A Project Report

On

MACHINE LEARNING FOR DIABETES PREDICTION

Submitted in partial fulfillment for the award of the degree of

Bachelor of Technology

in

COMPUTER SCIENCE AND ENGINEERING

Submitted by

T. BALAJI	**194E1A0504**
K.M. THANUJA	**194E1A0553**
G. JANANI	**194E1A0524**
D. HIMANTH	**194E1A0521**
R. POOJITH	**194E1A0537**

Under the esteemed guidance of

Mr. M. Manivannan, M.E.,

Associate Professor

Dept. of C.S.E.

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

SIDDARTHA INSTITUTE OF SCIENCE AND TECHNOLOGY

(AUTONOMOUS)

(Approved by A.I.C.T.E., New Delhi Affiliated to J.N.T.U. Anantapur, Ananthapuramu.)

(Accredited by NBA (EEE, ME, ECE & CSE) & NAAC with 'A' Grade)

Siddharth Nagar, Narayanavanam Road, Puttur– 517 583, Chittoor District

2022-2023

SIDDARTHA INSTITUTE OF SCIENCE AND TECHNOLOGY (AUTONOMOUS)

(Approved by A.I.C.T.E., New Delhi Affiliated to J.N.T.U. Anantapur, Ananthapuramu.)
(Accredited by NBA (EEE, ME, ECE & CSE) & NAAC with 'A' Grade)
Siddharth Nagar, Narayanavanam Road, Puttur-517583

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Certificate

This is to certify that the Project entitled
"MACHINE LEARNING FOR DIABETES PREDICTION"
that is being submitted by

T. BALAJI	**194E1A0504**
K.M. THANUJA	**194E1A0553**
G. JANANI	**194E1A0524**
D. HIMANTH	**194E1A0521**
R. POOJITH	**194E1A0537**

in partial fulfillment of the requirements for the award of **BACHELOR OF TECHNOLOGY** in **Computer Science and Engineering** to **JNTUA, Ananthapuramu**. This project work or part thereof has not been submitted to any other University or Institute for the award of any degree.

Guide **Head of the Department**

Submitted for the University Examination held on _____

INTERNAL EXAMINER **EXTERNAL EXAMINER**

Acknowledgement

An endeavor of a long period can be successful only with the advice of many well-wishers. We take this opportunity to express our deep gratitude and appreciationto all those who encouraged us for successful completion of the project work.

We wish to express my sincere thanks to **Dr. K. ASHOK RAJU**., **Ph.D**., ***Chairman*** of Siddartha Institute of Science and Technology, Puttur, for providing ample facilities to complete the project work.

Our special thanks to **Dr. M. JANARDHANA RAJU, M.E., Ph.D**., **Principal,** Siddartha Institute of Science and Technology, Puttur, for his wonderful support during the progress of project work.

We are very much thankful to **Dr. M. A. MANIVASAGAM, M.E, Ph.D., Head of the Department,** Computer Science and Engineering, Siddartha Institute of Science and Technology, Puttur, for his valuable suggestions in completion of the project work.

We owe our deep gratitude to our project guide **Mr. M. MANIVANNAN, M.E., Associate Professor,** Department of Computer Science and Engineering, Siddartha Institute of Science and Technology, Puttur., who took keen interest on our project work and guided us all along, till the completion of our project work by providing all the necessary information for developing a good system.

Finally, we would like to express sincere thanks to our project coordinator **Mrs. R. PRIYADARSHINI, M.Tech., (Ph.D)., Associate Professor** and **D.R.C (Departmental Review Committee) members, faculty members of C.S.E** and Lab technicians, one and all who have helped us to complete the project work successfully.

As a gesture of respect towards our ***Family Members, friends*** and their supportwe wholeheartedly dedicate this work.

TABLE OF CONTENTS

Chapter No.	Title	Page No.
	ABSTRACT	**i**
	LIST OF FIGURES	**ii**
	LIST OF TABLES	**iii**
1.	**INTRODUCTION**	
	1.1 Domain Description	1
	1.2 About the Project	5
2.	**LITERATURE SURVEY**	7
3.	**SYSTEM ANALYSIS**	
	3.1 Problem Statement	17
	3.2 Problem Description	17
	3.3 Existing System	17
	3.4 Disadvantages of Existing System	18
	3.5 Proposed System	18
	3.6 Advantages of Proposed System	18
4.	**SYSTEM REQUIREMENTS**	
	4.1 Hardware Requirements	19
	4.2 Software Requirements	19
	4.3 Feasibility Study	19

Chapter No.	Title	Page No.

5. SYSTEM DEVELOPMENT

5.1 SYSTEM DESIGN

5.1.1 System Architecture 21
5.1.2 Modules 21
5.1.2.1 User 22
5.1.2.2 Diabetes Care 22
5.1.3 Algorithm 22
5.1.4 UML Diagrams 24
5.1.4.1 Introduction to UML 24
5.1.4.2 Class Diagram 26
5.1.4.3 Use case Diagram 27
5.1.4.4 Sequence Diagram 28
5.1.4.5 Activity Diagram 29
5.1.4.6 Deployment Diagram 30

5.2 SYSTEM IMPLEMENTATION

5.2.1 Software Description 30
5.2.2 Hardware Description 41
5.2.3 Database Description 42
5.2.4 Sample Code 43

5.3 SYSTEM TESTING

5.3.1 Software Testing 49
5.3.2 Types of Testing 49
5.3.2.1 Unit Testing 50
5.3.2.2 Integration Testing 51

Chapter No.	Title	Page No.

5.3.2.3 Acceptance Testing 51

5.3.3 Test Cases 52

6. RESULTS

6.1 Execution Procedure 54

6.2 Screen Shots 54

7. CONCLUSION AND FUTURE ENHANCEMENT

7.1 Conclusion 62

7.2 Future Enhancement 62

REFERENCES

ABSTRACT

Nowadays Poor lifestyle, diet, and work pressure lead the diabetes disease which may cause several fatal health issues like heart attack, strokes, kidney failure, etc. Diabetes mellitus can be effectively managed when caught early with high accuracy. Machine Learning approaches are very effective in the early detection and prediction of diabetes. The goal of this system is to offer the inclusive examination of the diagnosis of diabetes by Machine learning algorithms. Machine learning algorithms such as Support vector machine (SVM), linear discriminant analysis (LDA), K-means, and Decision tree-based algorithm such as the AdaBoost algorithm have predicted diabetes with high accuracy.

Keywords: Diabetes Prediction, Machine Learning, Decision Tree, Linear discriminant analysis, K-means, and Support Vector Machine.

LIST OF FIGURES

Figure No.	Figure Name	Page No.
5.1	System Architecture	21
5.2	Class Diagram	26
5.3	Use Case Diagram	27
5.4	Sequence Diagram	28
5.5	Activity Diagram	29
5.6	Deployment Diagram	30
6.2	Screenshots	54

LIST OF TABLES

Table No.	Table Name	Page No.
5.1	Test Case 1	52
5.2	Test Case 2	53

CHAPTER 1
INTRODUCTION

1.1 DOMAIN DESCRIPTION

Machine Learning

Machine learning algorithms learn, but it's often hard to find a precise meaning for the term learning because different ways exist to extract information from data, depending on how the machine learning algorithm is built. Generally, the learning process requires huge amounts of data that provide an expected response given particular inputs. Each input/response pair represents an example and more examples make it easier for the algorithm to learn. That's because each input/response pair fits within a line, cluster, or other statistical representation that defines a problem domain.

Machine learning is the act of optimizing a model, which is a mathematical, summarized representation of data itself, such that it can predict or otherwise determine an appropriate response even when it receives input that it hasn't seen before. The more accurately the model can come up with correct responses, the better the model has learned from the data inputs provided. An algorithm fits the model to the data, and this fitting process is training.

Features of Machine Learning

- Machine learning uses data to detect various patterns in a given dataset.
- It can learn from past data and improve automatically.
- It is a data-driven technology.

Need For Machine Learning

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

We can train machine learning algorithms by providing them a huge amount of data and let them explore the data, construct the models, and predict the required output automatically. The performance of the machine learning algorithm depends on the

amount of data, and it can be determined by the cost function. With the help of machine learning, we can save both time and money.

The importance of machine learning can be easily understood by its use cases, currently, machine learning is used in self-driving cars, cyber fraud detection, face recognition, and friend suggestion by Facebook, etc. Various top companies such as Netflix and Amazon have built machine learning models that are using a vast amount of data to analyze the user interest and recommend products accordingly.

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

- Rapid increment in the production of data
- Solving complex problems, which are difficult for a human
- Decision-making in various sector including finance
- Finding hidden patterns and extracting useful information from data.

Classification of Machine Learning

Machine learning offers a number of different ways to learn from data. Depending on your expected output and on the type of input you provide, you can categorize algorithms by learning style. The style you choose depends on the sort of data you have and the result you expect. The four learning styles used to create algorithms are:

- Supervised machine learning
- Unsupervised machine learning
- Semi-supervised machine learning
- Reinforcement machine learning

Supervised Machine Learning

When working with supervised machine learning algorithms, the input data is labeled and has a specific expected result. You use training to create a model that an algorithm fits to the data. As training progresses, the predictions or classifications become more accurate.

Types of Supervised Machine Learning Algorithms

The supervised learning algorithm can be further categorized into two types of problems:

- **Regression**

Regression algorithms are used if there is a relationship between the input variable and the output variable. It is used for the prediction of continuous variables, such as Weather forecasting, Market Trends, etc. Below are some popular Regression algorithms which come under supervised learning:

- Linear Regression
- Regression Trees
- Non-Linear Regression
- Bayesian Linear Regression
- Polynomial Regression

- **Classification**

Classification algorithms are used when the output variable is categorical, which means there are two classes such as Yes-No, Male-Female, True-false, etc.

- Random Forest
- Decision Trees
- Logistic Regression
- Support Vector Machines

Unsupervised Machine Learning

When working with unsupervised machine learning algorithms, the input data isn't labeled and the results aren't known. In this case, analysis of structures in the data produces the required model. The structural analysis can have a number of goals, such as reducing redundancy or grouping similar data.

Types of Unsupervised Learning Algorithms

The unsupervised learning algorithm can be further categorized into two types of problems:

- **Clustering**

Clustering is a method of grouping the objects into clusters such that objects with most similarities remains into a group and has less or no similarities with the objects of another group. Cluster analysis finds the commonalities between the data objects and categorizes them as per the presence and absence of those commonalities.

- **Association**

An association rule is an unsupervised learning method that is used for finding the relationships between variables in a large database. It determines the set of items

that occurs together in the dataset. The Association rule makes marketing strategy more effective. Such as people who buy X items (suppose a bread) are also tend to purchase Y (Butter/Jam) items. A typical example of the Association rule is Market Basket Analysis.

Unsupervised Learning Algorithms

Below is the list of some popular unsupervised learning algorithms:

- K-means clustering
- KNN (k-nearest neighbors)
- Hierarchal clustering
- Anomaly detection
- Neural networks
- Principle component analysis
- Independent component analysis
- Apriori algorithm
- Singular value decomposition

Like unsupervised machine learning, self-supervised learning requires no data labeling. However, unsupervised learning focuses on data structure that is, patterns within the data. Therefore, you don't use self-supervised learning for tasks such as clustering, grouping, dimensionality reduction, recommendation engines, or the like.

Semi-Supervised Machine Learning

A semi-supervised learning solution works like an unsupervised learning solution in that it looks for data patterns. However, semi-supervised learning relies on a mix of labeled and unlabeled data to perform its tasks faster than is possible using strictly unlabeled data. Self-supervised learning never requires labels and uses context to perform its task, so it would actually ignore the labels when supplied.

Reinforcement Machine Learning

You can view reinforcement learning as an extension of self-supervised learning because both forms use the same approach to learning with unlabeled data to achieve similar goals. However, reinforcement learning adds a feedback loop to the mix. When a reinforcement learning solution performs a task correctly, it receives positive feedback, which strengthens the model in connecting the target inputs and output.

Reinforcement learning is an area of Machine Learning. It is about taking suitable action to maximize reward in a particular situation. It is employed by various software and machines to find the best possible behavior or path it should take in a specific situation.

1.2 ABOUT THE PROJECT

Diabetes is one of the most dangerous chronic diseases that could lead to others serious complicating diseases. Diabetes diseases are also called as diabetes mellitus, which describes a set of metabolic disease. Diabetes leads to many other kinds of diseases and that are- heart attack, blindness, kidney diseases and so on. Diabetes is also called as Diabetes Mellitus is a chronic disease and is considered as one of the deadliest diseases. Diabetes disease can be categorized as Type 1 or Type 2. If the pancreas does not create adequate amount of insulin in body, is called as Type 1. In Type 2, the body either cannot effectively use the insulin that it produces or an inadequate amount of insulin is released into the bloodstream Type 1 disease generally occurs in children and adolescents, but it can occur in older people. Type 2 diabetes is normally milder compare to people have type 2 diabetes. Type 1 diabetes can be cured by inserting insulin into the fatty tissue under the skin of patient. However, Type 2 diabetes can be cured by having a healthy diet, weight and exercising. Many of diseases can be prevented if diabetes can be diagnosed in the early stages. Early diagnosis and prediction of disease is possible due to recent technological development of IOT, Artificial Intelligence (AI) and Block chain in the current healthcare system. AI presented a paradigm shift in diabetes care from conservative management approaches to construct the targeted data-driven precision care. IOT offers connected environment to the smart healthcare system. ML and deep learning are AI based techniques. ML has a potential of improving efficiency and decrease the cost of treatment in the healthcare system. Various texts are available for diagnosis and prediction of diabetes based on data mining and ML. Data mining and ML methods are equally important to their specific objective. Data mining techniques are useful to extract rules and pattern from the vast amount of diabetes data set, while ML is significant to learn and automate the machine along with pattern recognition. Several ML techniques are used to form digital support in diabetes care.

