEH ALLUHAIDAN 2 , AND SAHAR A. EL\_RAHMAN 3 , (Senior Member, IEEE)

**YEAR:** 2022

**OBJECTIVE:** . The proposed system can help doctors to make data-driven decisions and enhance patients' treatment. Several machine learning algorithms that are Decision Tree, Support Vector Classifier, Random Forest, Gradient Boosting, Multi-layer Perceptron's, Artificial Neural Network, k-Nearest Neighbors, Logistic Regression, and Naive Bayes are used. The proposed analytical model is evaluated based on two different datasets a synthetic dataset and PIMA Diabetes Dataset. The performance of the classification models was analyzed in terms of accuracy, recall, and precision based on the cross-validation strategy.

**PROS:** Models were built to test if a person is diabetic or not quickly

**CONS:** Accuracy was a big problem, less models were only made

**2.5. PROJECT TITLE:** AN EFFICIENT PROBABILISTIC ENSEMBLE CLASSIFICATION ALGORITHM FOR DIABETES HANDLING CLASS IMBALANCE MISSING VALUES.

**AUTHOR:** LIYAN JIA 1 , ZHIPING WANG 1 , SIQI LV 1 , AND ZHAOHUI XU 2

**YEAR:**2022

**OBJECTIVE:** The aim of this paper is to evaluate the PE\_DIM model, the experiment equally considered two diabetes datasets, RSMH and Tabriz, to demonstrate the generality of the method in diabetes prediction. Additionally, in terms of area under the receiver operating characteristic curve metric uses several statistical tests to measure the performance of different classification methods. The ultimate results demonstrate that the average rank of this method is ranked first after 5-fold cross-validation, which is significantly different from the basic classifiers.

**PROS:** Few algorithms gave good results with parameters mentioned for diabetes

**CONS:** Other algorithms lacked way back on accuracy

**Our model was trained with parameters like glucose, insulin ,age etc to provide better outcomes**

**2.6. PROJECT TITLE:** DIABETIC FOOT ULCER ISCHEMIA AND INFECTION CLASSIFICATION USING EFFICIENTNET DEEP LEARNING MODELS.

**AUTHOR:** Ziyang Liu , Josvin John , and Emmanuel Agu

**YEAR:**2022

**OBJECTIVE:** : The EfficientNets model achieved 99% accuracy in ischemia classification and 98% in infection classification, outperforming ResNet and Inception (87% accuracy) and Ensemble CNN, the prior state of the art (Classification accuracy of 90% for ischemia 73% for infection). EfficientNets also classified test images in a fraction (10% to 50%) of the time taken by baseline models.

**PROS:** Random forest out passed all families of algorithm

**CONS:** Different families of algorithms were tested and failed



**2.7. PROJECT TITLE:** Intelligent Machine Learning approach for effective recognition of diabetes in E-Healthcare using clinical data.

**AUTHOR:** by Amin Ul Haq <sup>1,\*</sup>, Jian Ping Li <sup>1</sup>, Jalaluddin Khan <sup>1</sup>, Muhammad Hammad Memon <sup>1</sup>, Shah Nazir <sup>2</sup>, Sultan Ahmad <sup>3</sup>, Ghufraan Ahmad Khan <sup>4</sup> and mjad Ali <sup>5</sup>

**YEAR:** 2020

**OBJECTIVE:** The proposed method has been tested on the diabetes data set which is a clinical dataset designed from patient's clinical history. Further, model validation methods, such as hold out, K-fold, leave one subject out and performance evaluation metrics, includes accuracy, specificity, sensitivity, F1-score, receiver operating characteristic curve, and execution time have been used to check the validity of the proposed system. We have proposed a filter method based on the Decision Tree (Iterative Dichotomiser 3) algorithm for highly important feature selection. Two ensemble learning algorithms, Ada Boost and Random Forest, are also used for feature selection and we also compared the classifier performance with wrapper based feature selection algorithms.

**PROS:** Our model was trained with parameters like glucose, insulin ,age etc to provide better outcomes

**CONS:** Only one algorithm gave good results with parameters mentioned

## **2.9 FEASIBILITY STUDY**

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

### **2.9.1 ECONOMICAL FEASIBILITY**

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

### **2.9.2 TECHNICAL FEASIBILITY**

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

### **2.9.3 SOCIAL FEASIBILITY**

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

# **CHAPTER 3**

## **SYSTEM DESIGN**

## **SYSTEM DESIGN**

Health issues are increasing tremendously these days. Many people start going for complete medical check-ups during their late 40's or 50's of age. But our lifestyle has a great impact on health causing diabetic and other health problems. Early detection of diabetic can prevent death rates. As most of our health care industries are aiming at diagnosing these diseases in early stages, by using machine learning technique, we can detect disease at an early stage and help to cure it completely at zero cost. By using a proper dataset, a trained ML model is created which can diagnose a normal person and generate an output report which shows whether the person is affected by diabetic or not and also classify the diabetic level.

### **3.1 EXISTING SYSTEM**

The project asked several researchers to examine it in the field of diabetes using machine learning techniques. Extract knowledge from existing medical data. This predictive analytics model uses support vector machine algorithms, including logistic regression, naive Bayes, and support vector machines, where support vector machines provide better performing algorithms. In this work, we examine real diagnostic medical data against many risk factors using popular machine learning classification techniques to assess their predictive performance. The proposed fuzzy decision system achieves an accuracy rate of 94%.87, superior to other existing systems.

#### **3.1.1 DISADVANTAGES**

- i) Diabetes is one of the deadliest diseases in this world and it is increasing rapidly. It is not just a disease but the creator of different types of diseases.
- ii) We chose these algorithms for this study after some initial experimentation where we found that these techniques are more efficient for this problem.

### 3.2 PROPOSED SYSTEM:

This project proposes a fusion model for the prediction of diabetes. The proposed model has two main steps. Our first step is to form layers, while the second step is to test layers. However, in our proposed model, we only use two commonly used ML algorithms. If our proposed model does not meet the learning requirements, it will be recycled. The second stage of the proposed framework is reflected by the test layer.

The test layer gets the dataset from the medical database and loads the preprocessed training model from the cloud. In our proposed framework, the validation layer is related to the real-time diagnosis and classification of diabetes. The proposed ML fusion model can use real-time patient data as input and improve disease detection systems.

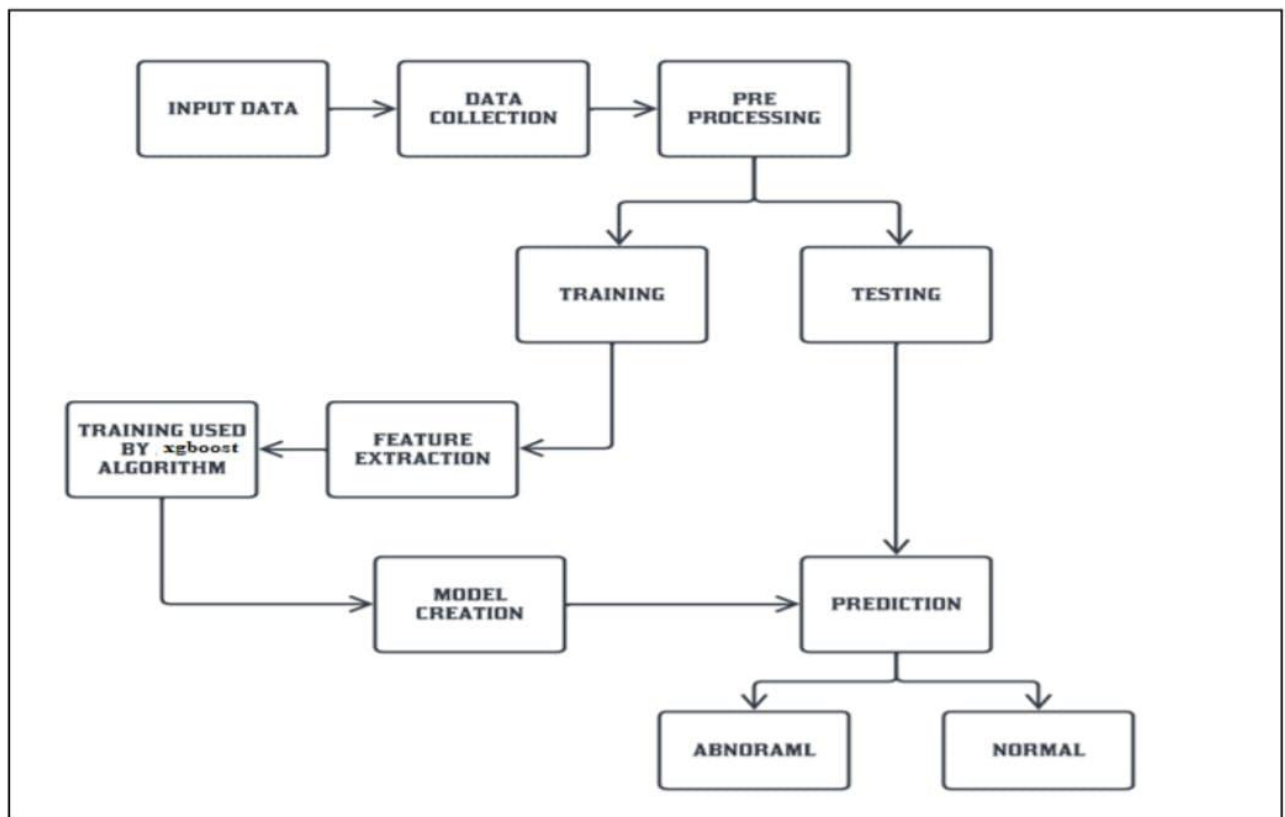


Fig 3.1 PROPOSED SYSTEM ARCHITECTURE OF DIABETES PREDICTION SYSSYEM

The proposed model has two main steps. Our first step is to form layers, while the second step is to test layers. However, in our proposed model, we only use two commonly used ML algorithms. If our proposed model does not meet the learning requirements, it will be recycled. The second stage of the proposed framework is reflected by the test layer. The test layer gets the dataset from the medical database and loads the preprocessed training model from the cloud. In our proposed framework, the validation layer is related to the real-time diagnosis and classification of diabetes. The proposed ML fusion model can use real-time patient data as input and improve disease detection systems.

The above block diagram is the flow of our proposed diabetics prediction system. The input data is collected from the source and go under preprocessing. And preprocessed data is first trained using the machine learning algorithm and the features are extracted and trained by the highest accuracy algorithm that is XGBoost. Then the created model is tested and compared using the machine learning algorithm. And finally the prediction is done and the output will be predicted as 0's and 1's.

### 3.2.1 UML DIAGRAM

UML, which stands for Unified Modeling Language, is a way to visually represent the architecture, design, and implementation of complex software systems.

#### 3.2.1.1 E-R DIAGRAMS

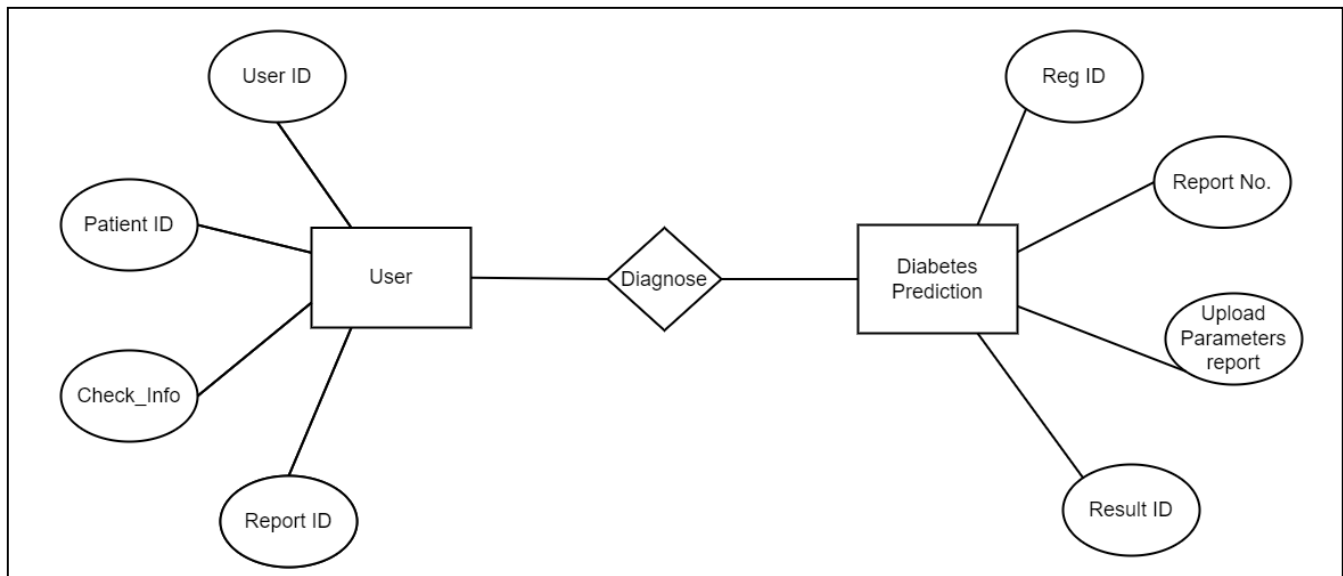


Fig 3.3 E-R DIAGRAM

An Entity Relationship (ER) Diagram is a type of flowchart that illustrates how “entities” such as people, objects or concepts relate to each other within a system.

### 3.2.1.2 DFD LEVEL 0 DIAGRAM

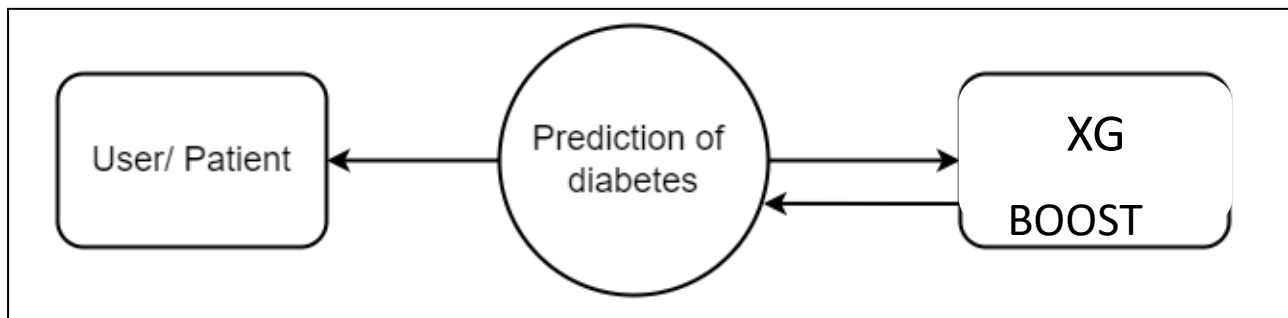


Fig 3.2 DFD LEVEL 0 DIAGRAM

DFD Level 0 is also called a Context Diagram. It's a basic overview of the whole system or process being analyzed or modeled. It's designed to be an at-a-glance view, showing the system as a single high-level process, with its relationship to external entities.

### 3.2.1.3 DFD LEVEL 1 DIAGRAM

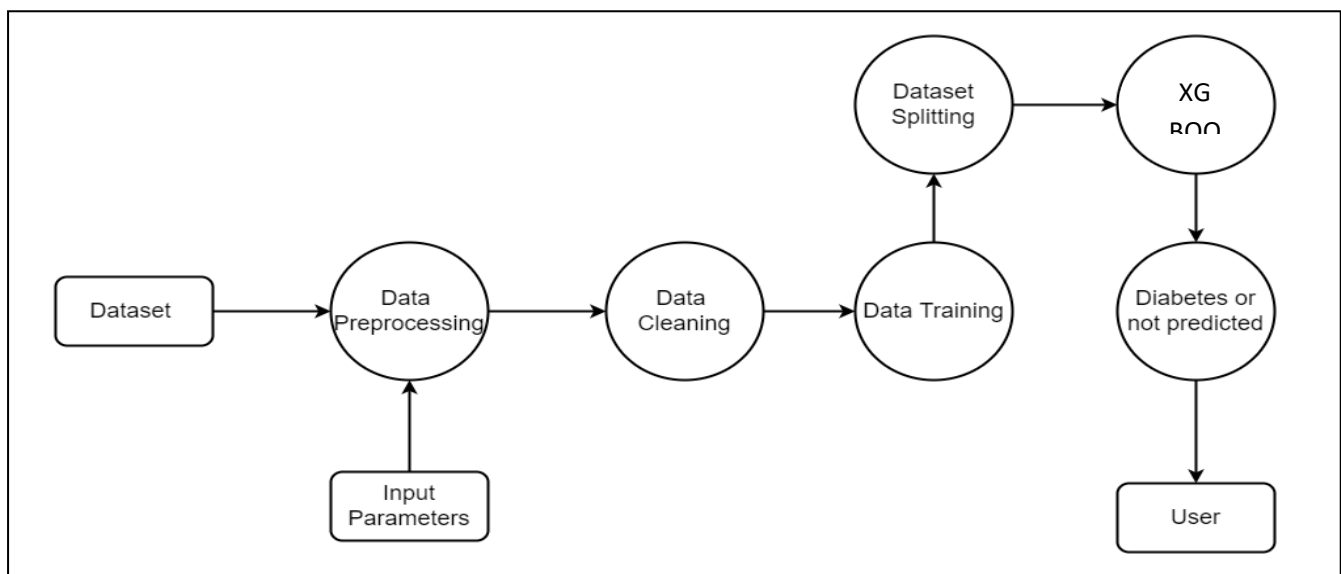


Fig 3.4 DFD LEVEL 1 DIAGRAM

Level 1 DFDs are still a general overview, but they go into more detail than a context diagram. In level 1 DFD, the single process node from the context diagram is broken down into sub-processes. As these processes are added, the diagram will need additional data flows and data stores to link them together.