We review the several ML techniques for diabetes detection and prediction. There are mainly two categories of learning i.e. supervised and unsupervised learning

that made foremost impacts in the detection, prediction and treatment of diabetes. Supervised learning algorithms take direct feedback for the prediction. Supervised learning can be categorized in classification and regression methods. Are some popular algorithm of supervised learning? The basic objective of classification techniques is to detect and predict of the possibility of diabetes in patients with maximum accuracy. National Institute of Diabetes and Digestive Kidney Disease dataset and many techniques like Data transformation, Association rule mining is also used in. In this study, clustering techniques are used to predict diabetes with best accuracy.

CHAPTER 2
LITERATURE SURVEY

[1] Blockchain-Enabled Healthcare System for Detection of Diabetes

Blockchain has penetrated numerous domains such as, industries, government agencies, online voting, and healthcare, etc. Among these domains, healthcare is one of the trending and most important one, which consists of a control system and an Electronic Health Records (EHRs). Diabetes is one of the most rapidly growing chronic diseases that increases the death ratio across the globe. This paper presents a Blockchain-enabled diabetes disease detection framework that provides an earlier detection of this disease by using various machine learning classification algorithms and maintains the EHRs of the patients in a secure manner. Our EHRs sharing framework combines symptom-based disease prediction, Blockchain, and interplanetary file system (IPFS) in which the patient's health information are collected via wearable sensor devices. This information is then sent to EHRs manager, where an ML model is executed for further processing to collect the desired results. The results along with the physiological parameters are then stored in the Blockchain with the approval of concerned patient and his/her practitioner. It is anticipated that our proposed system will help the healthcare society in order to store, process, and share the patient health information in a secure manner.

[2] An Ensemble Approach for Classification and Prediction of Diabetes Mellitus Using Soft Voting Classifier

Diabetes is a dreadful disease identified by escalated levels of glucose in the blood. Machine learning algorithms help in identification and prediction of diabetes at an early stage. The main objective of this study is to pre- dict diabetes mellitus with better accuracy using an ensemble of machine learning algorithms. The Pima Indians Diabetes dataset has been considered for experimentation, which gathers details of patients with and without having diabetes. The proposed ensemble soft voting classier gives binary classification and uses the ensemble of three machine learning algorithms viz. random forest, logistic regression, and Naive Bayes for the classification. Empirical evaluation of the proposed methodology has been conducted with state-of-the-art methodologies and base classifiers such as AdaBoost, Logistic regression, Support

vector machine, Random forest, Naïve bayes, Bag-ging, Gradient Boost, XGBoost, AdaBoost. By taking accuracy, precision, recall, and F1-score as the evaluation criteria. The proposed ensemble approach gives the highest accuracy, precision, recall, and F1_score value with 79.04%, 73.48%, 71.45% and 80.6% respectively on the PIMA diabetes dataset. Further, the efficiency of the proposed methodology has also been compared and analyzed with breast cancer dataset. The proposed ensemble soft voting classier has given 97.02% accuracy on the breast cancer dataset.

[3] Cloud-Based Diabetes Decision Support System Using Machine Learning Fusion

Diabetes mellitus, generally known as diabetes, is one of the most common diseases worldwide. It is a metabolic disease characterized by insulin deficiency, or glucose (blood sugar) levels that exceed 200 mg/dl (11.1 ml/l) for prolonged periods, and may lead to death if left uncontrolled by medication or insulin injections. Diabetes is categorized into two main types—type 1 and type 2—both of which feature glucose levels above "normal," defined as 140 mg/dl. Diabetes is triggered by malfunction of the pancreas, which releases insulin, a natural hormone responsible for controlling glucose levels in blood cells. Diagnosis and comprehensive analysis of this potentially fatal disease necessitate application of techniques with minimal rates of error. The primary purpose of this research study is to assess the potential role of machine learning in predicting a person's risk of developing diabetes. Historically, research has supported the use of various machine algorithms, such as naïve Bayes, decision trees, and artificial neural networks, for early diagnosis of diabetes.

[4] Detection and Prediction of Diabetes Using Data Mining: A Comprehensive Review

Diabetes is one of the most rapidly growing chronic diseases, which has affected millions of people around the globe. Its diagnosis, prediction, proper cure, and management are crucial. Data mining based forecasting techniques for data analysis of diabetes can help in the early detection and prediction of the disease and the related critical events such as hypo/hyperglycemia. Numerous techniques have been developed in this domain for diabetes detection, prediction, and classification. In this paper, we present a comprehensive review of the state-of-the-art in the area of diabetes diagnosis and prediction using data mining. The aim of this paper is twofold; firstly, we explore

and investigate the data mining based diagnosis and prediction solutions in the field of glycemic control for diabetes. Secondly, in the light of this investigation, we provide a comprehensive classification and comparison of the techniques that have been frequently used for diagnosis and prediction of diabetes based on important key metrics. Moreover, we highlight the challenges and future research directions in this area that can be considered in order to develop optimized solutions for diabetes detection and prediction.

[5] Diabetes Detection Using Deep Learning Techniques with Oversampling and Feature Augmentation

Diabetes is a chronic pathology which is affecting more and more people over the years. It gives rise to a large number of deaths each year. Furthermore, many people living with the disease do not realize the seriousness of their health status early enough. Late diagnosis brings about numerous health problems and a large number of deaths each year so the development of methods for the early diagnosis of this pathology is essential. In this paper, a pipeline based on deep learning techniques is proposed to predict diabetic people. It includes data augmentation using a variational autoencoder (VAE), feature augmentation using a sparse autoencoder (SAE) and a convolutional neural network for classification. Pima Indians Diabetes Database, which takes into account information on the patients such as the number of pregnancies, glucose or insulin level, blood pressure or age, has been evaluated. Results: A 92.31% of accuracy was obtained when CNN classifier is trained jointly the SAE for featuring augmentation over a well-balanced dataset. This means an increment of 3.17% of accuracy with respect the state-of-the-art.

[6] Improving Accuracy for Diabetes Mellitus Prediction by Using Deepnet

Diabetes is a salient issue and a significant health care concern for many nations. The forecast for the prevalence of diabetes is on the rise. Hence, building a prediction machine learning model to assist in the identification of diabetic patients is of great interest. This study aims to create a machine learning model that is capable of predicting diabetes with high performance. The following study used the BigML platform to train four machine learning algorithms, namely, Deepnet, Models (decision tree), Ensemble and Logistic Regression, on data sets collected from the Ministry of National Guard Hospital Affairs (MNGHA) in Saudi Arabia between the years of 2013 and 2015. The

comparative evaluation criteria for the four algorithms examined included; Accuracy, Precision, Recall, F-measure and PhiCoefficient. Results show that the Deepnet algorithm achieved higher performance compared to other machine learning algorithms based on various evaluation matrices.

[7] Deep Learning Approach for Diabetes Prediction Using PIMA Indian Dataset

International Diabetes Federation (IDF) stated that 382 million people are living with diabetes worldwide. Over the last few years, the impact of diabetes has been increased drastically, which makes it a global threat. At present, Diabetes has steadily been listed in the top position as a major cause of death. The number of affected people will reach up to 629 million i.e. 48% increase by 2045. However, diabetes is largely preventable and can be avoided by making lifestyle changes. These changes can also lower the chances of developing heart disease and cancer. So, there is a dire need for a prognosis tool that can help the doctors with early detection of the disease and hence can recommend the lifestyle changes required to stop the progression of the deadly disease. Diabetes if untreated may turn into fatal and directly or indirectly invites lot of other diseases such as heart attack, heart failure, brain stroke and many more. Therefore, early detection of diabetes is very significant so that timely action can be taken and the progression of the disease may be prevented to avoid further complications. Healthcare organizations accumulate huge amount of data including Electronic health records, images, omics data, and text but gaining knowledge and insight into the data remains a key challenge. The latest advances in Machine learning technologies can be applied for obtaining hidden patterns, which may diagnose diabetes at an early phase. This research paper presents a methodology for diabetes prediction using a diverse machine learning algorithm using the PIMA dataset.

[8] Diabetes Prediction by Using Big Data Tool and Machine Learning Approaches

The use of big data in daily life is increasing from health care, social networks, banking systems, entry into the banking system, use of sensors and smart devices, leading to large amounts of data. That's why, it is necessary to develop a model and device that handles data in optimized form. In this paper, diabetes predicated from a data set with the help of various machine-learning algorithms such as Naive Bayes, KNN algorithm, Random forest, logistic regression. The main objective of the paper is

to observe diabetes disease with the help of big data tools and machine learning model. For doing this, the authors can select more accurate model with the help of some matrices. This paper predict diabetes disease using four machine learning models and then compare their performance among themselves. Machine learning provides more flexible and scalability than the older bio statistical method that helps it perform a variety of tasks such as risk detection, diagnosis, classification, and prediction.

[9] Internet of Things-Inspired Healthcare System for Urine-Based Diabetes Prediction

Healthcare industry is the leading domain that has been revolutionized by the incorporation of Internet of Things (IoT) technology resulting in smart medical applications. Conspicuously, this study presents an effective system of home-centric Urine-based Diabetes (UbD) monitoring system. Specifically, the proposed system comprises of 4-layers for predicting and monitoring diabetes-oriented urine infection. The system layers including Diabetic Data Acquisition (DDA) layer, Diabetic Data Classification(DDC) layer, Diabetic Mining and Extraction (DME) layer, and Diabetic Prediction and Decision Making (DPDM) layer allow an individual not exclusively to track his/her diabetes measure on regular basis but the prediction procedure is also accomplished so that prudent steps can be taken at early stages. Additionally, probabilistic measurement of UbD monitoring in terms of Level of Diabetic Infection (LoDI), which is cumulatively quantified as Diabetes Infection Measure (DIM) has been performed for predictive purposes using Recurrent Neural Network (RNN).

[10] Diabetes Disease Prediction Using Artificial Intelligence

The major problem area for researchers is disease diagnosis and the main interest of the medicine is an accurate diagnosis. Many engineering techniques have been developed in the past to help the medical staff with a diagnosis tool. There are many traditional methods of disease diagnosis, but the application of machine learning techniques has given a new dimension to this area. In this work, two different approaches have been used for the purpose of classification between diabetic and non-diabetic, using Pima Indian Diabetes Dataset. Principal Component Analysis has been used in the purpose of feature dimension reduction before applying any proposed classifier. Support Vector Machine (SVM) and Naïve Bayes (NB) are the two classifiers used in our study. 94.14 % and 93.88% are the accuracies obtained for the SVM and

NB approaches. The results obtained are very interesting and show improvement from the previous works. With this accurate learning technique, there is enough scope for improvement considerably in this field.

[11] Diabetes Prediction Using Ensembling of Different Machine Learning Classifiers

Diabetes, also known as chronic illness, is a group of metabolic diseases due to a high level of sugar in the blood over a long period. The risk factor and severity of diabetes can be reduced significantly if the precise early prediction is possible. The robust and accurate prediction of diabetes is highly challenging due to the limited number of labeled data and also the presence of outliers (or missing values) in the diabetes datasets. In this literature, we are proposing a robust framework for diabetes prediction where the outlier rejection, filling the missing values, data standardization, feature selection, K-fold cross-validation, and different Machine Learning (ML) classifiers (k-nearest Neighbor, Decision Trees, Random Forest, AdaBoost, Naive Bayes, and XGBoost) and Multilayer Perceptron (MLP) were employed. The weighted ensembling of different ML models is also proposed, in this literature, to improve the prediction of diabetes where the weights are estimated from the corresponding Area under ROC Curve (AUC) of the ML model. AUC is chosen as the performance metric, which is then maximized during hyper parameter tuning using the grid search technique. All the experiments, in this literature, were conducted under the same experimental conditions using the Pima Indian Diabetes Dataset.

[12] Prediction of Diabetes Based on Personal Lifestyle Indicators

Diabetes Mellitus or Diabetes has been portrayed as worse than Cancer and HIV (Human Immunodeficiency Virus). It develops when there are high blood sugar levels over a prolonged period. Recently, it has been quoted as a risk factor for developing Alzheimer, and a leading cause for blindness & kidney failure. Prevention of the disease is a hot topic for research in the healthcare community. Many techniques have been discovered to find the causes of diabetes and cure it. This research paper is a discussion on establishing a relationship between diabetes risk likely to be developed from a person's daily lifestyle activities such as his/her eating habits, sleeping habits, physical activity along with other indicators like BMI (Body Mass Index), waist circumference etc. Initially, a Chi-Squared Test of Independence was performed followed by

application of the CART (Classification and Regression Trees) machine learning algorithm on the data and finally using Cross-Validation, the bias in the results was removed.

[13] Random Forest Algorithm for the Prediction of Diabetes

Diabetes is a dreadful disease identified by escalated levels of glucose in the blood. Machine learning algorithms help in identification and prediction of diabetes at an early stage. The main objective of this study is to predict diabetes mellitus with better accuracy using an ensemble of machine learning algorithms. The Pima Indians Diabetes dataset has been considered for experimentation, which gathers details of patients with and without having diabetes. The proposed ensemble soft voting classifier gives binary classification and uses the ensemble of three machine learning algorithms viz. random forest, logistic regression, and Naive Bayes for the classification. Empirical evaluation of the proposed methodology has been conducted with state-of-the-art methodologies and base classifiers.