#### 3.2.1.4 USECASE DIAGRAM

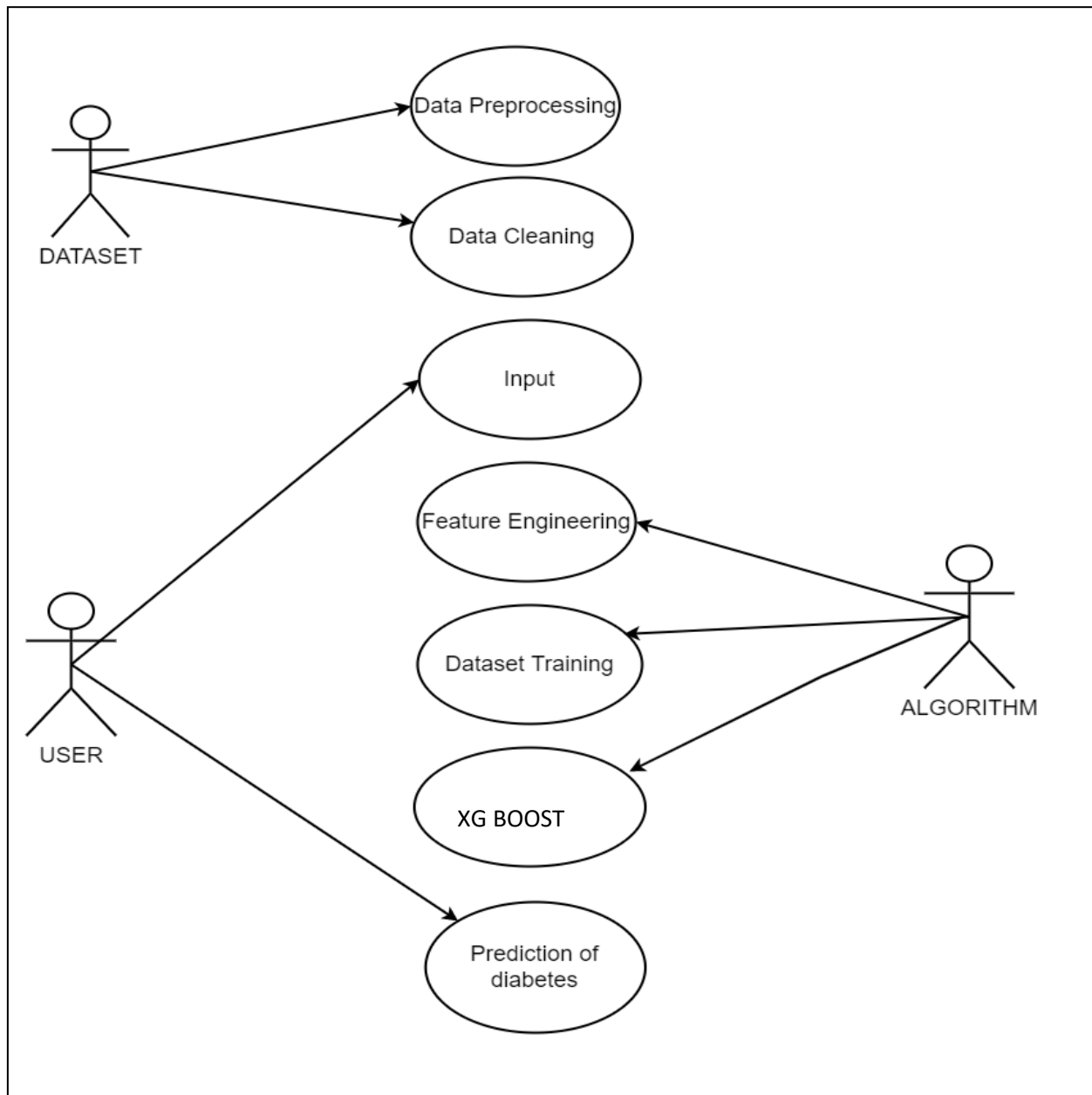


Fig 3.5 USECASE DIAGRAM



Use case diagram is a way to summarize details of a system and the users within that system. It is generally shown as a graphic depiction of interactions among different elements in a system.

### 3.2.1.5 ACTIVITY DIAGRAM

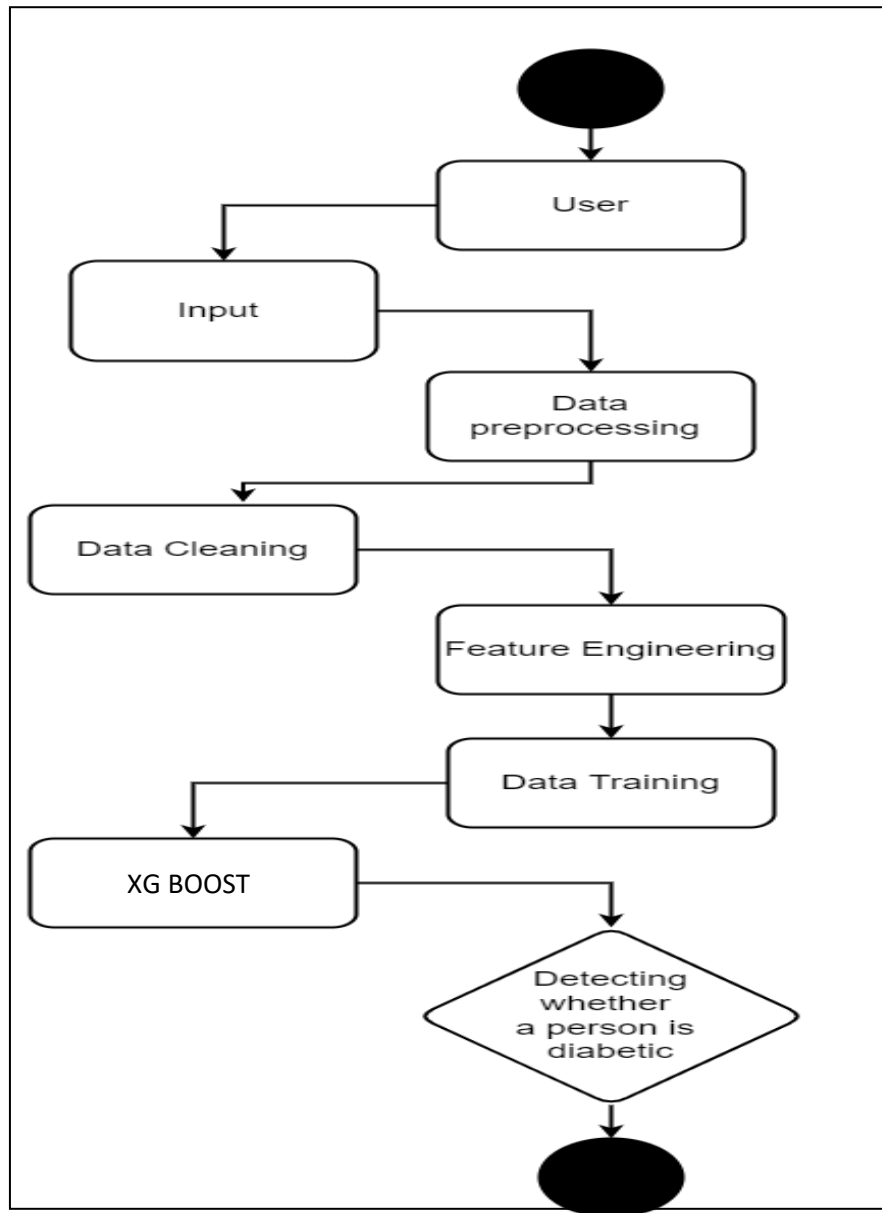


Fig 3.6 ACTIVITY DIAGRAM

An activity diagram visually presents a series of actions or flow of control in a system similar to a flowchart or a data flow diagram. Activity diagrams are often used in business process modeling. They can also describe the steps in a use case diagram. Activities modeled can be sequential and concurrent.

### 3.2.1.6 CLASS DIAGRAM

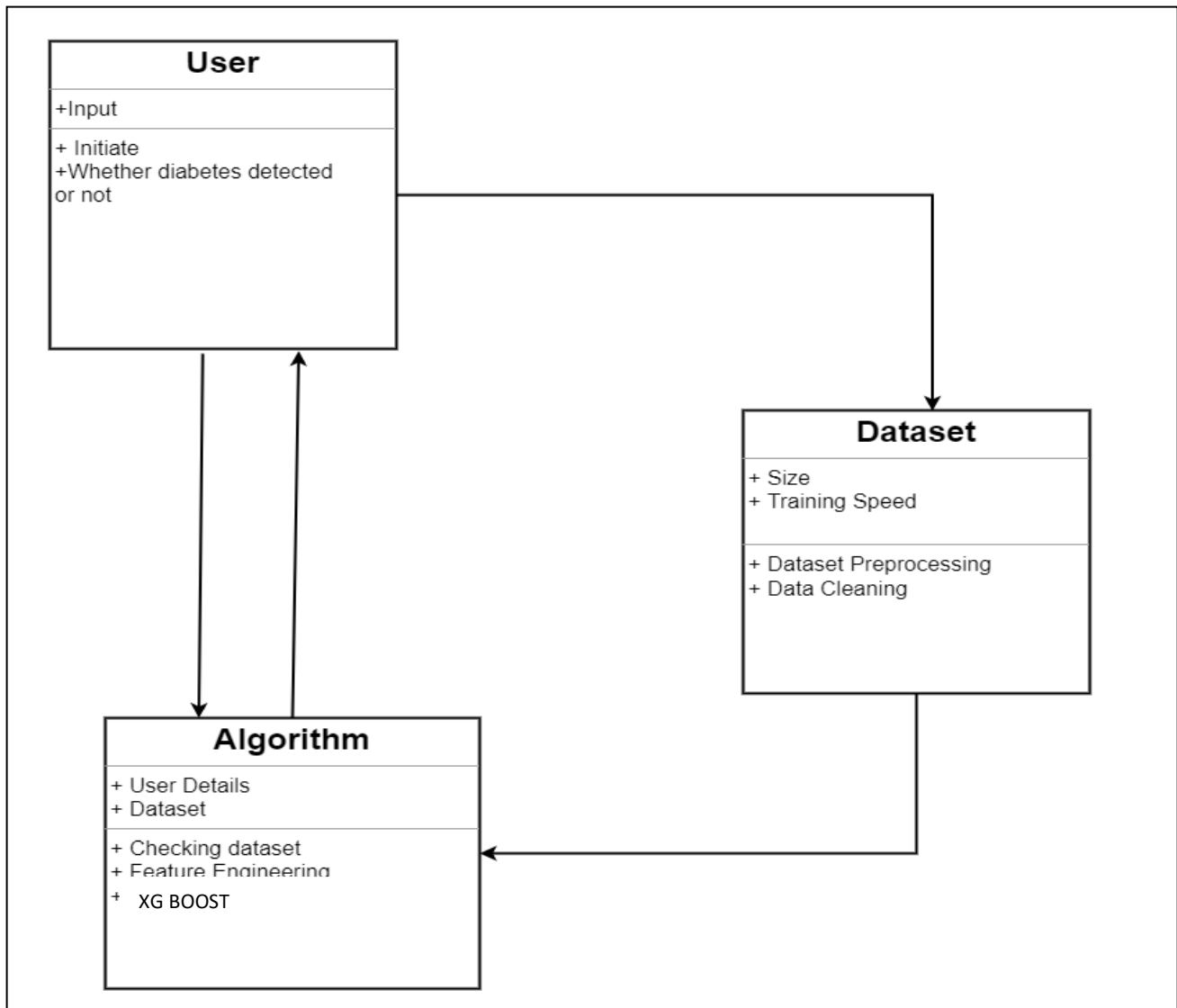


Fig 3.7 CLASS DIAGRAM

Class diagrams are one of the most useful types of diagrams in UML as they clearly map out the structure of a particular system by modeling its classes, attributes, operations, and relationships between objects. With our UML diagramming software, creating these diagrams is not as overwhelming as it might appear.

### 3.2.1.7 SEQUENCE DIAGRAM

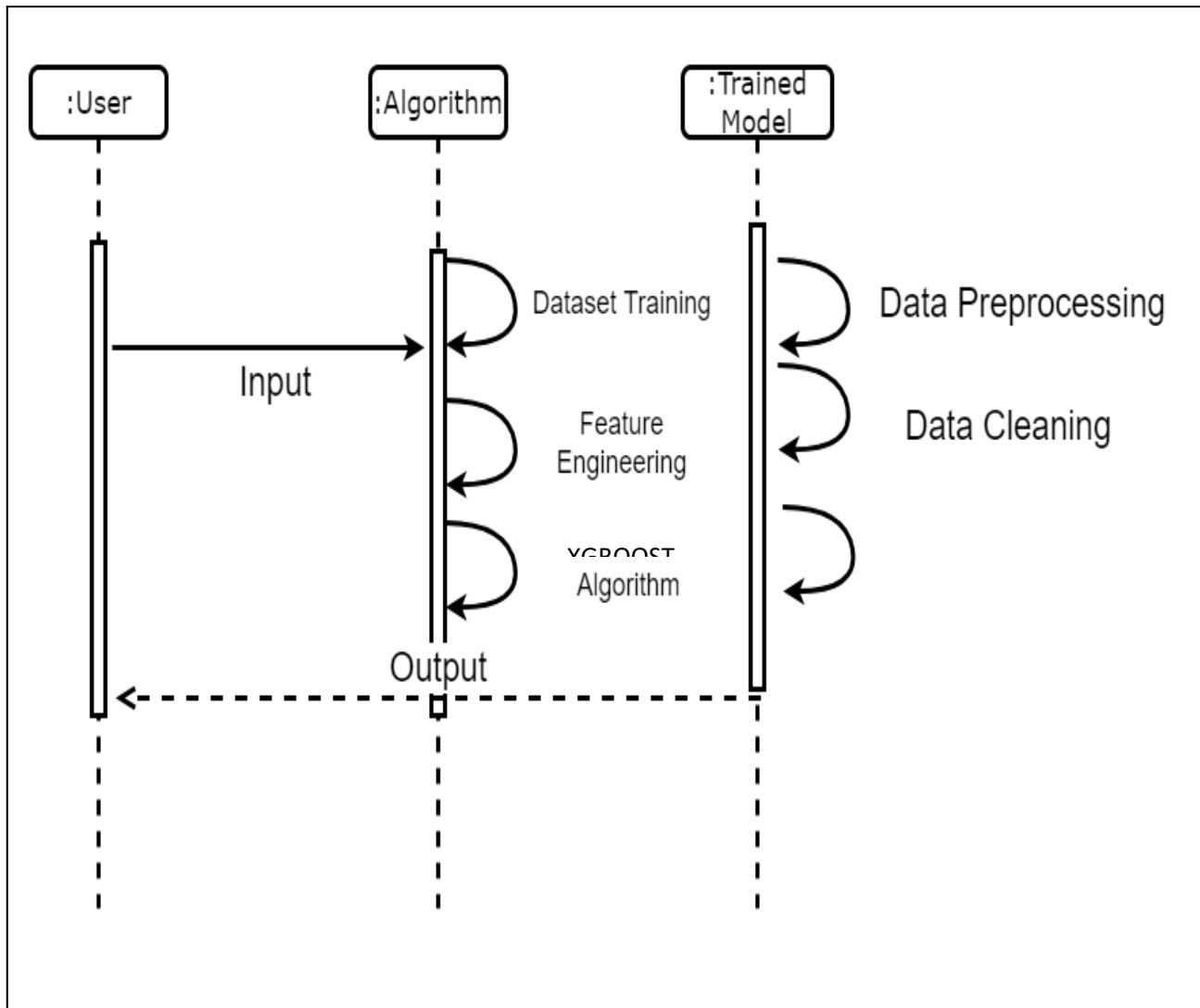


Fig 3.8 SEQUENCE DIAGRAM

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.

### 3.2.1.8 COLLOBORATION DIAGRAM

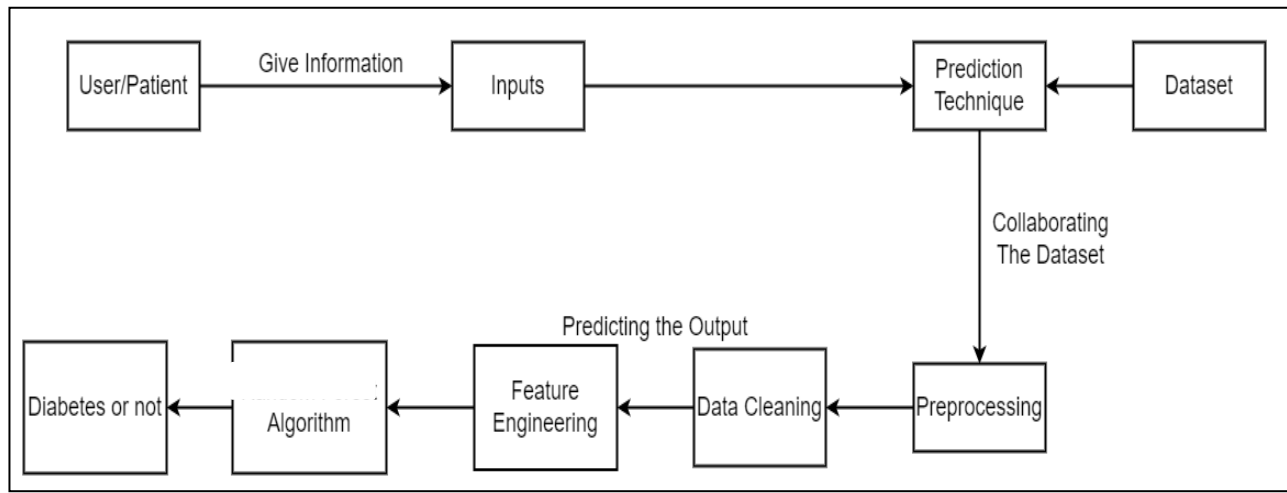


Fig 3.9 COLLOBORATION DIAGRAM

A collaboration diagram, also known as a communication diagram, is an illustration of the relationships and interactions among software objects in the Unified Modeling Language (UML). Developers can use these diagrams to portray the dynamic behavior of a particular use case and define the role of each object.

## 3.3 MODULE DESCRIPTION

### 3.3.1 DATA COLLECTION & PRE-PROCESSING

In the first step we collect data from reliable source. Glucose, Insulin, Blood pressure, Glucose pedigree function, Body mass index (BMI), Weight, Number of pregnancies and Age are some of the criteria in diabetes record set sample. Predicted outcome class, where '1' denotes a positive diabetes patient class and '0' denotes a negative diabetes patient class.

Pre-processing is next step. It's an important phase in data discovery process. The majority of health-care data contains gaps in value and inconsistencies. We apply Synthetic Minority Oversampling Technique (SMOTE) in this project, which is a well-known preprocessing approach for dealing with unbalanced datasets.

The diabetes data set we are using from <https://www.kaggle.com/johndasilva/diabetes>.

These diabetes dataset containing 2500 cases. The goal is to predict based on obtained measures to predict if the patient is affected by diabetes or not.

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	2	138	62	35	0	33.6	0.127	47	1
1	0	84	82	31	125	38.2	0.233	23	0
2	0	145	0	0	0	44.2	0.630	31	1
3	0	135	68	42	250	42.3	0.365	24	1
4	1	139	62	41	480	40.7	0.536	21	0

Fig 3.10 DATASET USED FOR PROPOSED DIABETES PREDICTION SYSTEM

The diabetes data set consists of 2500 data, with 9 features each. “Outcome” is the feature we are going to predict, 0 means no diabetes, 1 means diabetes.

### 3.3.2 FEATURE EXTRACTION

Feature extraction is the process of converting raw data into digital features that can be processed while preserving information from the original dataset. This produces better results than simply applying machine learning to raw data. This is an important classification function.

### ALGORITHMS USED IN PROPOSED SYSTEM:

#### 3.3.1.1 K-Nearest neighbours Algorithm:

K can be kept odd so that we can calculate clear majorities when only two sets are possible (eg.Red and blue). As K increases, we get smoother and better defined boundaries between the different classes. As we increase the number of data points in the training set, the accuracy of the above classifier increases.

### K-nearest neighbours Algorithm Steps:

1. Let  $m$  be the number of training data samples.
2. Let  $p$  be the unknown point. Store the training samples in the `arr []` data point array.
3. This means that each element of the array represents a tuple  $(x, y)$  .for  $i = 0$  to  $m$ :

Calculate the Euclidean distance  $d$ .

Obtain the set  $S$  of  $K$  shortest distances. Each of these distances corresponds to an already classified data point. Given the majority label under  $S$ .

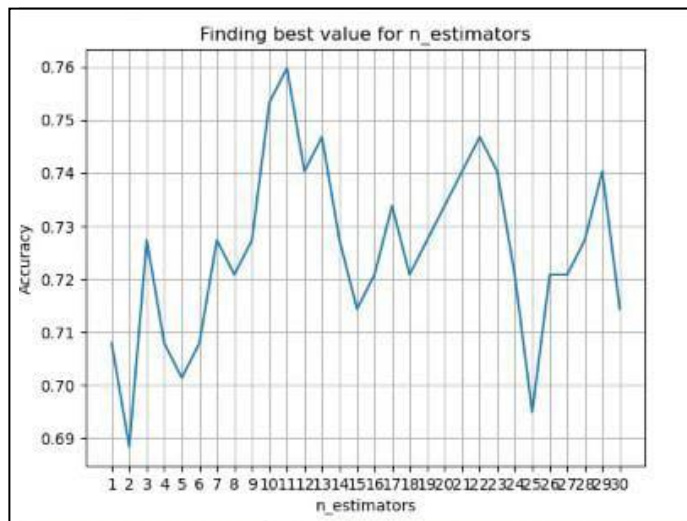


Fig 3.11 KNN ACCURACY GRAPH

### 3.3.1.2 Random Forest Algorithm:

Random forest is a commonly used machine learning algorithm, labeled by Leo Brian and Adele Cutler, which combines the outputs of multiple decision trees to obtain a single result. Its ease of use and flexibility have led to its adoption as it can handle both classification and regression issues.

The algorithm is used to predict behavior and outcomes in many industries, including banking and finance, e-commerce, and healthcare. It is increasingly used due to its ease of application, adaptability, and ability to perform classification and regression tasks.

### Random forest Algorithm steps:

1. Randomly select K data points from the training set.
2. Build a decision tree associated with the selected data points (subset).
3. Select the number N for the decision tree to build.
4. Repeat steps 1 and 2.
5. For a new data point, find the prediction of each decision tree and assign the new data point to the category that won the majority vote.

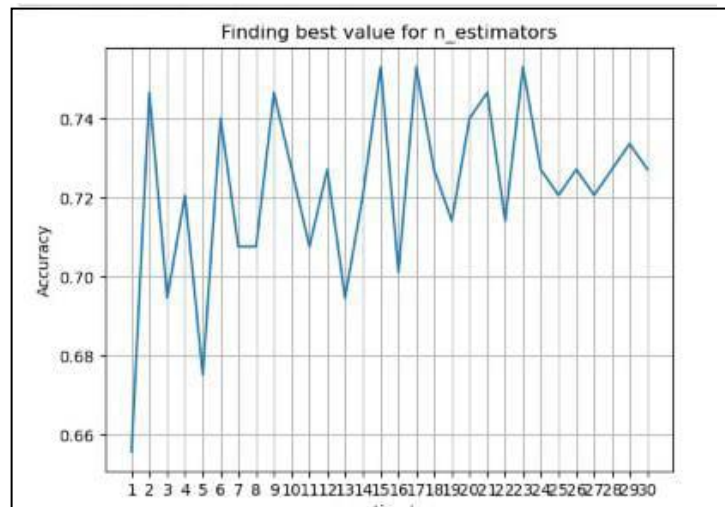


Fig 3.12 RANDOM FOREST ACCURACY GRAPH

### 3.3.1.3 Lbgbm or LightGBM Algorithm:

LightGBM is a gradient boosting ensemble method used by the Train Using Auto ML tool, which is based on decision trees. Like other decision tree-based methods, LightGBM can be used for both classification and regression. LightGBM is optimized for high performance in distributed systems.

#### LightGBM Algorithm Steps:

1. Assign small values to max\_bin and num\_leaves.
2. Using large amounts of training data.
3. Use max\_depth to avoid deep trees.

4. Use bagging by setting bagging\_fraction and bagging\_freq.
5. Set feature\_fraction to use feature downsampling.

### 3.3.1.4 XG Boost Algorithm:

XGBoost (eXtreme Gradient Boosting) is a popular and efficient open source implementation of the gradient boost tree algorithm. Gradient boosting is a supervised learning algorithm that attempts to accurately predict a target variable by combining ensemble estimates from a set of simpler, weaker models.

#### XGBoost Algorithm Steps:

1. Make initial predictions and calculate residuals.
2. Create the XGBoost tree.
3. Prune the tree.
4. Calculate the output value of the sheet
5. Make a new prediction
6. Calculate the residuals using the new prediction.

Below output shows the comparisons of all four Machine Learning Algorithms that gives us an accuracy rates that XGBoost gives the higher rate of 92.7%.

```
In [22]: from sklearn.metrics import accuracy_score
accuracy_knn = accuracy_score(Y_test, Y_pred_knn)
accuracy_ranfor = accuracy_score(Y_test, Y_pred_ranfor)
accuracy_lgbm = accuracy_score(Y_test, Y_pred_lgbm)
accuracy_xgbr = accuracy_score(Y_test, Y_pred_xgbr)

In [23]: print("K Nearest neighbors: " + str(accuracy_knn * 100))
print("Random Forest: " + str(accuracy_ranfor * 100))
print("Light BGM: " + str(accuracy_lgbm * 100))
print("XG Boost: " + str(accuracy_xgbr * 100))

K Nearest neighbors: 72.07792207792207
Random Forest: 68.83116883116884
Light BGM: 70.12987012987013
XG Boost: 92.72727272727273

In [24]: from sklearn.metrics import confusion_matrix
cm = confusion_matrix(Y_test, Y_pred_xgbr)
cm

Out[24]: array([[183, 17],
               [25, 29]], dtype=int64)

In [25]: from sklearn.metrics import classification_report
print(classification_report(Y_test, Y_pred_xgbr))

              precision    recall  f1-score   support

     0.0       0.77       0.83       0.80       100
     1.0       0.63       0.54       0.58        54

   accuracy          0.70
  macro avg          0.70      0.68      0.69       154
 weighted avg          0.72      0.73      0.72       154

In [26]: import pickle
pickle.dump(xgbr, open('model.pkl', 'wb'))
model = pickle.load(open('model.pkl', 'rb'))
```

Fig 3.13 COMPARISON OF ALL 4 ALGORITHMS USED IN THE TRAINING DATASET



### **3.3.3 MODEL CREATION**

The XGBoost Algorithm is used in this project. It's a classification and regression supervised machine learning algorithm. Every single accessible example is categorized according to XGBoost calculation. A case is assigned to the class with the most votes from classifier with the case being relegated to the Random forest class with most votes calculated using a separation function. Examining the set of information data determines the estimation.

### **3.3.4 TRAINING & TESTING**

During training, we train the machine from a data source. Test data is transformed and predicts accurate results. During training, machine learning automatically selects the correct training algorithm based on the target type specified in the training data source. The training dataset to validate with the test dataset model. After transforming the test data, the accurate prediction result is 92.7% is achieved by the XGBoost algorithm.

Training data set which will be validated using test dataset model. The test data is transformed and predicts accurate result is 92.7% is achieved by XGBoost algorithm.

### **3.3.5 PREDICTION PROCESS**

This module predicts whether the user is diabetic or not using the XGBoost algorithm. And with the help of diabetes symptoms to predict normal levels of diabetes, type 1 and type 2. After training the model, we measured it using different parameters and accuracies in the dataset. Diabetes predictions achieved a 92.7% machine learning rate using XGBoost. In the future, this layered framework combined with machine learning algorithms can be used to predict or analyze various anomalies.

Additional ML calculations can be used to improve and enhance diabetes screening efforts. Training data set which will be validated using test dataset model. The test data is transformed and predicts accurate result is 92.7% is achieved by XGBoost algorithm.

### 3.4 PROPOSED SYSTEM ALGORITHM

#### 3.4.1 XGBOOST ALGORITHM:

XG Boost can be used in a variety of applications, including Kaggle competitions, recommender systems, and click-through rate prediction. It is also customizable, and various model parameters can be tuned to optimize performance. XG Boost is an implementation of a gradient boost decision tree. The XG Boost model basically dominates a lot of Kaggle competition. In this algorithm, decision trees are built in sequential form. Weight plays an important role in XG Boost. All independent variables are assigned weights and entered into a decision tree that predicts an outcome. Variables that the tree predicts incorrectly are weighted and these variables are passed to the second decision tree. These individual classifiers/predictors are then combined to create a robust and accurate model. It can handle custom regression, classification, ranking, and prediction problems.

##### 3.4.1.1 The Mathematics of XGBoost:

Before learning the math of gradient boosting, here's a simple CART example to rate if someone likes the hypothetical PC game X. Tree example:

Obtained by summing the assertion predictions for each individual decision tree. Looking at this example, the important thing is that the two trees try to complement each other. Mathematically, the model can be written as:

$$\hat{y}_i = \sum_{k=1}^K f_k(x_i), f_k \in \mathcal{F} \quad \text{-----} \quad (1)$$

where, K is number of trees, f is functional space of F, F is set of possible CARTs. The objective function for the above model is given by:

$$obj(\theta) = \sum_i^n l(y_i, \hat{y}_i) + \sum_{k=1}^K \Omega(f_k) \quad \text{-----} \quad (2)$$

where, first term is loss function and second is the regularization parameter. Now, Instead of learning tree all at once which makes optimization harder, we apply the additive strategy, minimize loss what we have learned and add a new tree which can be summarized below:

$$\begin{aligned}
 \hat{y}_i^{(0)} &= 0 \\
 \hat{y}_i^{(1)} &= f_1(x_i) = \hat{y}_i^{(0)} + f_1(x_i) \\
 \hat{y}_i^{(2)} &= f_1(x_i) + f_2(x_i) = \hat{y}_i^{(1)} + f_2(x_i) \\
 &\dots \\
 \hat{y}_i^{(t)} &= \sum_{k=1}^t f_k(x_i) = \hat{y}_i^{(t-1)} + f_t(x_i)
 \end{aligned}
 \tag{3}$$

The objective function of above model can be defined as:

$$\begin{aligned}
 obj^{(t)} &= \sum_{i=1}^n l(y_i, \hat{y}_i^{(t)}) + \sum_{i=1}^t \Omega(f_i) \\
 &= \sum_{i=1}^n l(y_i, \hat{y}_i^{(t-1)} + f_t(x_i)) + \Omega(f_t) + constant \\
 obj^{(t)} &= \sum_{i=1}^n (y_i - (\hat{y}_i^{(t-1)} + f_t(x_i)))^2 + \sum_{i=1}^t \Omega(f_i) \\
 &= \sum_{i=1}^n [2(\hat{y}_i^{(t-1)} - y_i)f_t(x_i) + f_t(x_i)^2] + \Omega(f_t) + constant
 \end{aligned}
 \tag{4}$$

Now, let's apply Taylor series expansion up to second order:

$$obj^{(t)} = \sum_{i=1}^n [l(y_i, \hat{y}_i^{(t-1)}) + g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i)] + \Omega(f_t) + constant \tag{5}$$

where  $g_i$  and  $h_i$  can be defined as:

$$\begin{aligned}
 g_i &= \partial_{\hat{y}_i^{(t-1)}} l(y_i, \hat{y}_i^{(t-1)}) \\
 h_i &= \partial_{\hat{y}_i^{(t-1)}}^2 l(y_i, \hat{y}_i^{(t-1)})
 \end{aligned}
 \tag{6}$$

Simplifying and removing constant:

$$\sum_{i=1}^n [g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i)] + \Omega(f_t) \tag{7}$$

Now, we define regularization term, but first we need to define the model:

$$f_t(x) = w_{q(x)}, w \in R^T, q: R^d \rightarrow \{1, 2, \dots, T\} \quad \text{-----}(8)$$

Here,  $w$  is vector of scores on leaves of tree,  $q$  is function assigning each data point to corresponding leaf, and  $T$  is number of leaves. The regularization term is then defined by:

$$\Omega(f) = \gamma T + \frac{1}{2} \lambda \sum_{j=1}^T w_j^2 \quad \text{-----}(9)$$

Now, our objective function becomes:

$$\begin{aligned} obj^{(t)} &\approx \sum_{i=1}^n [g_i w_{q(x_i)} + \frac{1}{2} h_i w_{q(x_i)}^2] + \gamma T + \frac{1}{2} \lambda \sum_{j=1}^T w_j^2 \\ &= \sum_{j=1}^T [(\sum_{i \in I_j} g_i) w_j + \frac{1}{2} (\sum_{i \in I_j} h_i + \lambda) w_j^2] + \gamma T \end{aligned} \quad \text{-----}(10)$$

### 3.4.2 ADVANTAGES OF XGBOOST:

**i) Performance:** XG Boost has a proven track record of high quality results in a variety of machine learning tasks, especially in Kaggle competitions where it has been a popular choice for winning solutions.