[14] Blood Glucose Level Prediction for Diabetics Based on Nutrition and Insulin Administration Logs Using Personalized Mathematical Models

According to recent surveys, the current ways of diabetics trying to estimate their insulin need based on experience and conjecture are sometimes inefficient in practice. This paper proposes a prediction algorithm and presents the validation of the model in outpatient care. This algorithm consists of two state-of-the-art models that calculate nutrition absorption and glycaemia including insulin evolution. ,e combined model is extended with personalized parameter training including genetic algorithm and Nelder-Mead method, and a more realistic, diurnal parameter profile as a representation of the natural biorhythm. , is method implemented in a user-friendly application can help diabetics calculate their insulin need. E-tests were performed on a data set including a clinical trial involving more than 20 diabetic patients. We experienced 55% improvement in the results due to model training compared to the tests based on literature parameters. In the best case, 92.5% of the predicted blood glucose level values were in the range of clinically acceptable errors, which means around 2.8 mmol/l root mean square error. Results of the validation based on outpatient data are promising compared to others found in the literature. Handling other important factors such as physical activity and stress remains a challenge for future research.

[15] A Model for Early Prediction of Diabetes

Diabetes is a common, chronic disease. Prediction of diabetes at an early stage can lead to improved treatment. Data mining techniques are widely used for prediction of disease at an early stage. In this research paper, diabetes is predicted using significant attributes, and the relationship of the differing attributes is also characterized. Various tools are used to determine significant attribute selection, and for clustering, prediction, and association rule mining for diabetes. Significant attribute selection was done via the principal component analysis method. Our findings indicate a strong association of diabetes with body mass index (BMI) and with glucose level, which was extracted via the Apriori method. Artificial neural network (ANN), random forest (RF) and K-means clustering techniques were implemented for the prediction of diabetes. The ANN technique provided a best accuracy of 75.7%, and may be useful to assist medical professionals with treatment decisions.

[16] Big Data and Diabetes: The Applications of Big Data for Diabetes Care Now And In The Future.

We review current applications of Big Data in diabetes care and consider the future potential by carrying out a scoping study of the academic literature on Big Data and diabetes care. Healthcare data are being produced at ever -increasing rates, and this information has the potential to transform the provision of diabetes care. Big Data is beginning to have an impact on diabetes care through data research. The use of Big Data for routine clinical care is still a future application. Vast amounts of healthcare data are already being produced, and the key is harnessing these to produce actionable insights. Considerable development work is required to achieve these goals.

[17] Diabetes Prediction Using Data Mining Techniques

Data science methods have the potential to benefit other scientific fields by shedding new light on common questions. One such task is help to make predictions on medical data. Diabetes mellitus or simply diabetes is a disease caused due to the increase level of blood glucose. Various traditional methods, based on physical and chemical tests, are available for diagnosing diabetes. The methods strongly based on the data mining techniques can be effectively applied for high blood pressure risk prediction. In this paper, we explore the early prediction of diabetes via five different data mining methods including: GMM, SVM, Logistic regression, ELM, ANN. The

experiment result proves that ANN (Artificial Neural Network) provides the highest accuracy than other techniques.

[18] Prediction of Diabetes Using Machine Learning Algorithms in Healthcare

There are several machine learning techniques that are used to perform predictive analytics over big data in various fields. Predictive analytics in healthcare is a challenging task but ultimately can help practitioners make big data-informed timely decisions about patient's health and treatment. This paper discusses the predictive analytics in healthcare, six different machine learning algorithms are used in this research work. For experiment purpose, a dataset of patient's medical record is obtained and six different machine learning algorithms are applied on the dataset. Performance and accuracy of the applied algorithms is discussed and compared. Comparison of the different machine learning techniques used in this study reveals which algorithm is best suited for prediction of diabetes. This paper aims to help doctors and practitioners in early prediction of diabetes using machine learning.

[19] Early Detection of Prediabetes and T2DM Using Wearable Sensors and Internet-of-Things-Based Monitoring Applications

Prediabetes and type 2 diabetes mellitus (T2DM) are one of the major long-term health conditions affecting global healthcare delivery. One of the few effective approaches is to actively manage diabetes via a healthy and active lifestyle. Objectives this research is focused on early detection of prediabetes and T2DM using wearable technology and Internet-of-Things-based monitoring applications. We developed an artificial intelligence model based on adaptive neuro fuzzy inference to detect prediabetes and T2DM via individualized monitoring. The key contributing factors to the proposed model include heart rate, heart rate variability, breathing rate, breathing volume, and activity data (steps, cadence, and calories). The data was collected using an advanced wearable body vest and combined with manual recordings of blood glucose, height, weight, age, and sex. The model analyzed the data alongside a clinical knowledgebase. Fuzzy rules were used to establish baseline values via existing interventions, clinical guidelines, and protocols. The proposed model was tested and validated using Kappa analysis and achieved an overall agreement of 91%. We also present a 2-year follow-up observation from the prediction results of the original model.

Moreover, the diabetic profile of a participant using M-health applications and a wearable vest (smart shirt) improved when compared to the traditional/routine practice.

[20] Diabetes Prediction Model Using Cloud Analytics

Diabetes is now a global disease, which can affect the normal living lifestyle and workflow of a person. Due to the lack of the insulin in the body, a man can get a diagnosis of Diabetes. There are Type-1 and Type-2, two kinds of Diabetes. There are few factors in a person's daily life like Hypertension, Heredity, Daily Standard Activity, Smoking Habits, Body Mass index (BMI), for women sometimes it is the number of pregnancies, any kind of heart problem etc. that may cause Diabetes. There are also some medical factors as well (like the level of glucose). Our aim is to reduce the rate of Diabetes by predicting the most important factor, which can cause a person Diabetes in the future. Therefore, by the help of this prediction a person can easily get the information about how he or she can control their Diabetes by controlling the main causing factors and get the advantage of knowing if he or she has any chance of getting a diagnosis with Diabetes in future. In this paper, we use the classification and predictive analysis algorithm to predict the important factors for the cause of diabetes and discussed how the predictive model can be implemented in the cloud environment to make the model non-temporal, which will be helpful to find the probability of a person getting a diagnosis of Diabetes in future.

CHAPTER 3
SYSTEM ANALYSIS

3.1 PROBLEM STATEMENT

Diabetes is one of the leading causes of death worldwide and remains an important health burden. Every year millions of people die due to Diabetes, but up to 80% of Diabetes can be prevented if we can identify or predict the occurrence of Diabetes in its early stage.

3.2 PROBLEM DESCRIPTION

Poor lifestyle, diet, and work pressure lead the diabetes disease which may cause several fatal health issues like heart attack, strokes, kidney failure, nerve damage, etc. In the medical field, Diabetes is detected by using so many techniques such as Cloud-based, Blockchain, Internet of things, Deep learning, Data mining, and Big data which are very time-consuming and do not produce accurate results. Therefore, to overcome this problem, an alternative way is to design a system that will identify Diabetes by using the health condition of a person using algorithms in machine learning, which also provides faster and more accurate solutions which are very useful to save life's.

3.3 EXISTING SYSTEM

Diabetes can be effectively managed when caught early with high accuracy. In the existing system, Cloud-based, Blockchain, Internet of things, Deep learning, Data mining, Big data, and Machine learning algorithms such as Random Forest, Logistic Regression, k-Nearest Neighbor, and Naive Bayes have predicted diabetes with low accuracy in the early stage. Implementation of Some machine learning algorithms is a bit complex to build due to the lack of information about the data visualization. Mathematical calculations are used in the existing systems for model building this may takes a lot of time and complexity. The prediction results are not accurate in existing systems.

3.4 DISADVANTAGES OF EXISTING SYSTEM

- Less accurate compared to the proposed system.
- Difficult to make or deal with large sizes of data.
- Time-consuming to predict value.

3.5 PROPOSED SYSTEM

To improve the accuracy in the detection of diabetics in the early stage, We use Machine Learning Techniques like Support Vector Machine, Decision Tree, AdaBoost, Linear Discriminant Analysis, and K-means algorithms have given the best accuracy to detect whether the patient has Diabetes or not. This system uses machine learning packages available in the scikit-learn library to improve the accuracy of the prediction.

3.6 ADVANTAGES OF PROPOSED SYSTEM

- It takes less time to compute results.
- More accurate results.
- It will deal with large size of data where the existing system can’t.
- More flexible compared to the existing system.

CHAPTER 4
SYSTEM REQUIREMENTS

4.1 HARDWARE REQUIREMENTS

- Operating system : Windows 7 or above
- RAM : 8 GB
- Hard disc or SSD : More than 500 GB
- Processor : Intel 3rd generation or high or Ryzen with 8 GB Ram

4.2 SOFTWARE REQUIREMENTS

- Software's : Python 3.6 or high version
- IDE : PyCharm.
- Framework : Flask
- Server Deployment : XAMPP Server
- Database Management Tool : SQLyog

4.3 FEASIBILITY STUDY

The feasibility of the project is analyzed in this phase and the 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.

Three key considerations involved in the feasibility analysis are:

- Economical Feasibility
- Technical Feasibility
- Social Feasibility

4.3.1 Economical Feasibility

This study is carried out to check the economic impact that the system will have on the organization. The amount of funds 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.

4.3.2 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.

4.3.3 Social Feasibility

The aspect of the 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/ Her level of confidence must be raised so that he/she is also able to make some constructive criticism, which is welcomed, as he/she is the final user of the system.

CHAPTER 5

SYSTEM DEVELOPMENT

5.1 SYSTEM DESIGN

5.1.1 System Architecture

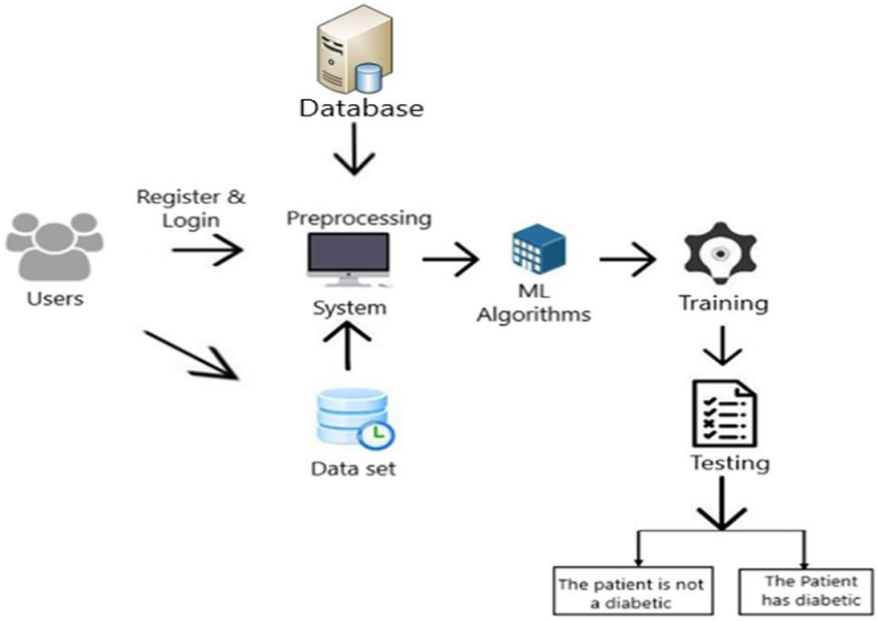


Figure 5.1: *System Architecture*

Description:

Initially user register into the diabetes care portal by giving details like username, email and password and the details stored in database. After registration user login into the home page of the diabetes care. Next user upload the diabetes data set (pima diabetes data set), After uploading the data set user view the uploaded data set. The user train the system using machine learning algorithms available in the training portal of the diabetes care. Finally use get the results by entering the user details whether the person has diabetes or not.

5.1.2 Modules

Our system has two main modules such as,

1. User
2. Diabetes Care

5.1.2.1 User

Users can register for the Diabetes Care application. After registering, the user can access Diabetes care portal, Here is where the user uploads his dataset, The user can view the dataset which he was uploaded, User select the model that predicts disease with better accuracy, User view the accuracy of selected model, The user must provide input values for the certain fields in order to get results, User view's the generated results from the model, After completing all processes, the user can logout from diabetes login home page.

5.1.2.2 Diabetes Care

The Diabetes care train the machine learning model to improve model accuracy by using user uploaded dataset. After training, the system generate the accuracy of selected model and generate the results wheatear the person has diabetes or not based on user input values.

5.1.3 Algorithm

- **Support Vector Machine**

Support Vector Machine or SVM algorithm is a simple yet powerful Supervised Machine Learning algorithm that can be used for building both regression and classification models. SVM algorithm can perform really well with both linearly separable and non-linearly separable datasets. Even with a limited amount of data, the support vector machine algorithm does not fail to show its magic.

Step 1: Load Pandas library and the dataset using pandas.

Step 2: Define the features and the target.

Step 3: Split the dataset into train and test using Sklearn before building the SVM algorithm model.

Step 4: Import the support vector classifier function or SVC function from Sklearn SVM module. Build the Support Vector Machine model with the help of the SVC function.

Step 5: Predict values using the SVM algorithm model.

Step 6: Evaluate the Support Vector Machine model.

- **Linear Discriminant Analysis**

Linear Discriminant analysis is one of the most popular dimensionality reduction techniques used for supervised classification problems in machine learning. It is also considered a pre-processing step for modeling differences in ML and applications of pattern classification. Whenever there is a requirement to separate two or more classes having multiple features efficiently, the Linear Discriminant Analysis model is considered the most common technique to solve such classification problems. For e.g., if we have two classes with multiple features and need to separate them efficiently. When we classify them using a single feature, then it may show overlapping.

Linear Discriminant analysis is used as a dimensionality reduction technique in machine learning, using which we can easily transform a 2-D and 3-D graph into a 1-dimensional plane.

- **Decision Tree**

A tree has many analogies in real life, and turns out that it has influenced a wide area of machine learning, covering both classification and regression. In decision analysis, a decision tree can be used to visually and explicitly represent decisions and decision making. As the name goes, it uses a tree-like model of decisions. Though a commonly used tool in data mining for deriving a strategy to reach a particular goal.