**ii) Scalability:** XG Boost is designed for efficient and scalable training of machine learning models, making it suitable for large datasets.

**iii) Tunability:** XG Boost has a wide range of hyper parameters that can be tuned to optimize performance, making it highly customizable.

**iv) Missing value handling:** XG Boost has built-in support for missing value handling, which can easily handle real data that often contains missing values.

**v) Interpretability:** Unlike some machine learning algorithms that are difficult to interpret, XG Boost provides feature importance, which provides insight into which variables are most important for making predictions.

### **3.5 ADVANTAGES OF THE PROPOSED MODEL:**

- i) Applicable to unstructured and semi-structured datasets such as images and text.
- ii) Accurate and robust results can be obtained.
- iii) Successfully used in medical applications.

# **CHAPTER 4**

## **REQUIREMENT SPECIFICATION**

## **4.1 REQUIREMENT SPECIFICATIONS**

- **Software specifications:**
  - Python 3.6 and higher
  - Anaconda software
  
- **Hardware specifications:**
  - Microsoft Server enabled computers, preferably workstations
  - Higher RAM, of about 4GB or above
  - Processor of frequency 1.5GHz or above

## **4.2 SOFTWARE SPECIFICATIONS**

### **4.2.1 PYTHON 3.6**

Python is a high-level object-oriented programming language that was created by Guido van Rossum. It is also called general-purpose programming language as it is used in almost every domain we can think of as mentioned below:

- Web Development
- Software Development
- Game Development
- AI & ML
- Data Analytics

This list can go on as we go but why python is so much popular let's see it in the next topic.

### **4.1.2 ANACONDA SOFTWARE**

Anaconda is a distribution of the Python and R programming languages for scientific computing (data science, machine learning applications, large-scale data processing, predictive analytics, etc.), that aims to simplify package management and deployment. The distribution includes data-science packages suitable for Windows, Linux, and macOS. It is developed and maintained by Anaconda, Inc., which was founded by Peter Wang and Travis Oliphant in 2012. As an Anaconda, Inc. product, it is also known as Anaconda Distribution or Anaconda Individual Edition, while other products from the company are Anaconda Team Edition and Anaconda Enterprise Edition, both of which are not free.

Package versions in Anaconda are managed by the package management system conda. This package manager was spun out as a separate open-source package as it ended up being useful on its own and for things other than Python. There is also a small, bootstrap version of Anaconda called Miniconda, which includes only conda, Python, the packages they depend on, and a small number of other packages.

## **4.3 HARDWARE REQUIREMENTS**

### **4.3.1 MICROSOFT SERVER ENABLED COMPUTERS, PREFERABLY WORKSTATIONS**

Workstation is a computer which is used to access LAN or Internet to get access to documents or perform a task, whereas a server is a software which responds to services requested by a client. These are the types of workstation: Computer workstations. Printer workstations. General workstations. Remote engine workstations.

Any computer, even a home desktop or laptop computer, can act as a server with the right software. For example, you could install an FTP server program on your computer to share files between other users on your network.

### **4.3.2 HIGHER RAM, OF ABOUT 4GB**

4GB RAM is a significant memory on a laptop, as it offers enough power and storage for most users. However, not all laptops come with this amount of memory.



### **4.3.3 PROCESSOR OF FREQUENCY 1.5GHZ**

The GHz rating, or Gigahertz rating of a processor refers to the frequency at which that processor can process instructions. Essentially it means that a 1.5GHz processor can process 1.5 billion instructions per second.

There is a range of spectrum uses currently supported in the 1.5 GHz band. These include applications in the mobile (aeronautical mobile), fixed (both point-to-point and point-multipoint), radio astronomy, and meteorological satellite services.

A clock speed of 3.5 GHz to 4.0 GHz is generally considered a good clock speed for gaming but it's more important to have good single-thread performance. This means that your CPU does a good job of understanding and completing single tasks. This is not to be confused with having a single-core processor. Intel claims that the 5.6GHz turbo boost clock speed of the new Core i9 chip is the fastest mobile processor as of December 2022, with 11 percent faster single-thread performance and 49 percent faster multitask performance over the prior-generation Intel Core i9-12900HK chip.

Clock speed is measured in GHz (gigahertz), a higher number means a faster clock speed. To run your apps, your CPU must continually complete calculations, if you have a higher clock speed, you can compute these calculations quicker and applications will run faster and smoother as a result of this.

The 8121U was a terrible demonstration of 10nm and a terrible product in its own right. The 10nm node was so broken that Intel could only manufacture a tiny dual-core CPU with its integrated graphics intentionally disabled, presumably because they didn't work properly.

# **CHAPTER 5**

## **IMPLEMENTATION**

## 5.1 IMPLEMENTATION

Inputs for our proposed web application with Diabetes Mellitus Predictive Analysis Using Medical System Parameter are obtained from the input dataset.

Once the system is trained with the training set using the mentioned algorithms a rule set is formed and when the input dataset is given as an input to the model, the data are processed according to the rule set developed, thus giving the outcome as diabetes diagnosed or not diagnosed.

## 5.2 WORKING

The proposed model has two main steps. Our first step is to form layers, while the second step is to test layers. However, in our proposed model, we only use two commonly used ML algorithms. If our proposed model does not meet the learning requirements, it will be recycled. The second stage of the proposed framework is reflected by the test layer. The test layer gets the dataset from the medical database and loads the preprocessed training model from the cloud. In our proposed framework, the validation layer is related to the real-time diagnosis and classification of diabetes. The proposed ML fusion model can use real-time patient data as input and improve disease detection systems. The input data is collected from the source and go under preprocessing. And preprocessed data is first trained using the machine learning algorithm and the features are extracted and trained by the highest accuracy algorithm that is XGBoost. Then the created model is tested and compared using the machine learning algorithm. And finally the prediction is done and the output will be predicted as 0's and 1's.

## 5.3 CODE

### 5.3.1 INDEX.HTML

```
<!DOCTYPE html>

<html >

<!--From https://codepen.io/frytyler/pen/EGdtg-->

<head>

  <meta charset="UTF-8">

  <title>Diabetes Prediction system</title>
```

```

<link href='https://fonts.googleapis.com/css?family=Pacifico' rel='stylesheet' type='text/css'>
<link href='https://fonts.googleapis.com/css?family=Arimo' rel='stylesheet' type='text/css'>
<link href='https://fonts.googleapis.com/css?family=Hind:300' rel='stylesheet' type='text/css'>
<link href='https://fonts.googleapis.com/css?family=Open+Sans+Condensed:300' rel='stylesheet'
type='text/css'>
<link rel="stylesheet" href="{ { url_for('static', filename='css/style.css') } }">

<style>

::placeholder {

    color: #FFFFFFF;

    opacity: 1;

}

:-ms-input-placeholder { /* Internet Explorer 10-11 */

    color: black;

}

::-ms-input-placeholder { /* Microsoft Edge */

    color: black;

}

.button {

    background-color: #4CAF50; /* Green */

    border: none;

    color: black;

    padding: 15px 32px;

    text-align: center;

```

```

text-decoration: none;

display: inline-block;

font-size: 10px;

margin: 4px 2px;

cursor: pointer;

}

</style>

</head>

<center><b><h2      style="color:black;      font-family:serif;      font-size:      30px";>{{
prediction_text}}</h2></b></center>

<body>

<div class="login" style="margin-left: 50px";>

<style>

body{

color:White;

}

</style>

<body>

    <h2>Diabetes Prediction system</h2>

</body>

    <form action="{{ url_for('predict')}}" method="post">

        <input type="text" name="Glucose Level" placeholder="Glucose Level"
required="required" />

        <input type="text" name="Insulin" placeholder="Insulin" required="required" />

        <input type="text" name="BMI" placeholder="BMI" required="required" />

```

```

        <input type="text" name="age" placeholder="Age" required="required" />

        <button      type="submit"      class="btn      btn-primary      btn-block      btn-large"
style='color:black'>Predict</button>

    </form>

    <br>

    <br>

</div>

</body>

</html>

```

### 5.3.2 STYLE.CSS

```

@import url(https://fonts.googleapis.com/css?family=Open+Sans);

.btn { display: inline-block; *display: inline; *zoom: 1; padding: 4px 10px 4px; margin-bottom:
0; font-size: 13px; line-height: 18px; color: #333333; text-align: center;text-shadow: 0 1px 1px
rgba(255, 255, 255, 0.75); vertical-align: middle; background-color: #f5f5f5; background-image:
-moz-linear-gradient(top, #ffffff, #e6e6e6); background-image: -ms-linear-gradient(top, #ffffff,
#e6e6e6); background-image: -webkit-gradient(linear, 0 0, 0 100%, from(#ffffff), to(#e6e6e6));
background-image: -webkit-linear-gradient(top, #ffffff, #e6e6e6); background-image: -o-linear-
gradient(top, #ffffff, #e6e6e6); background-image: linear-gradient(top, #ffffff, #e6e6e6);
background-repeat:                repeat-x;                filter:
progid:dximagetransform.microsoft.gradient(startColorstr=#ffffff,                endColorstr=#e6e6e6,
GradientType=0); border-color: #e6e6e6 #e6e6e6 #e6e6e6; border-color: rgba(0, 0, 0, 0.1) rgba(0,
0, 0, 0.1) rgba(0, 0, 0, 0.25); border: 1px solid #e6e6e6; -webkit-border-radius: 4px; -moz-border-
radius: 4px; border-radius: 4px; -webkit-box-shadow: inset 0 1px 0 rgba(255, 255, 255, 0.2), 0 1px
2px rgba(0, 0, 0, 0.05); -moz-box-shadow: inset 0 1px 0 rgba(255, 255, 255, 0.2), 0 1px 2px rgba(0,
0, 0, 0.05); box-shadow: inset 0 1px 0 rgba(255, 255, 255, 0.2), 0 1px 2px rgba(0, 0, 0, 0.05);
cursor: pointer; *margin-left: .3em; }

```

```

.btn:hover, .btn:active, .btn.active, .btn.disabled, .btn[disabled] { background-color: #e6e6e6; }

.btn-large { padding: 9px 14px; font-size: 15px; line-height: normal; -webkit-border-radius: 5px;
-moz-border-radius: 5px; border-radius: 5px; }

.btn:hover {
    color: #e01baf;
    text-decoration: none;
    background-color: #e6e6e6;
    background-position: 0 -15px;
    -webkit-transition: background-position 0.1s linear;
    -moz-transition: background-position 0.1s linear;
    -ms-transition: background-position 0.1s linear;
    -o-transition: background-position 0.1s linear;
    transition: background-position 0.1s linear;
}

.btn-primary, .btn-primary:hover { text-shadow: 0 -1px 0 rgba(0, 0, 0, 0.25); color: #ffffff; }

.btn-primary.active { color: rgba(255, 255, 255, 0.75); }

.btn-primary { background-color: #e01baf; background-image: -moz-linear-gradient(top,
#6eb6de, #4a77d4); background-image: -ms-linear-gradient(top, #6eb6de, #4a77d4); background-
image: -webkit-gradient(linear, 0 0, 0 100%, from(#6eb6de), to(#4a77d4)); background-image: -
webkit-linear-gradient(top, #6eb6de, #4a77d4); background-image: -o-linear-gradient(top,
#6eb6de, #4a77d4); background-image: linear-gradient(top, #6eb6de, #4a77d4); background-
repeat: repeat-x; filter: progid:dximagetransform.microsoft.gradient(startColorstr=#6eb6de,
endColorstr=#4a77d4, GradientType=0); border: 1px solid #3762bc; text-shadow: 1px 1px 1px
rgba(0,0,0,0.4); box-shadow: inset 0 1px 0 rgba(255, 255, 255, 0.2), 0 1px 2px rgba(0, 0, 0, 0.5);
}

.btn-primary:hover, .btn-primary:active, .btn-primary.active, .btn-primary.disabled, .btn-
primary[disabled] { filter: none; background-color: #4a77d4; }

.btn-block { width: 100%; display: block; }

```

```
* { -webkit-box-sizing:border-box; -moz-box-sizing:border-box; -ms-box-sizing:border-box; -o-box-sizing:border-box; box-sizing:border-box; }
```

```
html { width: 100%; height:100%; overflow:hidden; }
```

```
body {  
    width: 100%;  
    height: 100%;  
    font-family: 'Open Sans', sans-serif;  
    background: url("/static/bg1.jpg");  
    background-size: 100%;  
    color: #fff;  
    font-size: 18px;  
    text-align: center;  
    letter-spacing: 1.2px;  
}
```

```
.login {  
    position: absolute;  
    top: 40%;  
    left: 50%;  
    margin: -150px 0 0 -150px;  
    width:400px;  
    height:400px;  
}
```



```
.login h1 { color: #fff; text-shadow: 0 0 30px rgba(0,0,0,0.3); letter-spacing: 1px; text-align: center;
}
```

```
input {
    width: 100%;
    margin-bottom: 10px;
    background: rgba(0,0,0,0.3);
    border: none;
    outline: none;
    padding: 10px;
    font-size: 13px;
    color: #fff;
    text-shadow: 2px 2px 2px rgba(0,0,0,0.3);
    border: 1px solid rgba(0,0,0,0.3);
    border-radius: 4px;
    box-shadow: inset 0 -5px 45px rgba(100,100,100,0.2), 0 1px 1px rgba(255,255,255,0.2);
    -webkit-transition: box-shadow .5s ease;
    -moz-transition: box-shadow .5s ease;
    -o-transition: box-shadow .5s ease;
    -ms-transition: box-shadow .5s ease;
    transition: box-shadow .5s ease;
}

input:focus { box-shadow: inset 0 -5px 45px rgba(100,100,100,0.4), 0 1px 1px
rgba(255,255,255,0.2); }
```

### 5.3.3 APP.PY

```
import numpy as np

import pandas as pd

from flask import Flask, request, jsonify, render_template

import pickle


app = Flask(__name__)

model = pickle.load(open('model.pkl', 'rb'))


dataset = pd.read_csv('diabetes.csv')


dataset_X = dataset.iloc[:,[1, 2, 5, 7]].values


from sklearn.preprocessing import MinMaxScaler

sc = MinMaxScaler(feature_range = (0,1))

dataset_scaled = sc.fit_transform(dataset_X)


@app.route('/')

def home():

    return render_template('index.html')


@app.route('/predict',methods=['POST'])
```

```

def predict():

    float_features = [float(x) for x in request.form.values()]

    final_features = [np.array(float_features)]

    prediction = model.predict( sc.transform(final_features) )


    age=request.form['age']

    age=int(age)

    #print(age)

    if age <= 24:

        if prediction == 1:

            pred = """"You have Diabetes Type I, please consult a Doctor Soon.....!

            Precautions:

            i) Make healthy eating and physical activity part of your daily routine.

            ii) Maintain a healthy wait.

            iii)Take proper medications...!""""

        elif prediction == 0:

            pred = "You don't have Diabetes."


    else:

        if prediction == 1:

            pred = "You have Diabetes Type II, please consult a Doctor...Precautions: i)Keep weight
under control. ii)Do Exercise. iii)Eat healthy. iv) Do not smoke."

        elif prediction == 0:

            pred = "You don't have Diabetes !!!"