A decision tree is drawn upside down with its root at the top. In the image on the left, the bold text in black represents a condition/internal node, based on which the tree splits into branches/ edges. The end of the branch that doesn't split anymore is the decision/leaf, in this case, whether the passenger died or survived, represented as red and green text respectively.

- **AdaBoost**

AdaBoost algorithm, short for Adaptive Boosting, is a Boosting technique used as an Ensemble Method in Machine Learning. It is called Adaptive Boosting as the weights are re-assigned to each instance, with higher weights assigned to incorrectly classified instances. Boosting is used to reduce bias as well as variance for supervised learning. It works on the principle of learners growing sequentially. Except for the first, each subsequent learner is grown from previously grown learners. In simple words, weak learners are converted into strong ones. The AdaBoost algorithm works on the

same principle as boosting with a slight difference. Let's discuss this difference in detail.

First, let us discuss how boosting works. It makes 'n' number of decision trees during the data training period. As the first decision tree/model is made, the incorrectly classified record in the first model is given priority. Only these records are sent as input for the second model. The process goes on until we specify a number of base learners we want to create. Remember, repetition of records is allowed with all boosting techniques.

- **K-Means**

K-Means Clustering is an Unsupervised Learning algorithm, which groups the unlabeled dataset into different clusters. Here K defines the number of pre-defined clusters that need to be created in the process, as if K=2, there will be two clusters, and for K=3, there will be three clusters, and so on.

It allows us to cluster the data into different groups and a convenient way to discover the categories of groups in the unlabeled dataset on its own without the need for any training. It is a centroid-based algorithm, where each cluster is associated with a centroid. The main aim of this algorithm is to minimize the sum of distances between the data point and their corresponding clusters.

5.1.4 UML Diagrams

5.1.4.1 Introduction to UML

UML is a language for visualizing, specifying, constructing and documenting the artifacts of software intensive system. UML is simply another graphical representation of a common semantic model. UML diagrams are the ultimate output of the entire discussion. All the elements, relationships are used to make a complete UML diagram and the diagram represents a system.

The visual effect of the UML diagram is the most important part of the entire process. All the other elements are used to make it a complete one.

UML includes the following nine diagrams and the details are described in the following chapters.

- Class diagram
- Object diagram

- Use case diagram
- Sequence diagram
- Collaboration diagram
- Activity Diagram
- State chart diagram
- Deployment diagram
- Component diagram

UML defines several models for representing systems

1. Use case diagram: Represents the functions of a system from the user's point of view.

2. Class diagram: Represents the static structure in terms of classes and relationships. A class diagram shows a set of classes, interfaces and collaborations and their relationships.

3. Object diagram: represents objects and their relationships and correspond to simplified collaboration diagrams that do not represent message broadcasts.

4. Sequence diagram: Temporal representation of objects, links and interactions.

5. Collaboration diagram: Spatial representation of objects, links, and interactions.

6. State chart diagram: Represents the behavior of a class in terms of states at run time.

7. Activity diagram: Represents the behavior of an operation as a set of actions

8. Component diagram: Represents the physical components of an application.

9. Deployment diagram: Represents the deployment of components on particular pieces of hardware.

Advantages:

- To represent complete systems (instead of only the software portion) using object-oriented concepts.
- To establish an explicit coupling between concepts and executable code.
- To take into account the scaling factors that are inherent to complex and critical systems.
- To create a modeling language usable by both humans and machines.

Conceptual model of UML can be mastered by learning the following three major

elements:

- UML building blocks.
- Rules to connect the building blocks.
- Common mechanisms of UML.

5.1.4.2 Class Diagram

Aim*:* To implement class diagram for Machine Learning for Diabetes prediction.

Description: A class diagram shows a set of classes, interfaces and collaborations and their relationships. In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, attributes and operations.

Objective: The main objective of the class diagram to illustrate the static design of a view system.

Things: class, interfaces, collaboration, active class.

Relationships: Dependency, generalization and association.

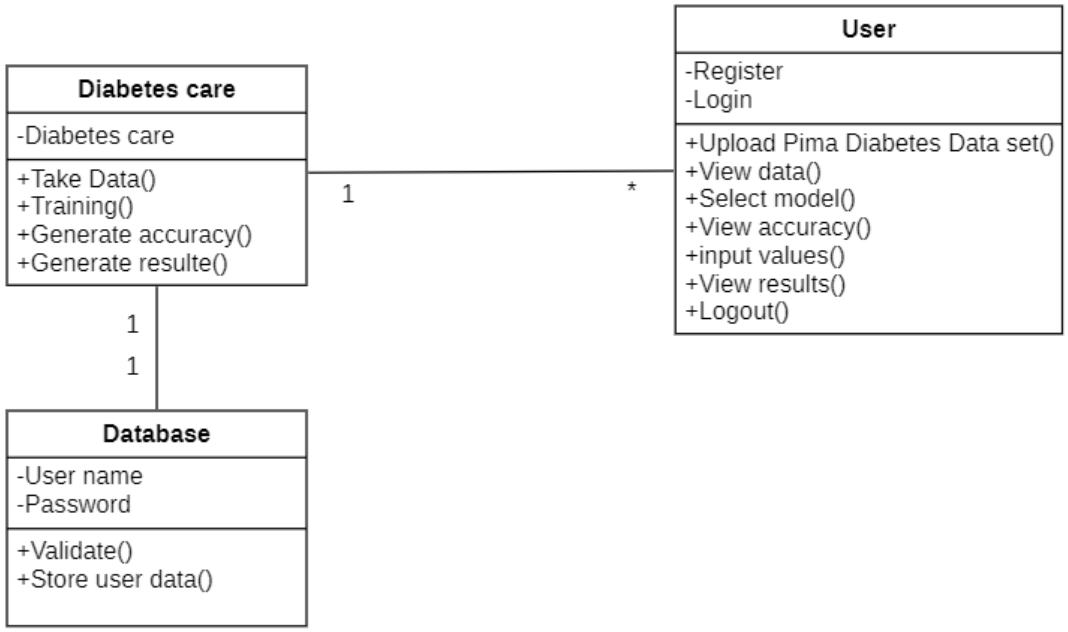


Figure 5.2: *Class Diagram*

5.1.4.3 Use Case Diagram

Aim: To implement use case diagram for Machine Learning for Diabetes prediction.

Description: A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of

actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.

Objective: Use case diagram organizes the behavior of the system.

Things: Use cases, Actors.

Relationships: Dependency, generalization and Association.

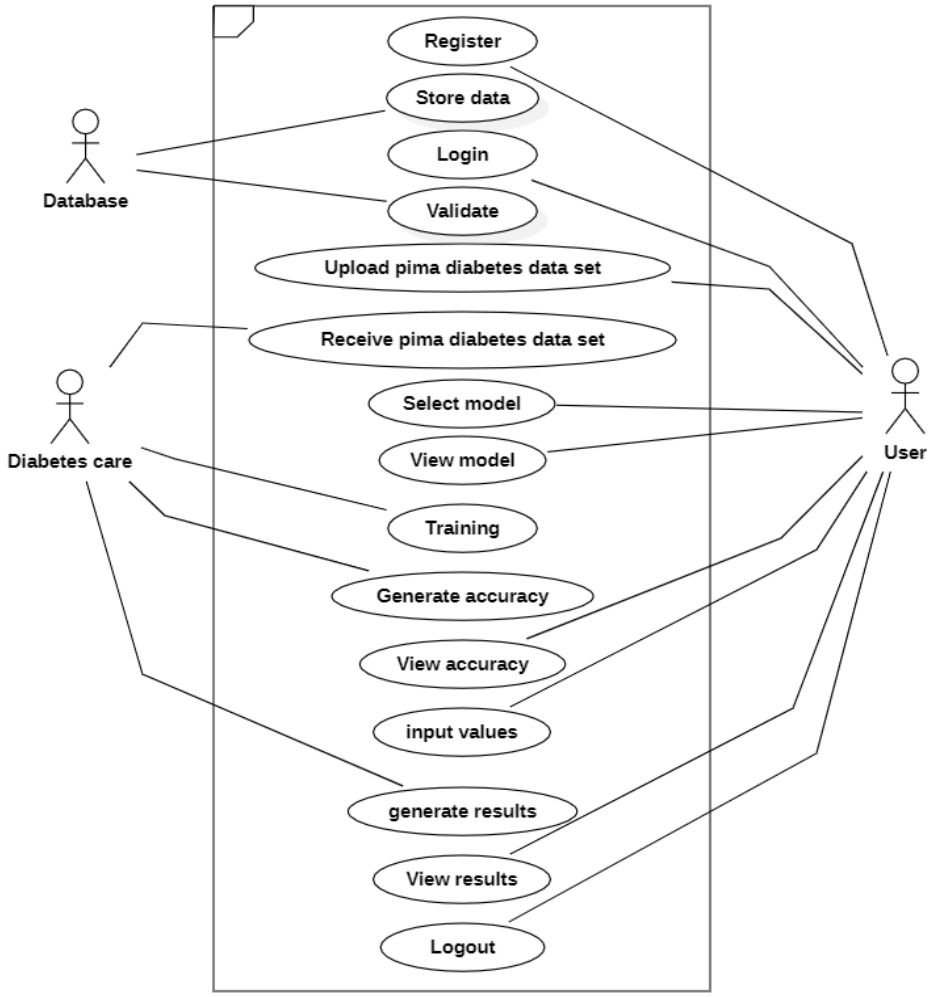


Figure 5.3: *Use Case Diagram*

5.1.4.4 Sequence Diagram

Aim: To implement sequence diagram for Machine Learning for Diabetes prediction.

Description: A Sequence diagram is an interaction diagram that emphasizes the time ordering messages. It shows a set of objects and messages sent and received by the by those objects.

Objective: To illustrate the dynamic view of system

Things: Objects and Messages

Relationships: Time and life line, Links.

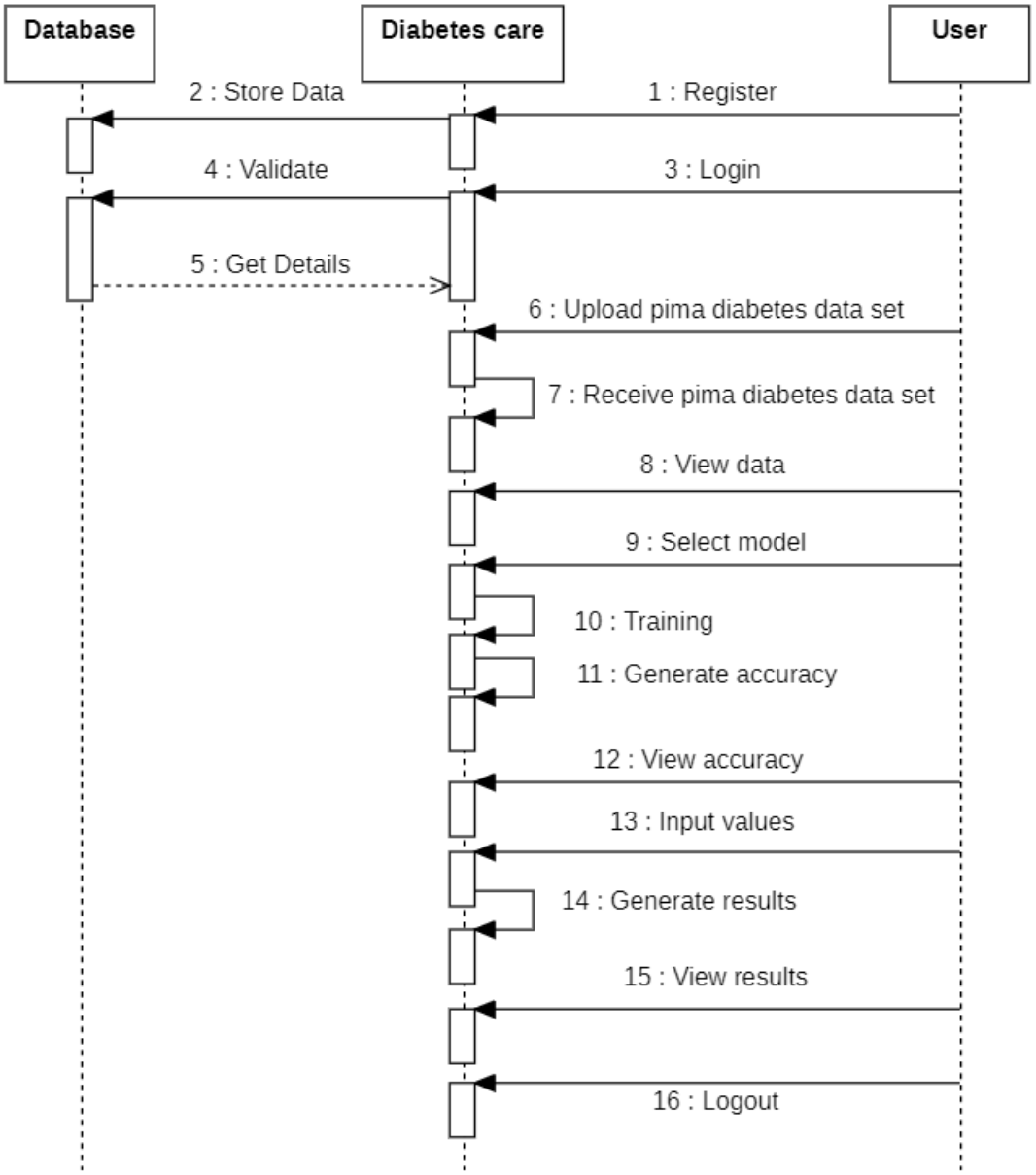


Figure 5.4*: *Sequence Diagram

5.1.4.5 Activity Diagram

Aim: To implement activity diagram for Machine Learning for Diabetes prediction.

Description: An activity diagram is essentially a flow chart, showing flow of control from activity to activity. It involves modeling the sequential steps in computational process. Activity diagram not only important for modeling dynamic aspects of a system, but also for constructing executable system through forward and reverse engineering.

Objective: Focused on flow of control from activity to activity.

Things: State and object.

Relationships: Transitions.

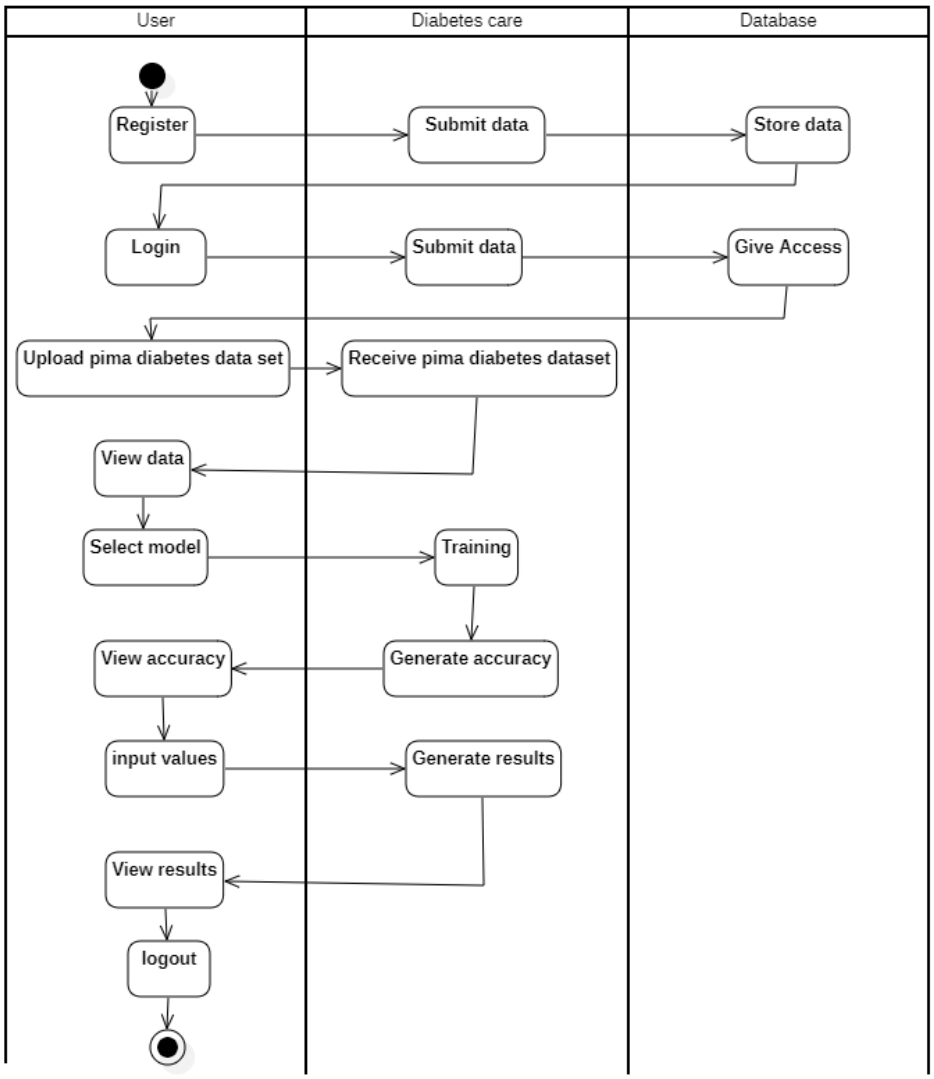


Figure 5.5: *Activity Diagram*

5.1.4.6 Deployment Diagram

Aim: To implement a deployment diagram for Machine Learning for Diabetes prediction.

Description: Deployment diagram is a type of diagram that specifies the physical hardware on which the software system will execute. The software system is manifested using various artifacts.

Objective: Used with the sole purpose of describing how software is deployed into the hardware system.

Things: Node, Component, Artifact and Interface. `

Relationships: Nodes.

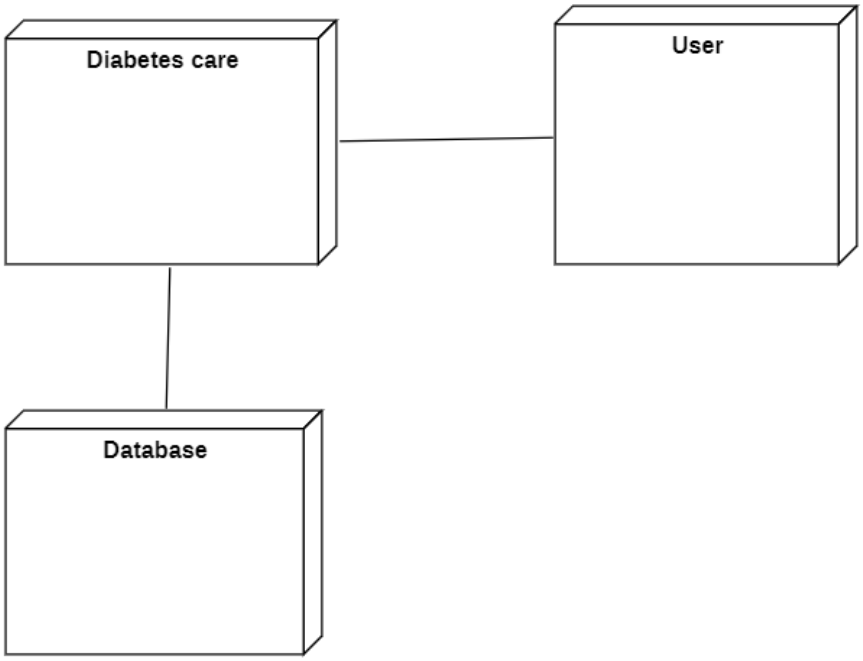


Figure 5.6: *Deployment Diagram*

5.2 SYSTEM IMPLEMENTATION

5.2.1 Software Description

Python

Script

This is a very useful capability that allows you to type in a program and to have it executed immediately in an interactive mode.

Scripts are reusable

Basically, a script is a text file containing the statements that comprise a Python program. Once you have created the script, you can execute it over and over without having to retype it each time.

Scripts are editable

Perhaps, more importantly, you can make different versions of the script by modifying the statements from one file to the next using a text editor. Then you can execute each of the individual versions. In this way, it is easy to create different programs with a minimum amount of typing.

You will need a text editor Just about any text editor will suffice for creating Python script files. You can use Microsoft Notepad, Microsoft WordPad, Microsoft Word, or just about any word processor if you want to.

Difference between a Script and a Program

Scripts are distinct from the core code of the application, which is usually written in a different language, and are often created or at least modified by the end-user. Scripts are often interpreted from source code or byte code, whereas the applications they control are traditionally compiled to native machine code.

Program

The program has an executable form that the computer can use directly to execute the instructions. The same program in its human-readable source code form, from which executable programs are derived (e.g., compiled).

Python

What is Python? Chances you are asking yourself this. You may have found this book because you want to learn to program but don't know anything about programming languages. Or you may have heard of programming languages like C, C++, C#, or Java and want to know what Python is and how it compares to "big name" languages. Hopefully I can explain it for you.

Python Concepts

Python is one of the best languages available and why it's a great one to start programming with.

- Open source general-purpose language.
- Object Oriented, Procedural, Functional
- Easy to interface with C/Java/Fortran
- Easy-is to interface with C++
- Great interactive environment

Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently where as other languages use punctuation, and it has fewer syntactical constructions than other languages.

- **Python is Interpreted** − Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL.
- **Python is Interactive** − you can actually sit at a Python prompt and interact with the interpreter directly to write your programs.

- **Python is Object-Oriented** − Python supports Object-Oriented style or technique of programming that encapsulates code within objects.
- **Python is a Beginner's Language** − Python is a great language for the beginner-level programmers and supports the development of a wide range of applications from simple text processing to WWW browsers to games.

History of Python: Python was developed by Guido van Possum in the late eighties at the National Research Institute for Mathematics and Computer Science in the Netherlands. Python is derived from many other languages, including ABC, Modula-3, C, C++, Algol-68, Smalltalk, and UNIX shell and other scripting languages.
Python is copyrighted. Like Perl, Python source code is now available under the GNU General Public License (GPL). Python is now maintained by a core development team at the institute, although Guido van Possum still holds a vital role in directing its progress.

Python Features
Python's features include,

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

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

Dynamic vs Static

Types Python is a dynamic-typed language. Many other languages are static typed, such as C/C++ and Java. A static typed language requires the programmer to explicitly tell the computer what type of "thing" each data value is.

For example, in C if you had a variable that was to contain the price of something, you would have to declare the variable as a "float" type.

This tells the compiler that the only data that can be used for that variable must be a floating point number, i.e., a number with a decimal point.

If any other data value was assigned to that variable, the compiler would give an error when trying to compile the program.

Python, however, doesn't require this. You simply give your variables names and assign values to them. The interpreter takes care of keeping track of what kinds of objects your program is using. This also means that you can change the size of the values as you develop the program. Say you have another decimal number (a.k.a. a floating point number) you need in your program.

Variables

Variables are nothing but reserved memory locations to store values. This means that when you create a variable you reserve some space in memory.

Based on the data type of a variable, the interpreter allocates memory and decides what can be stored in the reserved memory. Therefore, by assigning different data types to variables, you can store integers, decimals or characters in these variables.

Standard Data Types

The data stored in memory can be of many types. For example, a person's age is stored as a numeric value and his or her address is stored as alphanumeric characters. Python has various standard data types that are used to define the operations possible on them and the storage method for each of them.

Python has five standard data types −

- Numbers

- String
- List
- Tuple
- Dictionary

Python Numbers

Number data types store numeric values. Number objects are created when you assign a value to them

Python Strings

Strings in Python are identified as a contiguous set of characters represented in the quotation marks. Python allows for either pairs of single or double quotes. Subsets of strings can be taken using the slice operator ([ ] and [:]) with indexes starting at 0 in the beginning of the string and working their way from -1 at the end.

Python Lists

Lists are the most versatile of Python's compound data types. A list contains items separated by commas and enclosed within square brackets ([]). To some extent, lists are similar to arrays in C. One difference between them is that all the items belonging to a list can be of different data type.

The values stored in a list can be accessed using the slice operator ([ ] and [:]) with indexes starting at 0 in the beginning of the list and working their way to end -1. The plus (+) sign is the list concatenation operator, and the asterisk (*) is the repetition operator.

Python Tuples

A tuple is another sequence data type that is similar to the list. A tuple consists of a number of values separated by commas. Unlike lists, however, tuples are enclosed within parentheses.

The main differences between lists and tuples are: Lists are enclosed in brackets ([ ]) and their elements and size can be changed, while tuples are enclosed in parentheses (( )) and cannot be updated. Tuples can be thought of as read-only lists.

Python Dictionary

Python's dictionaries are kind of hash table type. They work like associative arrays or hashes found in Perl and consist of key-value pairs. A dictionary key can be almost any Python type, but are usually numbers or strings. Values, on the other hand, can be any arbitrary Python object.

Dictionaries are enclosed by curly braces ({ }) and values can be assigned and accessed using square braces ([]).

Different Modes in Python

Python has two basic modes: normal and interactive. The normal mode is the mode where the scripted and finished .pie files are run in the Python interpreter. Interactive mode is a command line shell which gives immediate feedback for each statement, while running previously fed statements in active memory. As new lines are fed into the interpreter, the fed program is evaluated both in part and in whole 10

Python Libraries:

1. Requests: The most famous http library written by Kenneth remits. It's a must have for every python developer.
2. Scrappy: If you are involved in web scraping then this is a must have library for you. After using this library you won't use any other.
3. Python: A guy toolkit for python. I have primarily used it in place of tinder. You will really love it.
4. Pillow: A friendly fork of PIL (Python Imaging Library). It is more user friendly than PIL and is a must have for anyone who works with images.
5. SQL Alchemy: A database library. Many love it and many hate it. The choice is yours.
6. Beautiful Soup: I know it's slow but this xml and html parsing library is very useful for beginners.
7. Twisted: The most important tool for any network application developer. It has a very beautiful ape and is used by a lot of famous python developers.
8. NumPy: How can we leave this very important library? It provides some advance math functionalities to python.
9. Skippy: we have to talk about skippy. It is a library of algorithms and mathematical tools for python and has caused many scientists to switch from ruby to python.

10. Matplotlib: A numerical plotting library. It is very useful for any data scientist or any data analyser.

Python Class and Objects

These are the building blocks of OOP. Class creates a new object. This object can be anything, whether an abstract data concept or a model of a physical object, e.g. a chair. Each class has individual characteristics unique to that class, including variables and methods. Classes are currently "the big thing" in most programming languages. Hence, there are several chapters dedicated to OOP later in the book.

The class is the most basic component of object-oriented programming. Previously, you learned how to use functions to make your program do something.

Now will move into the big, scary world of Object-Oriented Programming (OOP). To be honest, it took me several months to get a handle on objects. When I first learned C and C++, I did great; functions just made sense for me.

Once you learn OOP, you'll realize that it's actually a pretty powerful tool. Plus many Python libraries and APIs use classes, so you should at least be able to understand what the code is doing.

Here's a brief list of Python OOP ideas:

- The class statement creates a class object and gives it a name. This creates a new namespace.
- Assignments within the class create class attributes. These attributes are accessed by qualifying the name using dot syntax.
- Class attributes export the state of an object and its associated behavior. These attributes are shared by all instances of a class.
- Calling a class (just like a function) creates a new instance of the class. This is where the multiple copies part comes in.
- Each instance gets ("inherits") the default class attributes and gets its own namespace. This prevents instance objects from overlapping and confusing the program.