```

```
output = pred

return render_template('index.html', prediction_text='{}'.format(output))

if __name__ == "__main__":
    app.run(host='0.0.0.0', port='400')
```

## 5.4 OUTPUT

The sample output of the proposed model and working screenshots are attached below:

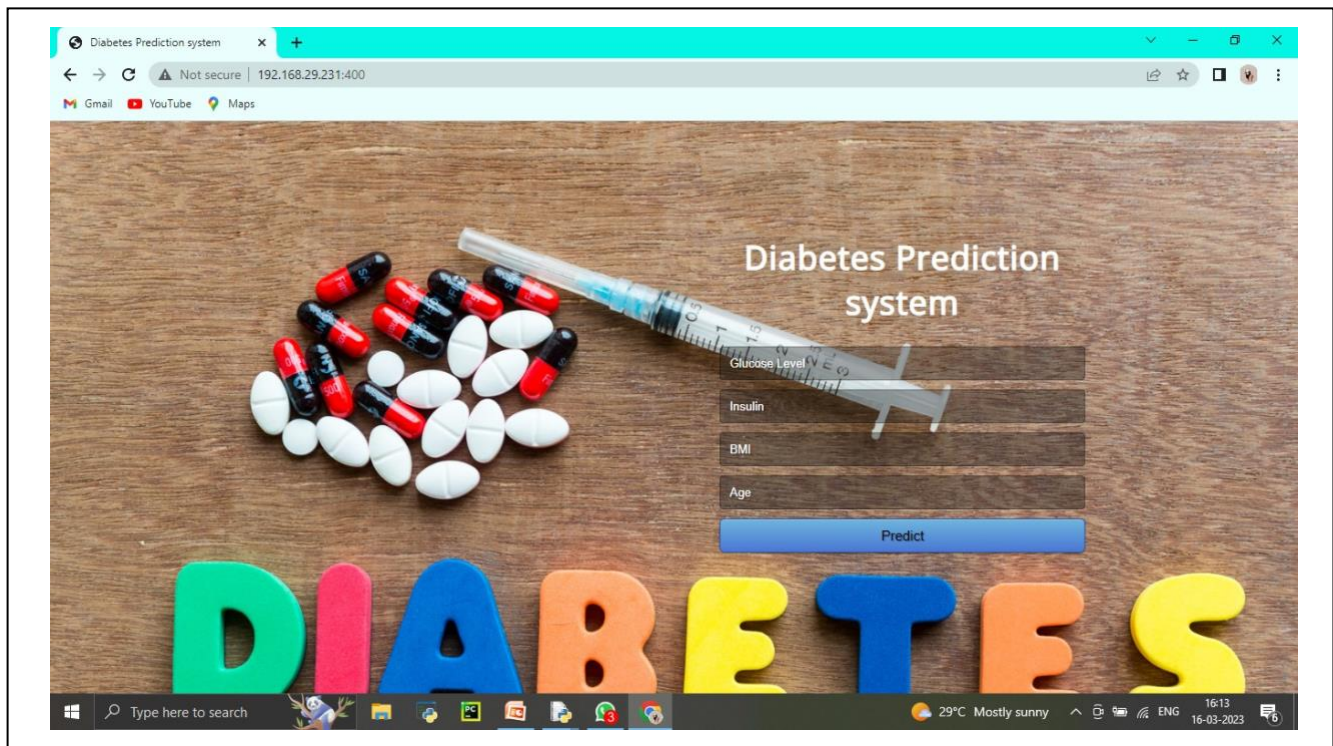


Fig 4.1 SAMPLE OUTPUT

# **CHAPTER 6**

## **TESTING AND MAINTENANCE**

## 6. TESTING AND MAINTENANCE

### 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 should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

### 6.2 INTEGRATION TESTING

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

### 6.3 FUNCTIONAL TEST

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

Functional testing is centered on the following items:

- Valid Input** : identified classes of valid input must be accepted.
- Invalid Input** : identified classes of invalid input must be rejected.
- Functions** : identified functions must be exercised.
- Output** : identified classes of application outputs must be exercised.

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

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

## **6.4 SYSTEM TEST**

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

## **6.5 WHITE BOX TESTING**

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

## **6.6 BLACK BOX TESTING**

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

# **CHAPTER 7**

## **OUTPUT AND DISCUSSION**



## 7. OUTPUT AND DISCUSSION

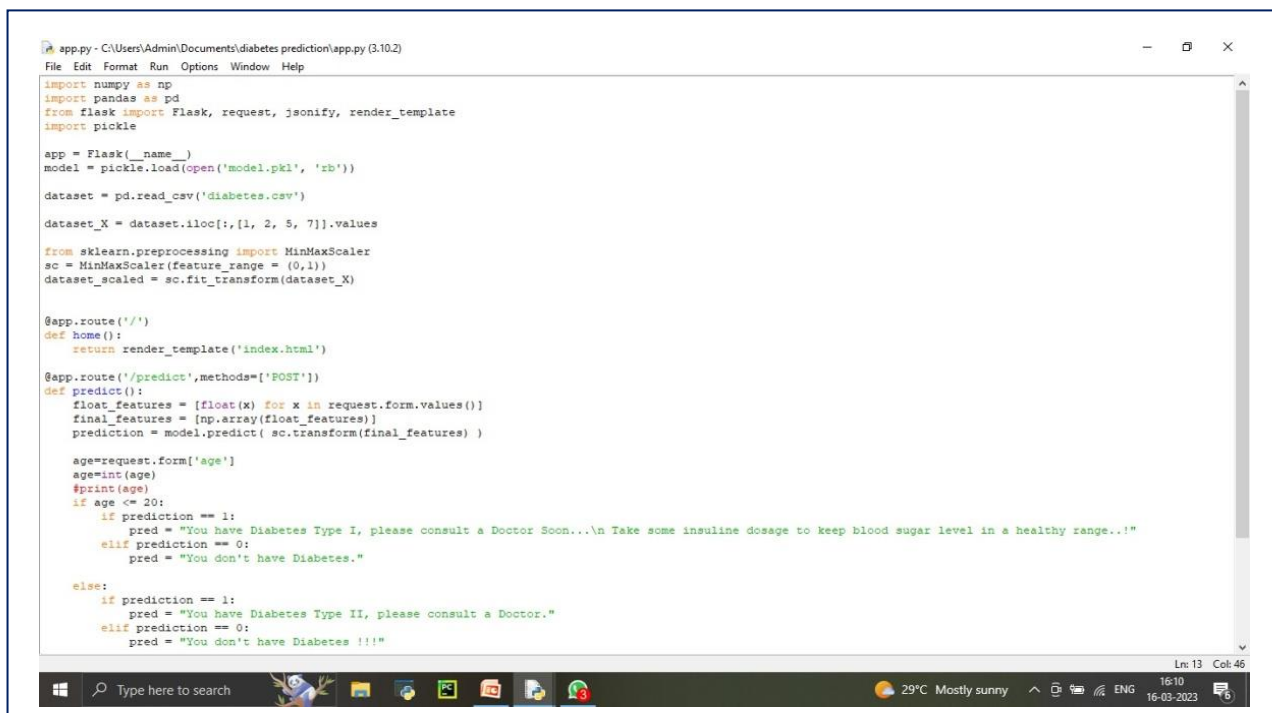
### 7.1 PREDICTION

This module predicts that the user is suffering from diabetic or not using XG Boost algorithm. And also predict diabetic level normal or high using diabetic symptoms. After training model we had measured with different parameters within datasets and accuracy.

Diabetes Prediction Using Machine learning rate of XGBoost with 92.7% is achieved. In future, this hierarchical framework combined with machine learning algorithms could be used to predict or analyses various disorders. Other ML computations can be used to enhance and improves job for diabetes examination.

### 7.2 PROPOSED SYSTEM IMPLEMENTATION

After training the data sets the prediction process is done and the output will displayed in the web app by running the python code which is giving the final output by using the python flask and many more functions used in it.



```
app.py - C:\Users\Admin\Documents\diabetes prediction\app.py (3.10.2)
File Edit Format Run Options Window Help

import numpy as np
import pandas as pd
from flask import Flask, request, jsonify, render_template
import pickle

app = Flask(__name__)
model = pickle.load(open('model.pkl', 'rb'))

dataset = pd.read_csv('diabetes.csv')

dataset_X = dataset.iloc[:, [1, 2, 5, 7]].values

from sklearn.preprocessing import MinMaxScaler
sc = MinMaxScaler(feature_range = (0,1))
dataset_scaled = sc.fit_transform(dataset_X)

@app.route('/')
def home():
    return render_template('index.html')

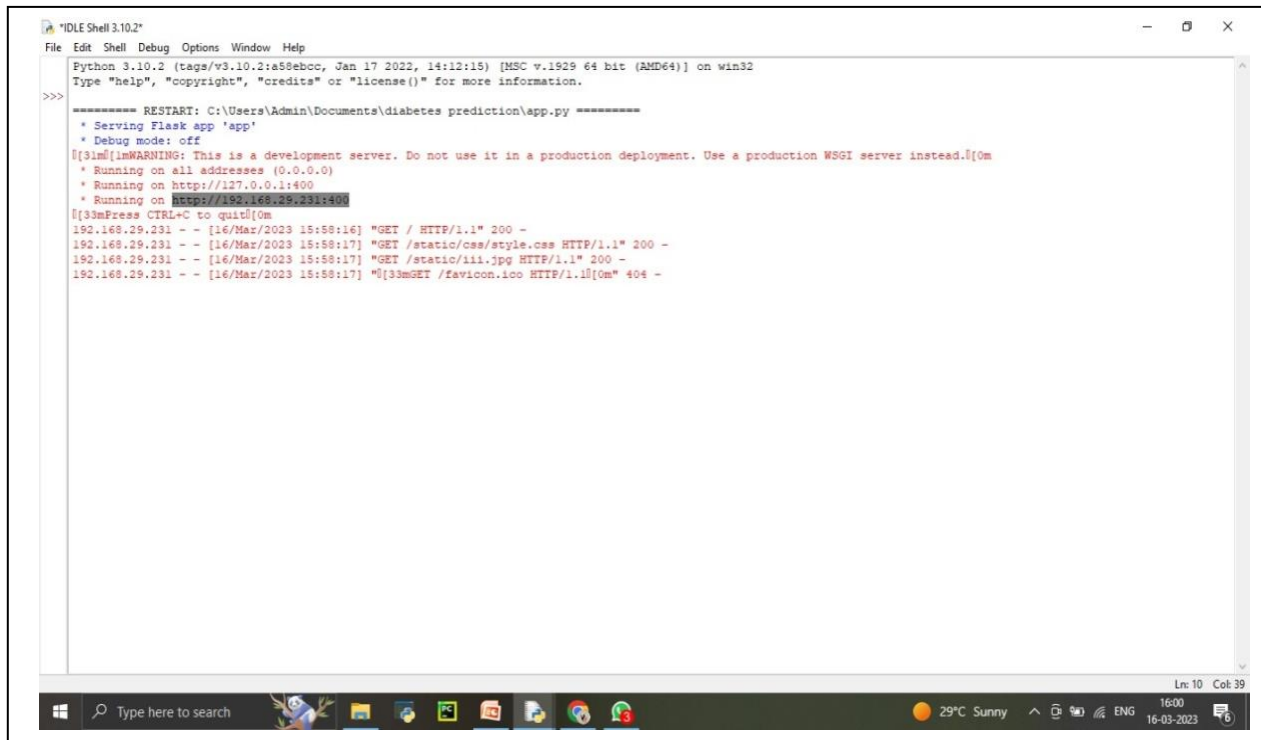
@app.route('/predict', methods=['POST'])
def predict():
    float_features = [float(x) for x in request.form.values()]
    final_features = [np.array(float_features)]
    prediction = model.predict( sc.transform(final_features) )

    age=request.form['age']
    age=int(age)
    #print(age)
    if age <= 20:
        if prediction == 1:
            pred = "You have Diabetes Type I, please consult a Doctor Soon...\n Take some insuline dosage to keep blood sugar level in a healthy range..."
        elif prediction == 0:
            pred = "You don't have Diabetes."
    else:
        if prediction == 1:
            pred = "You have Diabetes Type II, please consult a Doctor."
        elif prediction == 0:
            pred = "You don't have Diabetes !!!"

Ln: 13 Col: 46
```

Fig 7.1 RUNNING THE PYTHON FILE

After the python file is executed the port number of the web application should be copy and pasted in the web browser to see the output screen.



```
Python 3.10.2 (tags/v3.10.2:a58ebcc, Jan 17 2023, 14:12:15) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:\Users\Admin\Documents\diabetes prediction\app.py =====
* Serving Flask app 'app'
* Debug mode: off
[[31m[[mWARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.]]0m
* Running on all addresses (0.0.0.0)
* Running on http://127.0.0.1:5000
* Running on http://192.168.29.231:5000
[[33mPress CTRL+C to quit]]0m
192.168.29.231 - - [16/Mar/2023 15:58:16] "GET / HTTP/1.1" 200 -
192.168.29.231 - - [16/Mar/2023 15:58:17] "GET /static/css/style.css HTTP/1.1" 200 -
192.168.29.231 - - [16/Mar/2023 15:58:17] "GET /static/111.jpg HTTP/1.1" 200 -
192.168.29.231 - - [16/Mar/2023 15:58:17] "[33mGET /favicon.ico HTTP/1.1]]0m 404 -
```

Fig 7.2 COPYING THE PORT NUMBER FROM THE OUTPUT OF PYTHON FILE EXECUTED

## 7.3 RESULT

After pasting the port number in the web browser and give enter button the below web application will be running in it. Now we can give the required attribute data and predict the diabetes. By filling the attributes and clicking the “Predict” button given, the output will be displayed in the screen that the person is affected by the diabetes type1, type2 or not.