Inheritance

First off, classes allow you to modify a program without really making changes to it. To elaborate, by sub classing a class, you can change the behavior of the program

by simply adding new components to it rather than rewriting the existing components. As we've seen, an instance of a class inherits the attributes of that class.

However, classes can also inherit attributes from other classes. Hence, a subclass inherits from a superclass allowing you to make a generic superclass that is specialized via subclasses. The subclasses can override the logic in a superclass, allowing you to change the behavior of your classes without changing the superclass at all.

Operator Overloading

Operator overloading simply means that objects that you create from classes can respond to actions (operations) that are already defined within Python, such as addition, slicing, printing, etc.

Even though these actions can be implemented via class methods, using overloading ties the behavior closer to Python's object model and the object interfaces are more consistent to Python's built-in objects, hence overloading is easier to learn and use. User-made classes can override nearly all of Python's built-in operation methods.

Exceptions

I've talked about exceptions before but now I will talk about them in depth. Essentially, exceptions are events that modify program's flow, either intentionally or due to errors.

They are special events that can occur due to an error, e.g. trying to open a file that doesn't exist, or when the program reaches a marker, such as the completion of a loop.

Exceptions, by definition, don't occur very often; hence, they are the "exception to the rule" and a special class has been created for them. Exceptions are everywhere in Python.

Virtually every module in the standard Python library uses them, and Python itself will raise them in a lot of different circumstances.

Here are just a few examples:

- Accessing a non−existent dictionary key will raise a Key Error exception.
- Searching a list for a non−existent value will raise a Value Error exception.

- Calling a non−existent method will raise an Attribute Error exception.
- Referencing a non−existent variable will raise a Name Error exception.
- Mixing data types without coercion will raise a Type Error exception.

User-Defined Exceptions

Python does allow for a programmer to create his own exceptions. However, before making your own exceptions, make sure there isn't one of the built-in exceptions that will work for you.

They have been "tested by fire" over the years and not only work effectively, they have been optimized for performance and are bug-free. Making your own exceptions involves object-oriented programming. To make a custom exception, the programmer determines which base exception to use as the class to inherit from, e.g. making an exception for negative numbers or one for imaginary numbers would probably fall under the Arithmetic Error exception class. To make a custom exception, simply inherit the base exception and define what it will do.

Python Modules

Python allows us to store our code in files (also called modules). This is very useful for more serious programming, where we do not want to retype a long function definition from the very beginning just to change one mistake. In doing this, we are essentially defining our own modules, just like the modules defined already in the Python library. To support this, Python has a way to put definitions in a file and use them in a script or in an interactive instance of the interpreter. Such a file is called a module*;* definitions from a module can be imported into other modules or into the main module.

Testing Code

- As indicated above, code is usually developed in a file using an editor.
- To test the code, import it into a Python session and try to run it.
- Usually there is an error, so you go back to the file, make a correction, and test again. This process is repeated until you are satisfied that the code works.
- His entire process is known as the development cycle.

- There are two types of errors that you will encounter. Syntax errors occur when the form of some command is invalid.
- This happens when you make typing errors such as misspellings, or call something by the wrong name, and for many other reasons. Python will always give an error message for a syntax error.

Functions in Python

It is possible, and very useful, to define our own functions in Python. Generally speaking, if you need to do a calculation only once, then use the interpreter. But when you or others have need to perform a certain type of calculation many times, then define a function.

You use functions in programming to bundle a set of instructions that you want to use repeatedly or that, because of their complexity, are better self-contained in a sub-program and called when needed. That means that a function is a piece of code written to carry out a specified task.

To carry out that specific task, the function might or might not need multiple inputs. When the task is carved out, the function can or cannot return one or more values.

There are three types of functions in python:

Help (), min (), print ().

Python Namespace

Generally speaking, a namespace (sometimes also called a context) is a naming system for making names unique to avoid ambiguity. Everybody knows a name spacing system from daily life, i.e. the naming of people in first name and family name (surname).

An example is a network: each network device (workstation, server, printer,) needs a unique name and address. Yet another example is the directory structure of file systems. This way, the same identifier can be independently defined in multiple namespaces. (Like the same file names in different directories) Programming languages, which support namespaces, may have different rules that determine to which namespace an identifier belongs. Namespaces in Python are implemented as Python dictionaries, this means it is a mapping from names (keys) to objects (values). The user

doesn't have to know this to write a Python program and when using namespaces. Some namespaces in Python:

- **Global names** of a module
- **Local names** in a function or method invocation
- **Built-in names:** this namespace contains built-in functions (e.g. abs(), camp(), ...) and built-in exception names

Garbage Collection

Garbage Collector exposes the underlying memory management mechanism of Python, the automatic garbage collector. The module includes functions for controlling how the collector operates and to examine the objects known to the system, either pending collection or stuck in reference cycles and unable to be freed.

Python XML Parser

XML is a portable, open source language that allows programmers to develop applications that can be read by other applications, regardless of operating system and/or developmental language.

The Extensible Markup Language XML is a markup language much like HTML or SGML. This is recommended by the World Wide Web Consortium and available as an open standard. XML is extremely useful for keeping track of small to medium amounts of data without requiring a SQL-based backbone. XML Parser Architectures and APIs the Python standard library provides a minimal but useful set of interfaces to work with XML.

This is useful when your documents are large or you have memory limitations, it parses the file as it reads it from disk and the entire file is never stored in memory.

Python Web Frameworks

A web framework is a code library that makes a developer's life easier when building reliable, scalable and maintainable web applications. Web frameworks encapsulate what developers have learned over the past twenty years while programming sites and applications for the web. Frameworks make it easier to reuse code for common HTTP operations and to structure projects so other developers with knowledge of the framework can quickly build and maintain the application.

Common Web Framework Functionality

Frameworks provide functionality in their code or through extensions to perform common operations required to run web applications. These common operations include:

1. URL routing
2. HTML, XML, JSON, and other output format tinplating
3. Database manipulation
4. Security against Cross-site request forgery (CSRF) and other attacks
5. Session storage and retrieval

5.2.2 Hardware Description

I3/Intel Processor

Developed and manufactured by Intel, and first introduced and released in 2010, the Core i3 is a dual-core computer processor, available for use in both desktop and laptop computers. It is one of three types of processors in the "i" series (also called the Intel Core family of processors).

The Core i3 processor is available in multiple speeds, ranging from 1.30 GHz up to 3.50 GHz, and features either 3 MB or 4 MB of cache. It utilizes either the LGA 1150 or LGA 1155 socket on a motherboard. Core i3 processors are most often found as dual-core, having two cores. However, a select few high-end Core i3 processors are quad-core, featuring four cores. The most common type of RAM used with a Core i3 processor is DDR3 1333 or DDR3 1600.

Power usage varies for the Core i3 processors:

- Slower speeds (1.30 GHz to 1.80 GHz) use 11.5 W, 15 W or 25 W of power
- Medium speeds (2.00 GHz to 2.50 GHz) use 28 W, 35 W or 37 W of power
- Faster speeds (2.90 GHz to 3.50 GHz) use 35 W, 37 W or 54 W of power.

Hard Disk: 160GB

Hard disk, also called hard disk drive or hard drive, magnetic storage medium for a computer. Hard disks are flat circular plates made of aluminium or glass and coated with a magnetic material. Hard disks for personal computers can store terabytes (trillions of bytes) of information.

Keyboard

A keyboard is a peripheral device that enables a user to input text into a computer or any other electronic machinery. A keyboard is an input device and is the most basic way for the user to communicate with a computer. This device is patterned after its predecessor, the typewriter, from which the keyboard inherited its layout, although the keys or letters are arranged to function as electronic switches. The keys include punctuation, alphanumeric and special keys like the Windows key and various multimedia keys, which have specific functions assigned to them.

Monitor

A computer monitor is an output device that displays information in pictorial or text form. A monitor usually comprises a visual display, some circuitry, a casing, and a power supply. The display device in modern monitors is typically a thin film transistor liquid crystal display (TFT-LCD) with LED backlighting having replaced cold-cathode fluorescent lamp (CCFL) backlighting. Previous monitors used a cathode ray tube (CRT) and some Plasma (also called Gas-Plasma) displays. Monitors are connected to the computer via VGA, Digital Visual Interface (DVI), HDMI, DisplayPort, USB-C or other proprietary connectors and signals.

RAM

RAM (Random Access Memory) is the hardware in a computing device where the operating system (OS), application programs and data in current use are kept so they can be quickly reached by the device's processor. RAM is the main memory in a computer. It is much faster to read from and write to than other kinds of storage, such as a hard disk drive (HDD), solid-state drive (SSD) or optical drive.

Random Access Memory is volatile. That means data is retained in RAM as long as the computer is on, but it is lost when the computer is turned off. When the computer is rebooted, the OS and other files are reloaded into RAM, usually from an HDD or SSD.

5.2.3 Database Description

A database is an organized collection of data, generally stored and accessed electronically from a computer system. In this system, we use a MySQL database for storing the date and retrieving the data. We can use any database to store this data. The

user can check the details by logging into the SQLyog Enterprice. The data is stored in the MySQL once the user registered.

5.2.4 Sample Code

```
import pandas as pd
from flask import Flask,render_template,request
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
import os
from sklearn.metrics import accuracy_score, silhouette_score
from sklearn.tree import DecisionTreeClassifier
from xgboost import XGBClassifier
from sklearn.ensemble import AdaBoostClassifier
from sklearn.decomposition import PCA
from sklearn.discriminant_analysis import LinearDiscriminantAnalysis
from sklearn import svm
from sklearn.cluster import KMeans
import mysql
from mysql.connector import cursor
mydb = mysql.connector.connect(host='localhost', user='root', password='',
port='3306', database='diabetes')
app = Flask(__name__)
df = pd.read_csv(r'pima-data.csv')
encoder = LabelEncoder()
df['diabetes'] = encoder.fit_transform(df['diabetes'])
df['bmi'] = df['bmi'].apply(np.int64)
df['diab_pred'] = df['diab_pred'].apply(np.int64)
df['skin'] = df['skin'].apply(np.int64)
global  x_test, x_train, y_test, y_train
x = df.iloc[:, :-1]
y = df.iloc[:, -1]
pca = PCA(n_components=8)
pca.fit(x)
```

```
x_pca = pca.transform(x)
m = x_pca
n = y
m_train, m_test, n_train, n_test = train_test_split(m, n, test_size=0.33,
random_state=32)
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.33,
random_state=32)
@app.route('/')
def index():
    return render_template('index.html')
@app.route("/about")
def about():
    return render_template('about.html')
@app.route("/login", methods=['GET', 'POST'])
def login():
    if request.method == "POST":
        email = request.form['email']
        password = request.form['password']
        sql = "SELECT * FROM dc WHERE Email=%s and Password=%s"
        val = (email, password)
        cur = mydb.cursor()
        cur.execute(sql, val)
        results = cur.fetchall()
        mydb.commit()
        if len(results) >= 1:
            return render_template('loginhomepage.html', msg='success')
        else:
            return render_template('login.html', msg='fail')
    return render_template('login.html')
@app.route("/Register", methods=['GET', 'POST'])
def Register():
    if request.method == "POST":
        name = request.form['name']
        email = request.form['email']
```

```
        psw = request.form['psw']
        cpsw = request.form['cpsw']
        if psw == cpsw:
            sql = 'SELECT * FROM dc'
            cur = mydb.cursor()
            cur.execute(sql)
            all_emails = cur.fetchall()
            mydb.commit()
            all_emails = [i[2] for i in all_emails]
            if email in all_emails:
                return render_template('Register.html', msg='exists')
            else:
                sql = 'INSERT INTO dc(Email,Password) values(%s,%s)'
                cur = mydb.cursor()
                values = (email, psw)
                cur.execute(sql, values)
                mydb.commit()
                cur.close()
                return render_template('Register.html', msg='Success')
        else:
            return render_template('Register.html', msg='Mismatch')
    return render_template('Register.html')
@app.route("/uploaddata", methods=['GET', 'POST'])
def uploaddata():
    if request.method == "POST":
        file = request.files['file']
        print(file)
        global df
        df = pd.read_csv(file)
        print(df)
        return render_template('uploaddata.html',msg='Success')
    return render_template('uploaddata.html')
@app.route("/viewdata")
def viewdata():
```

```
    print(df)
    a = df
    print(a)
    return render_template('viewdata.html', cols=a.columns.values,
rows=a.values.tolist())
    return render_template('viewdata.html')
@app.route("/loginhomepage")
def loginhomepage():
    return render_template('loginhomepage.html')
@app.route("/training", methods=['GET', 'POST'])
def training():
    x = df.iloc[:, :-1]
    y = df.iloc[:, -1]
    print(x.shape)
    pca = PCA(n_components=9)
    pca.fit(x)
    x_pca = pca.transform(x)
    m = x_pca
    n = y
m_train, m_test, n_train, n_test = train_test_split(m, n, test_size=0.33,
random_state=32)
    if request.method == "POST":
        model = request.form['algo']
        if model == "1":
            print('kkkkkkkk')
            lda = LinearDiscriminantAnalysis()
            lda.fit_transform(m, n)
            y_preds = lda.predict(m_test)
            z=accuracy_score(n_test,y_preds)*100
            print(z)
            ldadc = 'Accuracy of  LinearDiscriminantAnalysis :' + str(z)
            return render_template('training.html', msg=ldadc)
        elif model == "2":
            print('eeeeeee')
```

```
        km = KMeans(n_clusters=2, random_state=42)
        km.fit(m)
        kmc = km.predict(m)
        print('qqqqqq')
        score = silhouette_score(m, km.labels_, metric='euclidean')*100
        print(score)
        kmdc = 'Accuracy of KMeans  :  ' + str(score)
        return render_template('training.html', msg=kmdc)
    elif model=='3':
        clf = svm.SVC()
        print('dddddd')
        clf.fit(m_train, n_train)
        y_pred = clf.predict(m_test)
        m=accuracy_score(n_test, y_pred) * 100
        print(m)
        svcdc = 'Accuracy of SVM :  ' + str(m)
        return render_template('training.html', msg=svcdc)
     elif model=='5':
        dt = DecisionTreeClassifier(random_state=10)
        dt.fit(x_train, y_train)
        xyz = dt.predict(x_test)
        dtc = accuracy_score(y_test, xyz)*100
        print(dtc)
        dtdc = 'Accuracy of DecisionTreeClassifier :  ' + str(dtc)
        return render_template('training.html', msg=dtdc)
    elif model == '6':
        ab = AdaBoostClassifier(random_state=10)
        ab.fit(x_train, y_train)
        zxc = ab.predict(x_test)
        abc = accuracy_score(y_test, zxc)*100
        print(abc)
        abdc = 'Accuracy of AdaBoostClassifier :  ' + str(abc)
        return render_template('training.html', msg=abdc)
    else :
```

```
            return render_template('training.html', msg="Please select a model")
    return render_template('training.html')
@app.route('/detection', methods=['GET','POST'])
def detection():
    if request.method == "POST":
        numpreg=request.form['numpreg']
        print(numpreg)
        glucoseconc = request.form["glucoseconc"]
        print(glucoseconc)
        diastolicbp = request.form["diastolicbp"]
        print(diastolicbp)
        thickness = request.form["thickness"]
        print(thickness)
        insulin = request.form["insulin"]
        print(insulin)
        bmi = request.form["bmi"]
        print(bmi)
        diabpred = request.form["diabpred"]
        print(diabpred)
        age = request.form["age"]
        print(age)
        skin = request.form["skin"]
        print(skin)
        mna=[numpreg, glucoseconc, diastolicbp, thickness, insulin, bmi, diabpred, age,
skin]
        model = AdaBoostClassifier(random_state=10)
        model.fit(x_train, y_train)
        output=model.predict([mna])
        print(output)
        if output==0:
            msg = '<span style = color:black;>The Patient is <span style = color:red;>not
a Diabetic</span></span>'
        else:
            msg = '<span style = color:black;>The Patient <span style = color:red;>has
```

```
Diabetes</span></span>'
        return render_template('detection.html',msg=msg)
    return render_template('detection.html')
if __name__ == '__main__':
app.run(debug=True)
```