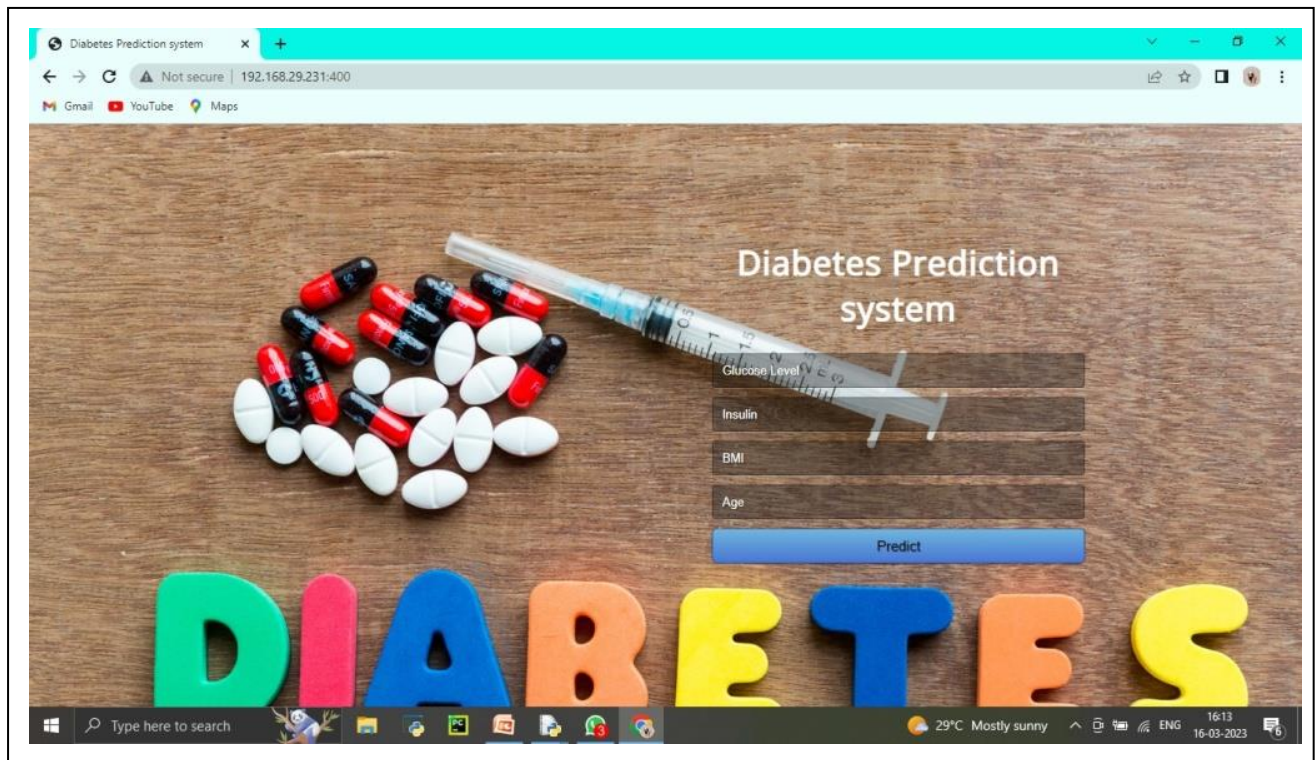


Fig 7.3 OUTPUT SCREEN 1

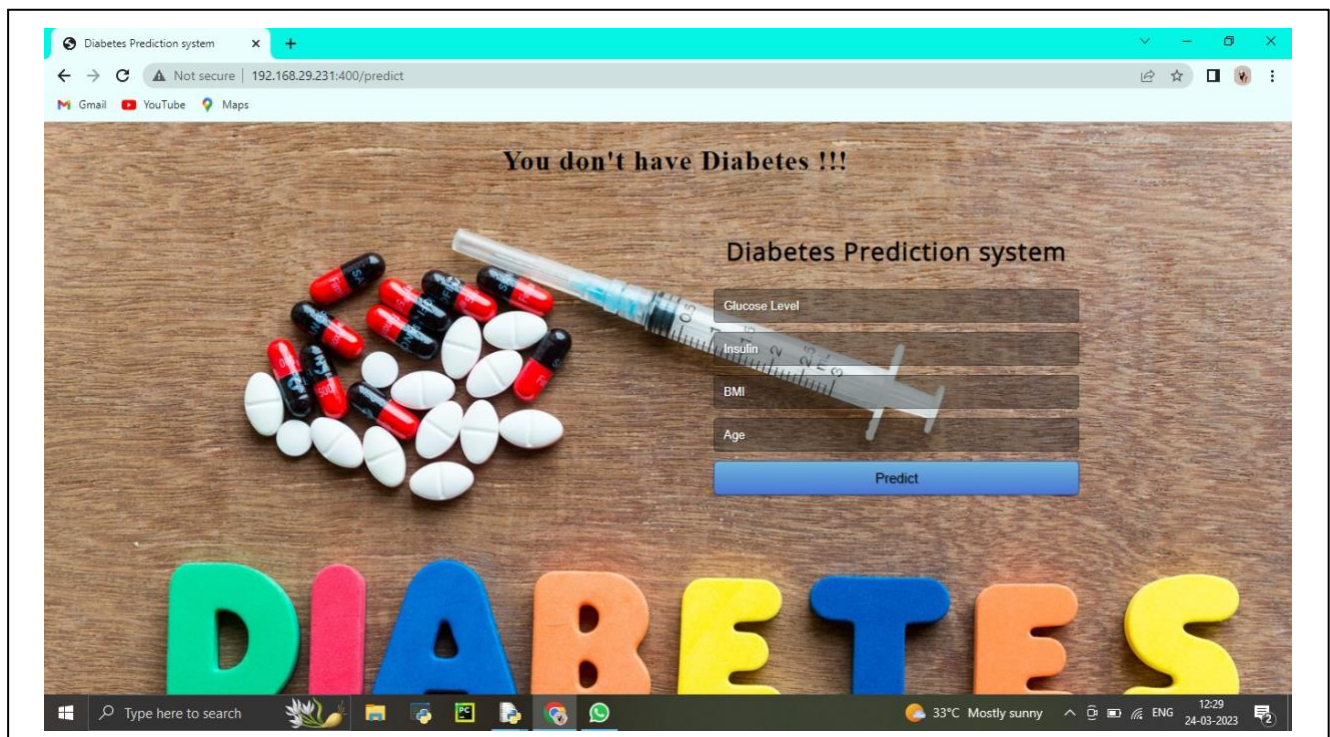


Fig 7.4 OUTPUT SCREEN 2





Fig 7.5 OUTPUT SCREEN 3

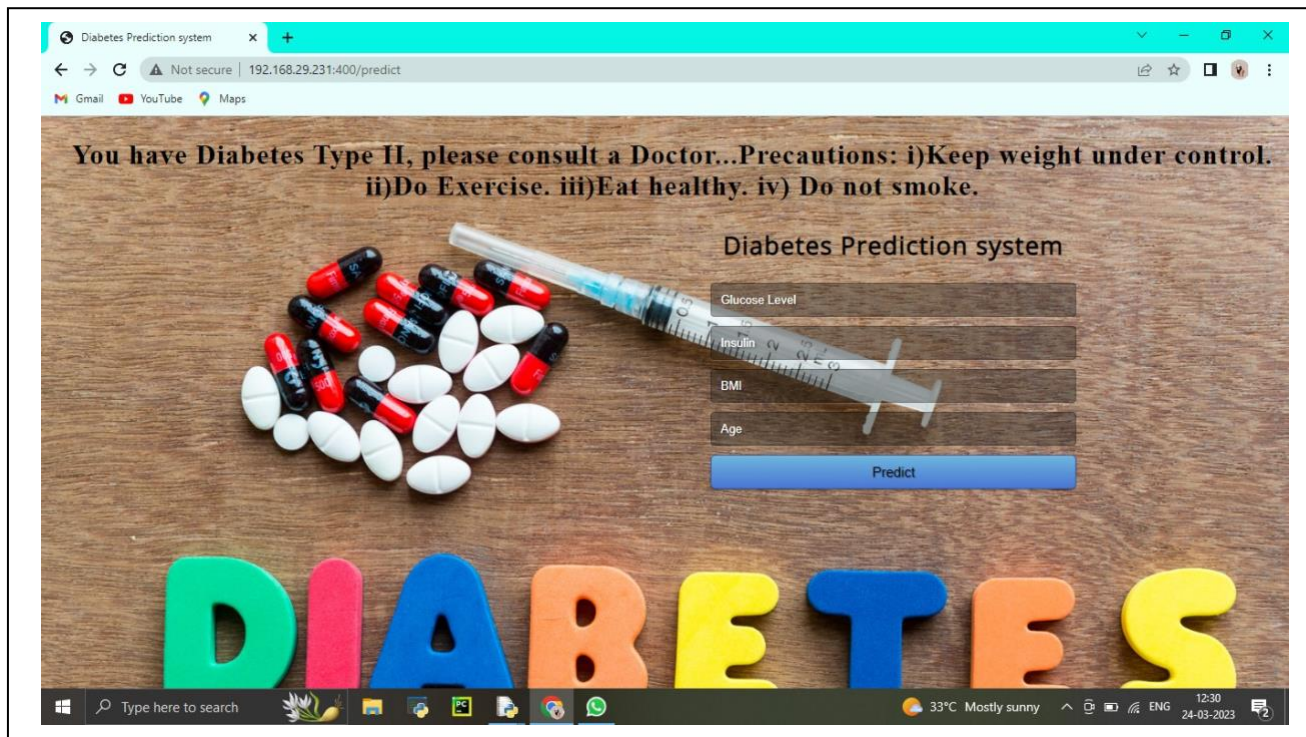


Fig 7.6 OUTPUT SCREEN 4

## **7.4 CONCLUSION**

This study proposes a machine learning-based decision support system for diabetes using decision layer fusion. Two commonly used machine learning techniques are integrated into the proposed model using fuzzy logic. The accuracy of the proposed fuzzy decision system is 92.7%, which is higher than other existing systems. With this diagnostic model, we can save many lives. In addition, diabetes mortality is manageable if the disease is diagnosed early and preventive measures are taken.

## REFERENCES

- [1] Random Forest Algorithm for the Prediction of Diabetes, Proceeding of International Conference on Systems & Computation Automation and Networking 2019 @IEEE 978-1-7281-1524-5.
- [2] F. Islam, R. Ferdousi, S. Rahman, and H. Y. Bushra, Computer Vision and Machine Intelligence in Medical Image Analysis. London, U.K.: Springer, 2019
- [3] A. Frank and A. Asuncion. (2010). UCI Machine Learning Repository. Accessed: Oct. 22, 2021. [Online]. Available: <http://archive.ics.uci.edu/ml>
- [4] G. Pradhan, R. Pradhan, and B. Khandelwal, “A study on various machine learning algorithms used for prediction of diabetes mellitus,” in Soft Computing Techniques and Applications (Advances in Intelligent Systems and Computing), vol. 1248. London, U.K.: Springer, 2021, pp. 553–561, doi: 10.1007/978-981-15-7394-1\_50.
- [5] S. Kumari, D. Kumar, and M. Mittal, “An ensemble approach for classification and prediction of diabetes mellitus using soft voting classifier,” Int. J. Cogn. Comput. Eng., vol. 2, pp. 40–46, Jun. 2021, doi:10.1016/j.ijcce.2021.01.001.
- [6] S. Saru and S. Subashree. Analysis and Prediction of Diabetes Using Machine Learning. Accessed: Oct. 22, 2022. [Online]. Available: [https://papers.ssrn.com/sol3/papers.cfm?abstract\\_id=3368308](https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3368308)
- [7] P. Sonar and K. Jaya Malini, “Diabetes prediction using different machine learning approaches,” in Proc. 3rd Int. Conf. Comput. Methodologies Commun. (ICCMC), Mar. 2019, pp. 367–371, doi:10.1109/ICCMC.2019.8819841
- [8] M. F. Faruque and I. H. Sarker, “Performance analysis of machine learning techniques to predict diabetes mellitus,” in Proc. Int. Conf. Electr., Comput. Commun. Eng. (ECCE), Feb. 2019, pp. 7–9, doi:10.1109/ECCE.2019.8679365.
- [9] B. Jain, N. Ranawat, P. Chittora, P. Chakrabarti, and S. Poddar, “A machine learning perspective: To analyze diabetes,” Mater. Today: Proc., pp. 1–5, Feb. 2021,
- [10] J. Liu, J. Feng, and X. Gao, “Fault diagnosis of rod pumping wells based on support vector machine optimized by improved chicken swarm optimization,” IEEE Access, vol. 7, pp. 171598–171608, 2019, doi:10.1109/ACCESS.2019.2956221.

## **APPENDICES**

# Diabetes Prediction System Using Machine Learning With Web App

HEMAVATHY J<sup>1a</sup>, LOKHITHA D<sup>2b</sup>, DURGA V<sup>3b</sup>, DEVIPRIYA S<sup>4b</sup>

<sup>a</sup> Assistant professor, Department of Information Technology, Panimalar Engineering College

<sup>b</sup> Student, Department of Information Technology, Panimalar Engineering College

[hemaramya27@gmail.com](mailto:hemaramya27@gmail.com)<sup>1</sup>, [lokhitharajan@gmail.com](mailto:lokhitharajan@gmail.com)<sup>2</sup>, [iamdurgavenkatesh@gmail.com](mailto:iamdurgavenkatesh@gmail.com)<sup>3</sup>,  
[devipriyasiva157@gmail.com](mailto:devipriyasiva157@gmail.com)<sup>4</sup>

## Abstract:

In the medical field, it is crucial to predict diseases in advance to help them. Diabetes is one of the most dangerous health conditions in the world. In an ultra-modern culture, sugar and fat are ubiquitous in our healthy habits, increasing the threat of diabetes. To predict complaints, it is important to understand their symptoms. Currently, machine learning (ML) algorithms are invaluable for complaint discovery. This compilation presents a model for predicting diabetes using a fusion machine literacy approach. The abstract framework consists of two classes of models: support vector machine (SVM) models and artificial neural network (ANN) models. These models dissect the data set to determine whether opinions about diabetes are positive or negative. The dataset used in this exploration is independently divided into training data and test data, with a ratio of 7030. The transactions of these models become the input class functions of the fuzzy models, and the feeling of fuzziness ultimately determines whether the perception of diabetes is positive or negative. The cloud storage system stores the fusion model for future use. Based on the real-time medical records of the cases, the fusion model predicts whether the cases are diabetic or not. The proposed ensemble ML model achieves a prediction accuracy of 92.7%, which is more advanced than the originally published method.

**Keywords:** Machine learning, Diabetes, Prediction, Accuracy, Regression, Decision tree

## 1. Introduction

Diabetes mellitus, commonly known as diabetes, is a metabolic disease in which diabetics experience blood sugar problems due to irregular production and release of insulin. It is also a long-term condition characterized by high blood sugar. It is one of the most serious diseases in the world and can have multiple consequences. Based on recent increases in incidence, the number of diabetes cases worldwide will reach 642 million by 2040, suggesting that one in 10 people will become ill. This daunting figure, no doubt, demands immediate attention. Type 1 and type 2 diabetes are the two most common forms. Type 1 diabetes can affect anyone at any age, but it most often affects teenagers and children. People with type 1 diabetes whose body produces little or no insulin and who receive daily insulin injections to control their blood sugar. Type 2 diabetes can strike anyone at any age, but is much more common in adults, accounting for over 90% of all diabetes cases. In type 2 diabetes, the body does not make good use of the insulin it produces. The basis of type 2 diabetes care is a healthy lifestyle, including increased physical activity and a balanced diet. On the other hand, adults with type 2 diabetes will eventually need medication or insulin to control their blood sugar. Another type of diabetes is gestational diabetes, which is characterized by high blood sugar levels during pregnancy and is associated with complications for both mother and child. Their children were more likely to develop type 2 diabetes later in life, although this decreased after pregnancy.

Machine learning algorithms are a great way to analyze large amounts of data and make recommendations based on that data. These algorithms are useful for studying data sets and predicting new input values. Various experimenters use machine learning algorithms to predict and control many conditions. Machine learning algorithms are particularly good at predicting color conditions. Apply machine learning algorithms to check their diabetes prediction range to take necessary action to avoid diabetes. Many experimenters use machine learning algorithms to predict and control color conditions.



The designed thing is whether a case is diabetic or not based on the diabetes dataset. From the medical biography, eight characteristics including number of pregnancies, insulin levels, glucose levels, blood pressure, skin firmness, body mass index (BMI), body function diabetic childbirth and age, could predict whether a patient was diabetic. These features are integrated with the XGBoost algorithm for diabetes prediction, which can provide performance as reliable as diagnosing diabetes. These estimates are based on symptoms seen in the early stages of diabetes and some physical illnesses.

## 2. Related Work

In [1], the authors state that diabetes could be the deadliest disease on earth. It's not just a disease, but a cause of everything from visual impairment to kidney failure and coronary heart attacks. The most difficult task for doctors is to quickly predict whether a patient has diabetes. Also, early prediction of disease requires treating patients before the condition worsens and becomes serious. The aim of this article is to propose an expert system that can predict if a patient is diabetic with high accuracy. Data mining can separate the hidden information from the vast amount of diabetes information available, which is a great help for diabetes research. The goal of this project is to develop a framework that can more accurately predict a patient's diabetes risk level using policy regression. The model developed using the artificial neural network has a total of six dense layers. Each of these layers is responsible for the effective functioning of the model. The model made predictions with an accuracy of 77%, which is quite good and reliable.

In [2], the authors claim that the growth of diabetes in humans is increasing exponentially. According to a health report, approximately 347 million people worldwide suffer from diabetes. Diabetes does not only affect the elderly, but also the younger generation. Early detection of diabetes is also a big challenge. Such detection will facilitate the decision-making process of the health system. Early prediction of diabetes helps us save human lives from diabetes. Long-term diabetes poses a risk of damage to vital organs in the body. Therefore, the early prediction of diabetes is very important to save human beings from diabetes. Data analysis is about finding patterns in large data sets. This helps us draw certain conclusions from the available data sets. The analysis process can be performed by different machine learning algorithms. This article presents two sets of machine learning methods to predict diabetes. One of them is a classification-based algorithm and the other is a hybrid algorithm. For classification, we used the random forest algorithm. For the hybrid approach, we chose the XGBoost algorithm. These two algorithms were implemented and compared to study the predictive accuracy of two different machine learning methods for diabetes and obtained an average score of 74.10%, outperforming the random forest algorithm.

In [3], the authors point out that remarkable advances in biotechnology and public health infrastructure have led to a significant generation of critical and sensitive health data. By applying smart data analysis techniques, very interesting models have been identified for early and early detection and prevention of various deadly diseases. Diabetes is an extremely deadly disease as it can lead to other deadly diseases viz.e) Heart, kidney and nerve damage. In this article, a machine learning-based approach is proposed for the classification, early recognition and prediction of diabetes. Additionally, it provides a hypothetical IoT-based diabetes monitoring system for a healthy and affected person to monitor their blood sugar (blood sugar) level. For the classification of diabetes, three different classifiers have been used, i.e) Random Forest (RF), Multilayer Perceptron (MLP) and Logistic Regression (LR). For predictive analysis, we used long-short-term memory (LSTM), moving average (MA), and linear regression (LR). For the experimental evaluation, the PIMA India Diabetes Reference Dataset is used. During the analysis, it was observed that MLP outperformed other classifiers by 86%.08% accuracy and LSTM significantly improved prediction with 87.26% accuracy in diabetes. In addition, a comparative analysis of the proposed method with the existing state of the art is performed, demonstrating the adaptability of the proposed method to many public health applications.

In [4], the authors state that diabetes is one of the most common diseases in the world. In recent years, many machine learning (ML) techniques have been used to predict diabetes. The increasing complexity of this problem has inspired researchers to explore a powerful set of deep learning (DL) algorithms. The highest accuracy achieved so far by the combined CNN-LSTM model is 95.1%. Although many ML algorithms have been used to solve this problem, there is a group of classifiers that are rarely or never used in this problem, so it is important to evaluate the performance of these classifiers in predicting diabetes. Also, there are no recent surveys examining and comparing the performance of all the proposed ML and DL techniques except the combined models. This article reviews all ML and DL techniques for the prediction of diabetes published in the last six years. Moreover, a study was developed to implement these rarely used and unused ML classifiers on the Indian Pima dataset to analyze their performance. The classifier achieved an accuracy of 68% to 74%. It is proposed to use these classifiers in the prediction of diabetes and to improve them by developing combined models.

In [5], the authors state that diabetes is a serious chronic disease that occurs when blood sugar exceeds a certain limit. In recent years, machine and deep learning techniques have been used to predict diabetes and its complications. However, researchers and

developers still face two major challenges when building predictive models for type 2 diabetes. First, there is considerable heterogeneity in previous studies on the techniques used, which makes it difficult to identification of the optimal technique. Second, the features used in the models lack clarity which reduces their interpretability. This systematic review aims to provide answers to the above challenges. The review mainly followed the PRISMA approach, with the addition of methods offered by Keele and Durham Universities. Ninety studies were included and reported model types, complementary techniques, data sets and performance parameters were extracted. Eighteen different model types were compared, and the tree-based algorithm showed the best performance. Although deep neural networks are capable of handling large and dirty data, they are not optimal. Balancing data and feature selection techniques can help improve model efficiency. Models trained on ordered datasets yield nearly perfect models.

## **2.1. Scope and motivation of the project:**

The system proposed in this project uses machine learning algorithms to find predictions of diabetic disease. We have many classification algorithms like Naive Bayes, SVM, Decision Trees, etc. In the future, we may add more algorithms to find the output and compare algorithms to find efficient ones. We can add a visitor query module where visitors can post queries to admins and admins can send responses to those queries. We can add a treatment module where doctors upload patient treatment details and patients can view those treatment details. Health problems are increasing dramatically these days. Many people start having full physical exams in their 40s and 50s. But our lifestyle has a big impact on our health and can lead to diabetes and other health problems.

Early detection of diabetes can prevent mortality. Since much of our healthcare industry is focused on early diagnosis of these diseases, using machine learning techniques we can detect diseases at an early stage and help cure them absolutely free. . Using an appropriate dataset, create a trained machine learning model that can diagnose a normal person and generate an output report of whether the person is affected by diabetes or not, and classifying the level of diabetes.

## **2.2. Existing system:**

The project asked several researchers to examine it in the field of diabetes using machine learning techniques. Extract knowledge from existing medical data. This predictive analytics model uses support vector machine algorithms, including logistic regression, naive Bayes, and support vector machines, where support vector machines provide better performing algorithms. In this work, we examine real diagnostic medical data against many risk factors using popular machine learning classification techniques to assess their predictive performance. The proposed fuzzy decision system achieves an accuracy rate of 94%.87, superior to other existing systems.

### **2.2.1. Disadvantages:**

i) Diabetes is one of the deadliest diseases in this world and it is increasing rapidly. It is not just a disease but the creator of different types of diseases.

ii) We chose these algorithms for this study after some initial experimentation where we found that these techniques are more efficient for this problem.

## **2.3. Proposed system:**

This project proposes a fusion model for the prediction of diabetes. The proposed model has two main steps. Our first step is to form layers, while the second step is to test layers. However, in our proposed model, we only use two commonly used ML algorithms. If our proposed model does not meet the learning requirements, it will be recycled. The second stage of the proposed framework is reflected by the test layer. The test layer gets the dataset from the medical database and loads the preprocessed training model from the cloud. In our proposed framework, the validation layer is related to the real-time diagnosis and classification of diabetes. The proposed ML fusion model can use real-time patient data as input and improve disease detection systems.

### **2.3.1 Advantages:**

i) Applicable to unstructured and semi-structured datasets such as images and text.

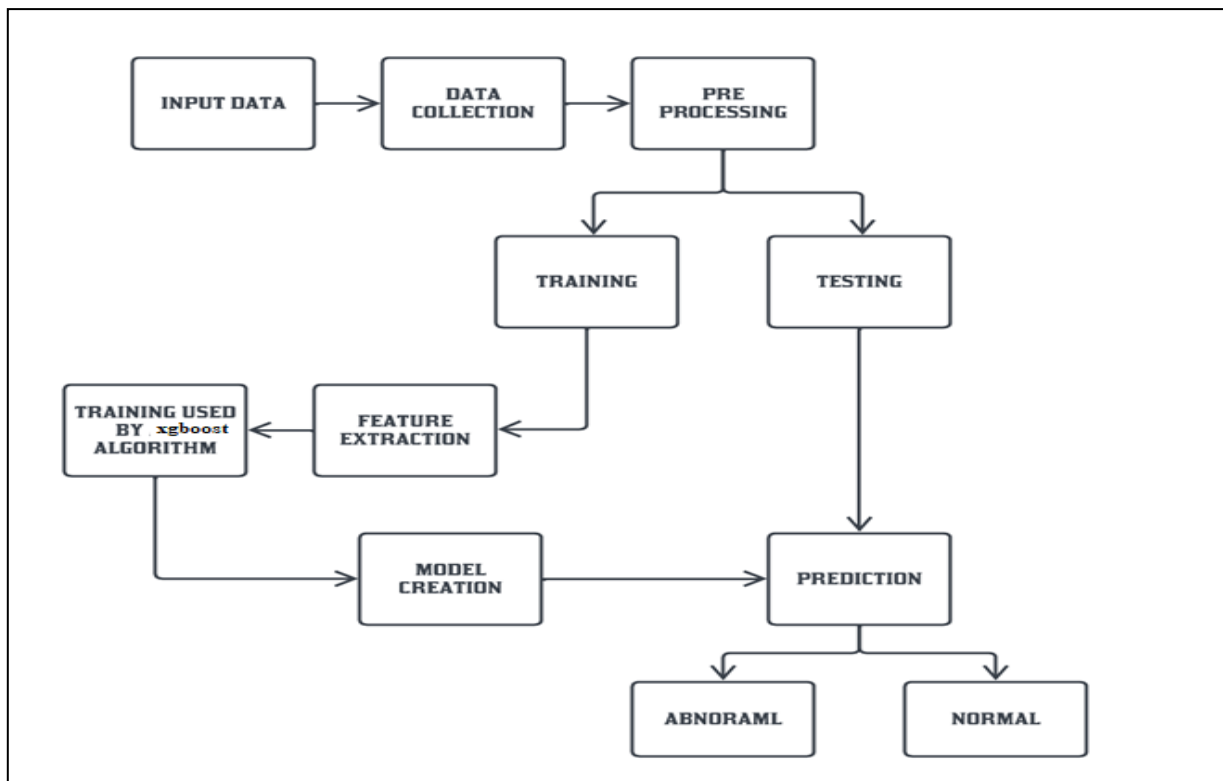
ii) Accurate and robust results can be obtained.

iii) Successfully used in medical applications.

### 3. Architecture diagram:

The proposed model has two main steps. Our first step is to form layers, while the second step is to test layers. However, in our proposed model, we only use two commonly used ML algorithms. If our proposed model does not meet the learning requirements, it will be recycled. The second stage of the proposed framework is reflected by the test layer. The test layer gets the dataset from the medical database and loads the preprocessed training model from the cloud. In our proposed framework, the validation layer is related to the real-time diagnosis and classification of diabetes. The proposed ML fusion model can use real-time patient data as input and improve disease detection systems.

The below block diagram is the flow of our proposed diabetics prediction system. The input data is collected from the source and go under preprocessing. And preprocessed data is first trained using the machine learning algorithm and the features are extracted and trained by the highest accuracy algorithm that is XGBoost. Then the created model is tested and compared using the machine learning algorithm. And finally the prediction is done and the output will be predicted as 0's and 1's.



**Fig1. Block diagram of the proposed diabetes prediction system**

#### 3.1. Dataset Description:

The diabetes data set we are using from <https://www.kaggle.com/johndasilva/diabetes>.

These diabetes dataset containing 2500 cases. The goal is to predict based on obtained measures to predict if the patient is affected by diabetes or not.

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	2	138	62	35	0	33.6	0.127	47	1
1	0	84	82	31	125	38.2	0.233	23	0
2	0	145	0	0	0	44.2	0.630	31	1
3	0	135	68	42	250	42.3	0.365	24	1
4	1	139	62	41	480	40.7	0.536	21	0

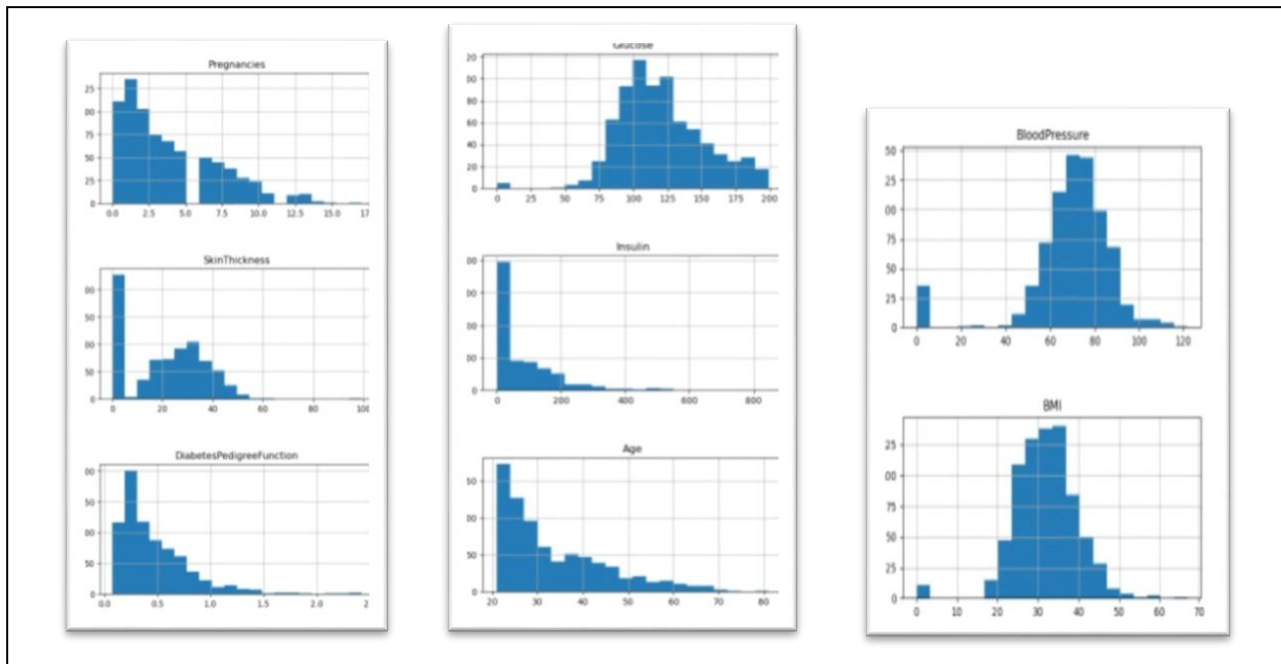
**Fig2.dataset used in training phase**

The diabetes data set consists of 2500 data, with 9 features each. “Outcome” is the feature we are going to predict, 0 means no diabetes, 1 means diabetes.

### 3.2. Data collection & pre-processing:

In the first step we collect data from reliable source. Glucose, Insulin, Blood pressure, Glucose pedigree function, Body mass index (BMI), Weight, Number of pregnancies and Age are some of the criteria in diabetes record set sample. Predicted outcome class, where '1' denotes a positive diabetes patient class and '0' denotes a negative diabetes patient class.

Pre-processing is next step. It's an important phase in data discovery process. The majority of health-care data contains gaps in value and inconsistencies. We apply Synthetic Minority Oversampling Technique (SMOTE) in this project, which is a well-known preprocessing approach for dealing with unbalanced datasets.



**Fig3. Graph diagram of the datasets loaded**

### 3.3. Feature Extraction:

Feature extraction is the process of converting raw data into digital features that can be processed while preserving information from the original dataset. This produces better results than simply applying machine learning to raw data. This is an important classification function.

### 3.4. Algorithms used in proposed system:

#### 3.4.1. K-Nearest neighbours Algorithm:

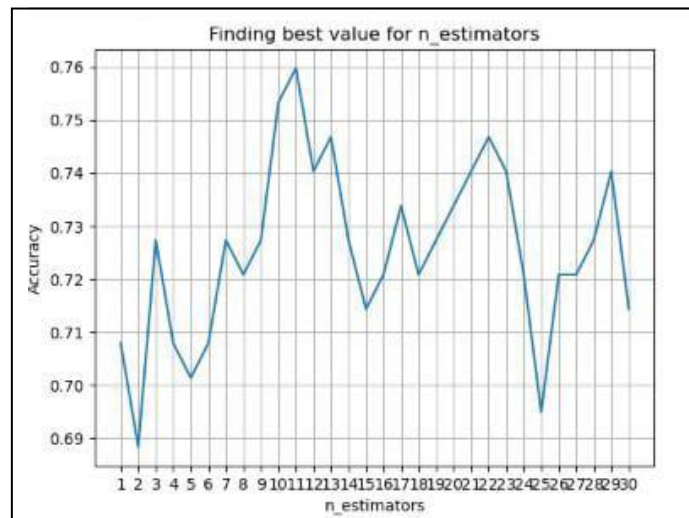
K can be kept odd so that we can calculate clear majorities when only two sets are possible (eg.Red and blue). As K increases, we get smoother and better defined boundaries between the different classes. As we increase the number of data points in the training set, the accuracy of the above classifier increases.

##### K-nearest neighbours Algorithm Steps:

1. Let  $m$  be the number of training data samples.
2. Let  $p$  be the unknown point. Store the training samples in the  $arr[]$  data point array.
3. This means that each element of the array represents a tuple  $(x, y)$  .for  $i = 0$  to  $m$ :

Calculate the Euclidean distance  $d$ .

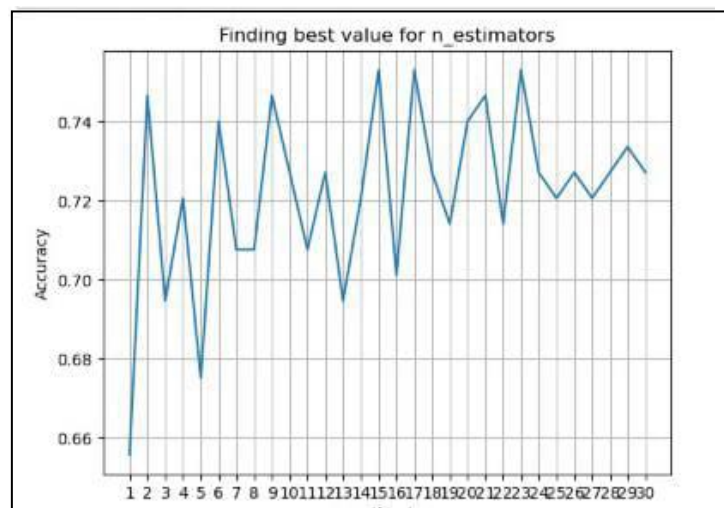
Obtain the set  $S$  of  $K$  shortest distances. Each of these distances corresponds to an already classified data point. Given the majority label under  $S$ .



The algorithm is used to predict behavior and outcomes in many industries, including banking and finance, e-commerce, and healthcare. It is increasingly used due to its ease of application, adaptability, and ability to perform classification and regression tasks.

#### Random forest Algorithm steps:

1. Randomly select K data points from the training set.
2. Build a decision tree associated with the selected data points (subset).
3. Select the number N for the decision tree to build.
4. Repeat steps 1 and 2.
5. For a new data point, find the prediction of each decision tree and assign the new data point to the category that won the majority vote.



XGBoost (eXtreme Gradient Boosting) is a popular and efficient open source implementation of the gradient boost tree algorithm. Gradient boosting is a supervised learning algorithm that attempts to accurately predict a target variable by combining ensemble estimates from a set of simpler, weaker models.

### XGBoost Algorithm Steps:

1. Make initial predictions and calculate residuals.
2. Create the XGBoost tree.
3. Prune the tree.
4. Calculate the output value of the sheet
5. Make a new prediction
6. Calculate the residuals using the new prediction.