5.3 SYSTEM TESTING

5.3.1 Software Testing

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies, assemblies and/or a finished product. It is the process of exercising software with the intent of ensuring that the software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

5.3.2 Types of Testing

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.

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.

Functional Testing

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.

System Testing

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

White Box Testing

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

Black Box Testing

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document.

5.3.2.1 Unit Testing

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

Test strategy and approach

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

Test objectives

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

Features to be tested

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

5.3.2.2 Integration Testing

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

Test Results:

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

5.3.2.3 Acceptance Testing

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user.

It also ensures that the system meets the functional requirements.

Test Results:

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

5.3.3 Test Cases

Test Case 1

Table 5.1: *Test Case 1*

SNO	Test Case ID	Diabetes Prediction
1	**Precondition**	User executes the code.
2	**Description**	Prediction of Diabetes using machine learning algorithms.
3	**Test Steps**	1. First enter the Registration / login details. 2. Then load the pima dataset. 3. Train the system with given dataset. 4. View the accuracy. 4. Enter patient details. 5. Find the whether the person has diabetes or not.
4	**Expected Output**	The patient is not a diabetic.
5	**Actual Output**	The patient is not a diabetic.
6	**Status**	PASS
7	**Remarks**	-

Test Case 2

Table 5.2: *Test Case 2*

SNO	Test case ID	User Registration and Login
1	**Precondition**	User executes the code.
2	**Description**	Register and login.
3	**Test Steps**	1. User enter the details. 2. Submit the details.
4	**Expected Output**	Success.
5	**Actual Output**	Success.
6	**Status**	PASS
7	**Remarks**	-

CHAPTER 6

RESULTS

6.1 EXECUTION PROCEDURE

1. Make sure PyCharm IDE is already installed in our system.
2. Install the python latest version for windows.
3. Create a Project.
4. Build an application.
5. Install required packages
7. After that run the application in PyCharm IDE.
8. It will generate a link and click it.
9. It will automatically run the application in your web.
10. User register into the diabetes care registration portal.
11. User login into the diabetes care login portal.
12. User upload the PIMA diabetes data set.
13. User view the uploaded diabetes data set.
14. User train the system using machine learning models.
15. User submit the patient details to predict whether the patient has diabetes or not.
16. User logout from the application.

6.2 SCREENSHOTS

- **Home Page**

Figure 6.1: *Home Page.*

Description: Here user view the home page of Diabetes Care web application. In this page the user wants to select whether he is already registered or to register now.

- **About Page**

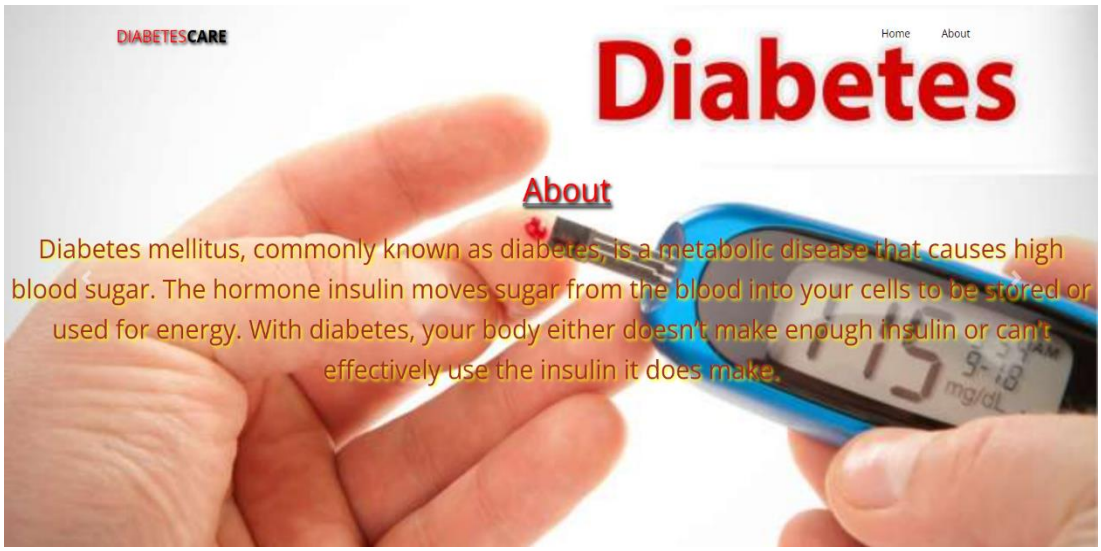


Figure 6.2: *About Page*

Description: In the about page, users can learn more about Diabetes Care and symptoms of the particular disease.

- **Registration Page**

Figure 6.3: *Registration Page*

Description: In this page the user wants to fill the given fields to complete the registration process or Users can register for the Diabetes Care application here.

- **Login Page**

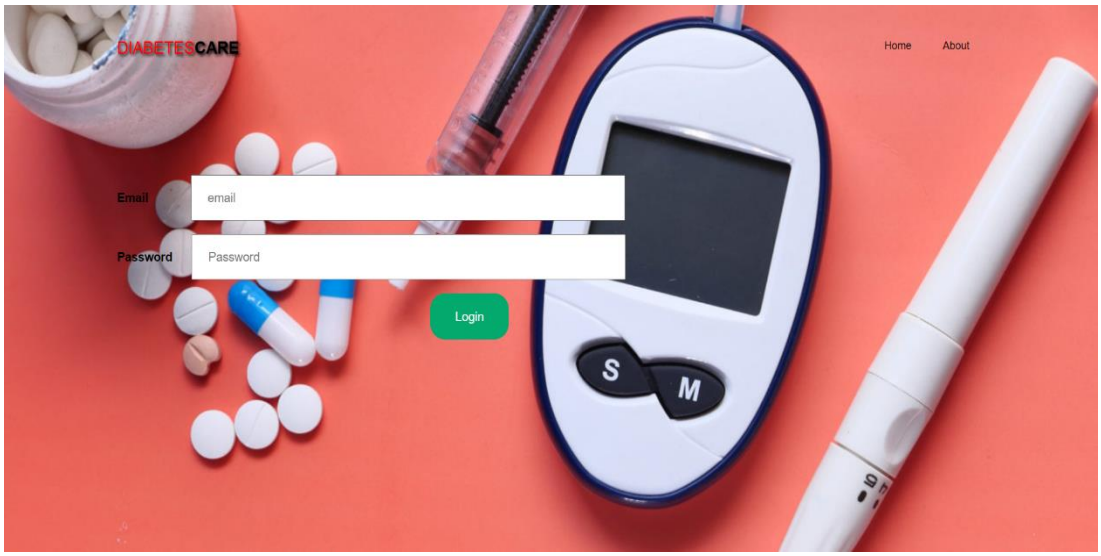


Figure 6.3: *Login Page*

Description: In this page the user wants to enter the email id and password correctly to login to the system.

- **Login Home Page**

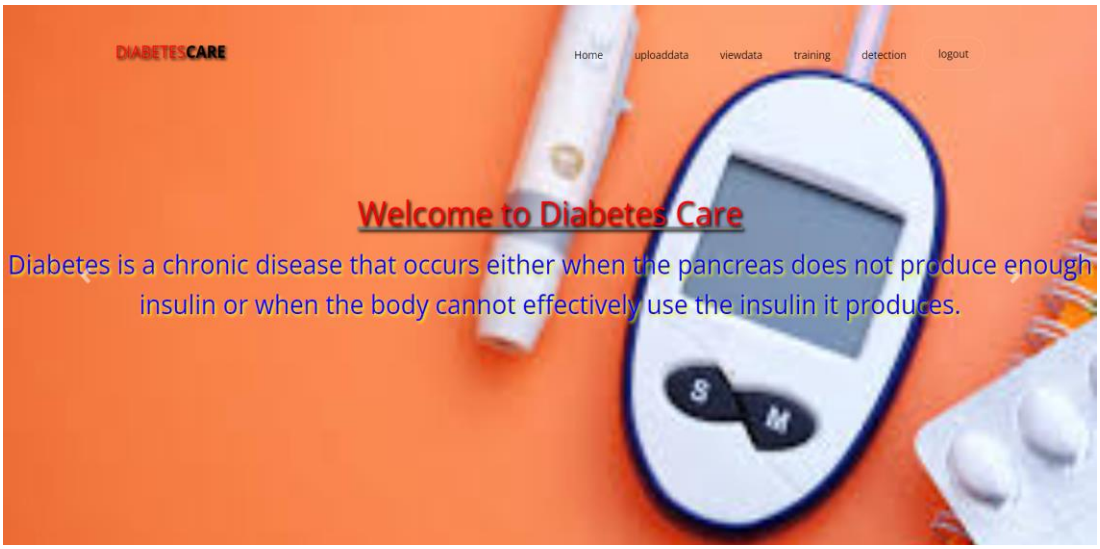


Figure 6.4: *Login Home Page*

Description: After filling the required details we can see this page (User login home page of diabetes care**).**

- **Upload Dataset**

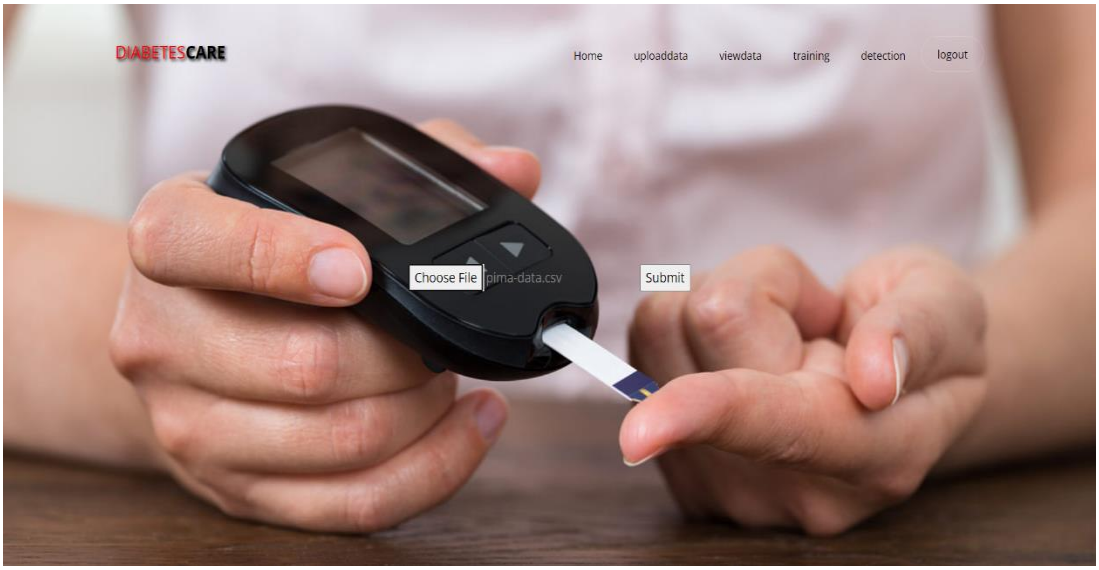
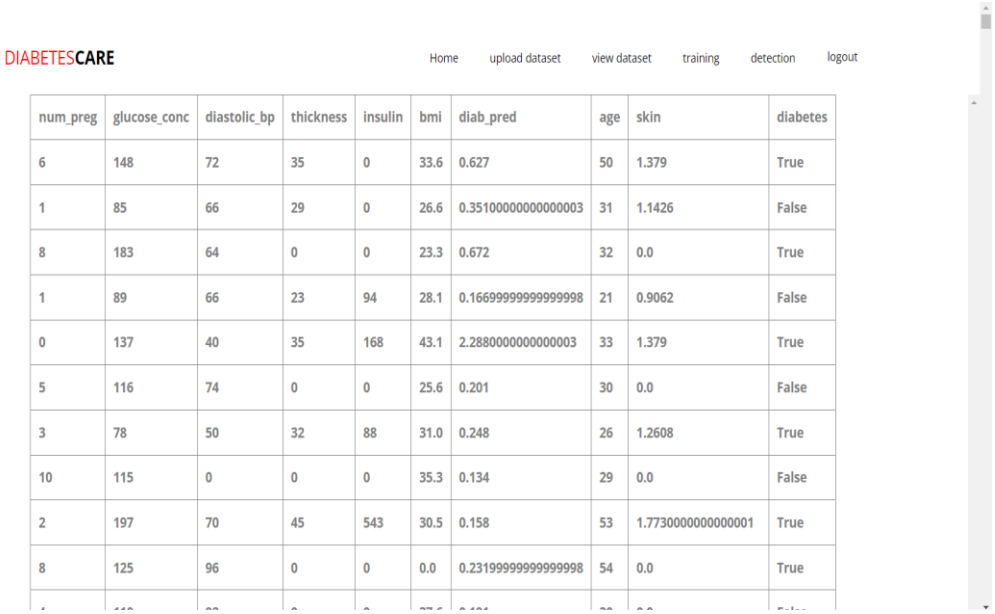


Figure 6.5: *Upload Dataset Page*

Description: We need to upload the dataset for predicting the user details.