Below output shows the comparisons of all four Machine Learning Algorithms that gives us an accuracy rates that XGBoost gives the higher rate of 92.7%.

```

In [22]: from sklearn.metrics import accuracy_score
accuracy_knn = accuracy_score(Y_test, Y_pred_knn)
accuracy_ranfor = accuracy_score(Y_test, Y_pred_ranfor)
accuracy_lgbm = accuracy_score(Y_test, Y_pred_lgbm)
accuracy_xgbr = accuracy_score(Y_test, Y_pred_xgbr)

In [23]: print("K Nearest neighbors: " + str(accuracy_knn * 100))
print("Random Forest: " + str(accuracy_ranfor * 100))
print("Light BGM: " + str(accuracy_lgbm * 100))
print("XG Boost: " + str(accuracy_xgbr * 100))

K Nearest neighbors: 72.07792207792207
Random Forest: 68.83116883116884
Light BGM: 70.12987012987013
XG Boost: 72.72727272727273

In [24]: from sklearn.metrics import confusion_matrix
cm = confusion_matrix(Y_test, Y_pred_xgbr)
cm

Out[24]: array([[83, 17],
               [25, 29]], dtype=int64)

In [25]: from sklearn.metrics import classification_report
print(classification_report(Y_test, Y_pred_xgbr))

              precision    recall  f1-score   support

     0.0       0.77      0.83      0.80       100
     1.0       0.63      0.54      0.58        54

   accuracy          0.70      0.68      0.69       154
  macro avg          0.70      0.68      0.69       154
 weighted avg          0.72      0.73      0.72       154

In [26]: import pickle
pickle.dump(xgbr, open('model.pkl', 'wb'))
model = pickle.load(open('model.pkl', 'rb'))

```

**Fig6. Comparison of all 4 algorithms used in the training dataset**

## 4. Result and discussion:

### 4.1. Training & testing:

During training, we train the machine from a data source. Test data is transformed and predicts accurate results. During training, machine learning automatically selects the correct training algorithm based on the target type specified in the training data source. The training dataset to validate with the test dataset model. After transforming the test data, the accurate prediction result is 92.7% is achieved by the XGBoost algorithm.

Training data set which will be validated using test dataset model. The test data is transformed and predicts accurate result is 92.7% is achieved by XGBoost algorithm.

#### 4.2. Prediction process:

This module predicts whether the user is diabetic or not using the XGBoost algorithm. And with the help of diabetes symptoms to predict normal levels of diabetes, type 1 and type 2. After training the model, we measured it using different parameters and accuracies in the dataset. Diabetes predictions achieved a 92.7% machine learning rate using XGBoost. In the future, this layered framework combined with machine learning algorithms can be used to predict or analyze various anomalies.

Additional ML calculations can be used to improve and enhance diabetes screening efforts. Training data set which will be validated using test dataset model. The test data is transformed and predicts accurate result is 92.7% is achieved by XGBoost algorithm.

#### 4.3. Proposed system Algorithm:

##### 4.3.1. XGBoost algorithm:

XG Boost can be used in a variety of applications, including Kaggle competitions, recommender systems, and click-through rate prediction. It is also customizable, and various model parameters can be tuned to optimize performance. XG Boost is an implementation of a gradient boost decision tree. The XG Boost model basically dominates a lot of Kaggle competition. In this algorithm, decision trees are built in sequential form. Weight plays an important role in XG Boost. All independent variables are assigned weights and entered into a decision tree that predicts an outcome. Variables that the tree predicts incorrectly are weighted and these variables are passed to the second decision tree. These individual classifiers/predictors are then combined to create a robust and accurate model. It can handle custom regression, classification, ranking, and prediction problems.

##### 4.3.2. The Mathematics of XGBoost:

Before learning the math of gradient boosting, here's a simple CART example to rate if someone likes the hypothetical PC game X. Tree example: Obtained by summing the assertion predictions for each individual decision tree. Looking at this example, the important thing is that the two trees try to complement each other. Mathematically, the model can be written as:

$$\hat{y}_i = \sum_{k=1}^K f_k(x_i), f_k \in \mathcal{F}$$

Equation (1)

where, K is number of trees, f is functional space of F, F is set of possible CARTs. The objective function for the above model is given by:

$$obj(\theta) = \sum_i^n l(y_i, \hat{y}_i) + \sum_{k=1}^K \Omega(f_k)$$

Equation (2)

where, first term is loss function and second is the regularization parameter. Now, Instead of learning tree all at once which makes optimization harder, we apply the additive strategy, minimize loss what we have learned and add a new tree which can be summarized below:

$$\begin{aligned} \hat{y}_i^{(0)} &= 0 \\ \hat{y}_i^{(1)} &= f_1(x_i) = \hat{y}_i^{(0)} + f_1(x_i) \\ \hat{y}_i^{(2)} &= f_1(x_i) + f_2(x_i) = \hat{y}_i^{(1)} + f_2(x_i) \\ &\dots \\ \hat{y}_i^{(t)} &= \sum_{k=1}^t f_k(x_i) = \hat{y}_i^{(t-1)} + f_t(x_i) \end{aligned}$$

Equation (3)

The objective function of above model can be defined as:



$$\begin{aligned}
obj^{(t)} &= \sum_{i=1}^n l(y_i, \hat{y}_i^{(t)}) + \sum_{i=1}^t \Omega(f_i) \\
&= \sum_{i=1}^n l(y_i, \hat{y}_i^{(t-1)} + f_t(x_i)) + \Omega(f_t) + constant \\
obj^{(t)} &= \sum_{i=1}^n (y_i - (\hat{y}_i^{(t-1)} + f_t(x_i)))^2 + \sum_{i=1}^t \Omega(f_i) \\
&= \sum_{i=1}^n [2(\hat{y}_i^{(t-1)} - y_i)f_t(x_i) + f_t(x_i)^2] + \Omega(f_t) + constant
\end{aligned}$$

Equation (4)

Now, let's apply Taylor series expansion up to second order:

$$obj^{(t)} = \sum_{i=1}^n [l(y_i, \hat{y}_i^{(t-1)}) + g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i)] + \Omega(f_t) + constant$$

Equation (5)

where  $g_i$  and  $h_i$  can be defined as:

$$\begin{aligned}
g_i &= \partial_{\hat{y}_i^{(t-1)}} l(y_i, \hat{y}_i^{(t-1)}) \\
h_i &= \partial_{\hat{y}_i^{(t-1)}}^2 l(y_i, \hat{y}_i^{(t-1)})
\end{aligned}$$

Equation (6)

Simplifying and removing constant:

$$\sum_{i=1}^n [g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i)] + \Omega(f_t)$$

Equation (7)

Now, we define regularization term, but first we need to define the model:

$$f_t(x) = w_{q(x)}, w \in R^T, q : R^d \rightarrow \{1, 2, \dots, T\}$$

Equation (8)

Here,  $w$  is vector of scores on leaves of tree,  $q$  is function assigning each data point to corresponding leaf, and  $T$  is number of leaves. The regularization term is then defined by:

$$\Omega(f) = \gamma T + \frac{1}{2} \lambda \sum_{j=1}^T w_j^2$$

Equation (9)

Now, our objective function becomes:

$$\begin{aligned} obj^{(t)} &\approx \sum_{i=1}^n [g_i w_{q(x_i)} + \frac{1}{2} h_i w_{q(x_i)}^2] + \gamma T + \frac{1}{2} \lambda \sum_{j=1}^T w_j^2 \\ &= \sum_{j=1}^T [(\sum_{i \in I_j} g_i) w_j + \frac{1}{2} (\sum_{i \in I_j} h_i + \lambda) w_j^2] + \gamma T \end{aligned}$$

Equation (10)

#### 4.3.3. Advantages of XGBoost:

**i) Performance:** XG Boost has a proven track record of high quality results in a variety of machine learning tasks, especially in Kaggle competitions where it has been a popular choice for winning solutions.

**ii) Scalability:** XG Boost is designed for efficient and scalable training of machine learning models, making it suitable for large datasets.

**iii) Tunability:** XG Boost has a wide range of hyper parameters that can be tuned to optimize performance, making it highly customizable.

**iv) Missing value handling:** XG Boost has built-in support for missing value handling, which can easily handle real data that often contains missing values.

**v) Interpretability:** Unlike some machine learning algorithms that are difficult to interpret, XG Boost provides feature importance, which provides insight into which variables are most important for making predictions.

#### 4.4. Model Creation:

The XGBoost Algorithm is used in this project. It's a classification and regression supervised machine learning algorithm. Every single accessible example is categorized according to XGBoost calculation. A case is assigned to the class with the most votes from classifier with the case being relegated to the Random forest class with most votes calculated using a separation function. Examining the set of information data determines the estimation.

#### 4.5. Prediction:

This module predicts that the user is suffering from diabetic or not using XG Boost algorithm. And also predict diabetic level normal or high using diabetic symptoms. After training model we had measured with different parameters within datasets and accuracy.

Diabetes Prediction Using Machine learning rate of XGBoost with 92.7% is achieved. In future, this hierarchical framework combined with machine learning algorithms could be used to predict or analyses various disorders. Other ML computations can be used to enhance and improves job for diabetes examination.

#### 5. Proposed system Implementation:

After training the data sets the prediction process is done and the output will displayed in the web app by running the python code which is giving the final output by using the python flask and many more functions used in it.

```
app.py - C:\Users\Admin\Documents\diabetes prediction\app.py (3.10.2)
File Edit Format Run Options Window Help

import numpy as np
import pandas as pd
from flask import Flask, request, jsonify, render_template
import pickle

app = Flask(__name__)
model = pickle.load(open('model.pkl', 'rb'))

dataset = pd.read_csv('diabetes.csv')
dataset_X = dataset.iloc[:, [1, 2, 5, 7]].values

from sklearn.preprocessing import MinMaxScaler
sc = MinMaxScaler(feature_range = (0,1))
dataset_scaled = sc.fit_transform(dataset_X)

@app.route('/')
def home():
    return render_template('index.html')

@app.route('/predict', methods=['POST'])
def predict():
    float_features = [float(x) for x in request.form.values()]
    final_features = [np.array(float_features)]
    prediction = model.predict( sc.transform(final_features) )

    age=request.form['age']
    age=int(age)
    #print(age)
    if age <= 20:
        if prediction == 1:
            pred = "You have Diabetes Type I, please consult a Doctor Soon...\n Take some insuline dosage to keep blood sugar level in a healthy range..!"
        elif prediction == 0:
            pred = "You don't have Diabetes."
    else:
        if prediction == 1:
            pred = "You have Diabetes Type II, please consult a Doctor."
        elif prediction == 0:
            pred = "You don't have Diabetes !!!"

Ln: 13 Col: 46
```

**Fig7.Running the python file**

After the python file is executed the port number of the web application should be copy and pasted in the web browser to see the output screen.

```
"IDLE Shell 3.10.2"
File Edit Shell Debug Options Window Help

Python 3.10.2 (tags/v3.10.2:a58ebcc, Jan 17 2022, 14:12:15) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.

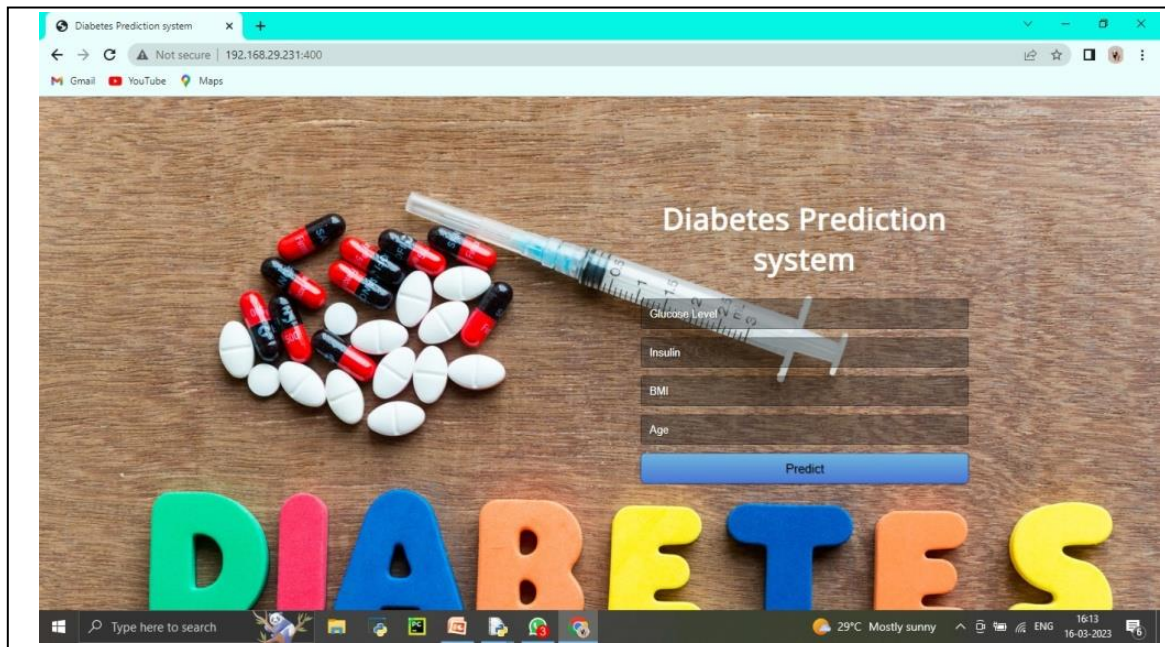
>>>
===== RESTART: C:\Users\Admin\Documents\diabetes prediction\app.py =====
* Serving Flask app 'app'
* Debug mode: off
[31m[!mWARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.[0m
* Running on all addresses (0.0.0.0)
* Running on http://127.0.0.1:4000
* Running on http://192.168.29.231:4000
[33mPress CTRL+C to quit[0m
192.168.29.231 - - [16/Mar/2023 15:58:16] "GET / HTTP/1.1" 200 -
192.168.29.231 - - [16/Mar/2023 15:58:17] "GET /static/css/style.css HTTP/1.1" 200 -
192.168.29.231 - - [16/Mar/2023 15:58:17] "GET /static/111.jpg HTTP/1.1" 200 -
192.168.29.231 - - [16/Mar/2023 15:58:17] "[33mGET /favicon.ico HTTP/1.1[0m" 404 -

Ln: 10 Col: 39
```

**Fig8.Copying the port number from the output of python file executed**

### 5.1.Result:

After pasting the port number in the web browser and give enter button the below web application will be running in it. Now we can give the required attribute data and predict the diabetes. By filling the attributes and clicking the “Predict” button given, the output will be displayed in the screen that the person is affected by the diabetes type1, type2 or not.



**Fig9.Result screen of the output**

### 5.2. Conclusion:

This study proposes a machine learning-based decision support system for diabetes using decision layer fusion. Two commonly used machine learning techniques are integrated into the proposed model using fuzzy logic. The accuracy of the proposed fuzzy decision system is 92.7%, which is higher than other existing systems. With this diagnostic model, we can save many lives. In addition, diabetes mortality is manageable if the disease is diagnosed early and preventive measures are taken.

### References:

- [1]Random Forest Algorithm for the Prediction of Diabetes, Proceeding of International Conference on Systems 6 Computation Automation and Networking 2019 @IEEE 978-1-7281-1524-5.
- [2] F. Islam, R. Ferdousi, S. Rahman, and H. Y. Bushra, Computer Vision andMachine Intelligence in Medical Image Analysis. London, U.K.: Springer,2019
- [3]A. Frank and A. Asuncion. (2010). UCI Machine Learning Repository.Accessed: Oct. 22, 2021.[Online]. Available: <http://archive.ics.uci.edu/ml>
- [4]G. Pradhan, R. Pradhan, and B. Khandelwal, “A study on various machinelearning algorithms used for prediction of diabetes mellitus,” in SoftComputing Techniques and Applications (Advances in Intelligent Systemsand Computing), vol. 1248. London, U.K.: Springer, 2021, pp. 553–561,doi: 10.1007/978-981-15-7394-1\_50.

- [5] S. Kumari, D. Kumar, and M. Mittal, “An ensemble approach for classification and prediction of diabetes mellitus using soft voting classifier,” *Int. J. Cogn. Comput. Eng.*, vol. 2, pp. 40–46, Jun. 2021, doi:10.1016/j.ijcce.2021.01.001.
- [6] S. Saru and S. Subashree, “Analysis and Prediction of Diabetes Using Machine Learning,” Accessed: Oct. 22, 2022, [Online]. Available: [https://papers.ssrn.com/sol3/papers.cfm?abstract\\_id=3368308](https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3368308)
- [7] P. Sonar and K. JayaMalini, “Diabetes prediction using different machine learning approaches,” in *Proc. 3rd Int. Conf. Comput. Methodologies Commun. (ICCMC)*, Mar. 2019, pp. 367–371, doi:10.1109/ICCMC.2019.8819841
- [8] M. F. Faruque and I. H. Sarker, “Performance analysis of machine learning techniques to predict diabetes mellitus,” in *Proc. Int. Conf. Electr., Comput. Commun. Eng. (ECCE)*, Feb. 2019, pp. 7–9, doi:10.1109/ECACE.2019.8679365.
- [9] B. Jain, N. Ranawat, P. Chittora, P. Chakrabarti, and S. Poddar, “A machine learning perspective: To analyze diabetes,” *Mater. Today: Proc.*, pp. 1–5, Feb. 2021,
- [10] J. Liu, J. Feng, and X. Gao, “Fault diagnosis of rod pumping wells based on support vector machine optimized by improved chicken swarm optimization,” *IEEE Access*, vol. 7, pp. 171598–171608, 2019, doi:10.1109/ACCESS.2019.2956221.