- **View Dataset**



num_preg	glucose_conc	diastolic_bp	thickness	insulin	bmi	diab_pred	age	skin	diabetes
6	148	72	35	0	33.6	0.627	50	1.379	True
1	85	66	29	0	26.6	0.35100000000000003	31	1.1426	False
8	183	64	0	0	23.3	0.672	32	0.0	True
1	89	66	23	94	28.1	0.16699999999999998	21	0.9062	False
0	137	40	35	168	43.1	2.2880000000000003	33	1.379	True
5	116	74	0	0	25.6	0.201	30	0.0	False
3	78	50	32	88	31.0	0.248	26	1.2608	True
10	115	0	0	0	35.3	0.134	29	0.0	False
2	197	70	45	543	30.5	0.158	53	1.7730000000000001	True
8	125	96	0	0	0.0	0.23199999999999998	54	0.0	True

Figure 6.6: *View Dataset Page*

Description: After uploading the dataset we can see uploaded data set in this page.

- **Training**

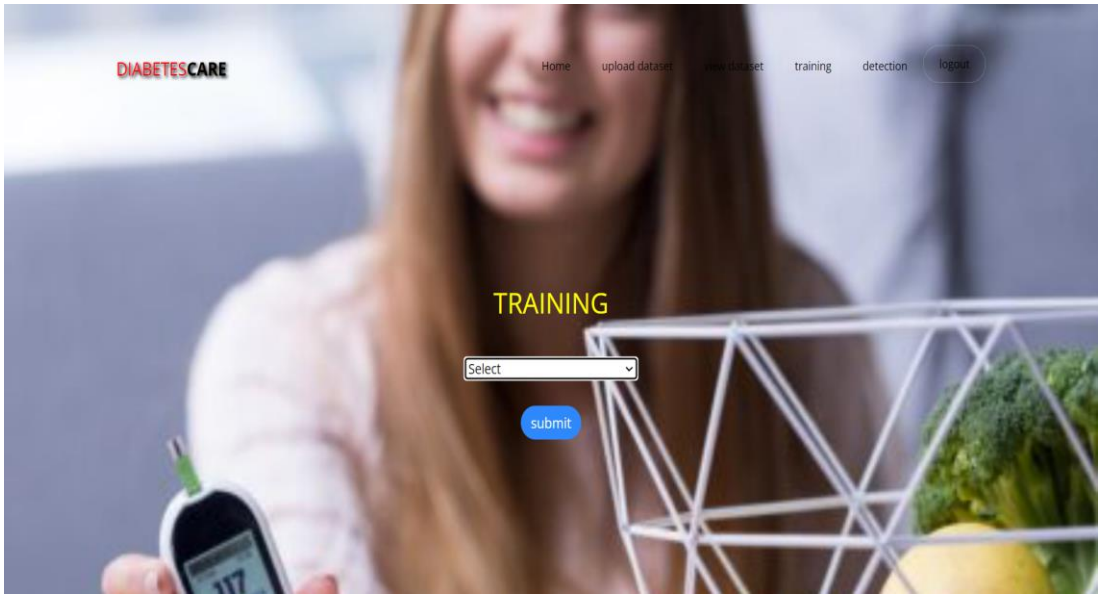
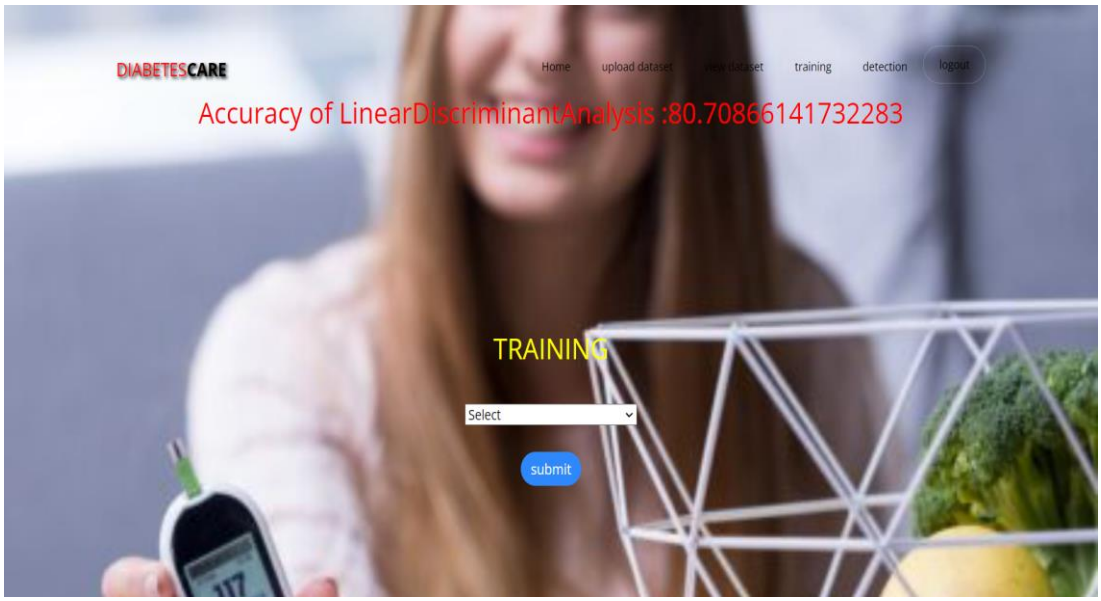


Figure *6.7:* *Training Page*

Description: In this page, we need to enter the value for training.

- **Linear Discriminant Analysis**



- **Figure *6.8:*** *Linear Discriminant Analysis*

Description: In this page, we can select the model and can see the accuracy.

- **SVM**

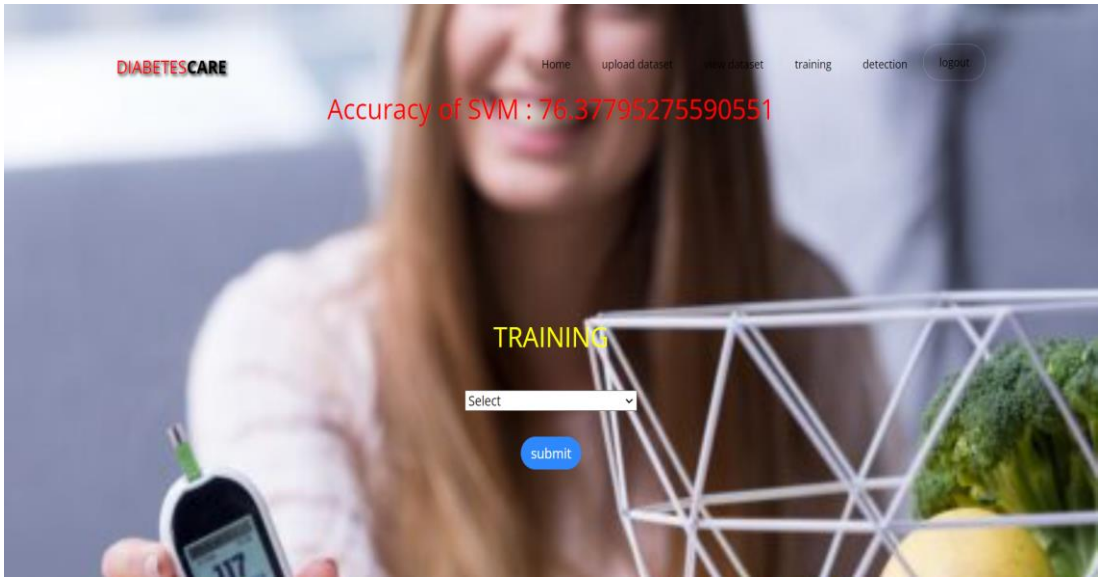


Figure 6.9: *SVM*

Description: In this page, we can select the model and can see the accuracy.

- **AdaBoost**

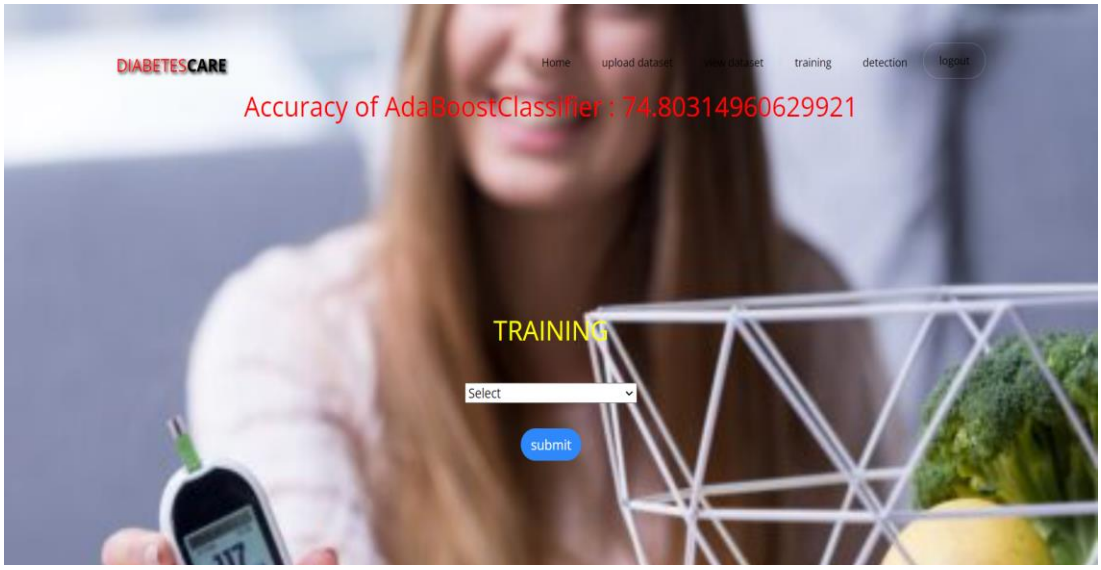


Figure 6.10: *AdaBoost*

Description: In this page, we can select the model and can see the accuracy.

- **Decision Tree**

Figure 6.11: *Decision Tree*

Description: In this page, we can select the model and can see the accuracy.

- **K-Means**

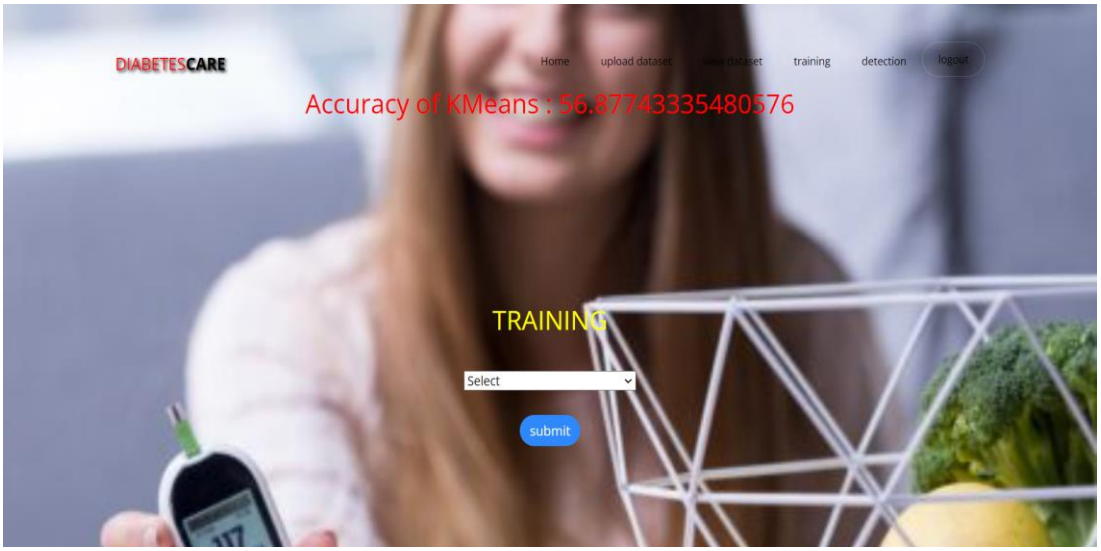


Figure 6.12: *K-Means*

Description: In this page, we can select the model and can see the accuracy.

- **Prediction**

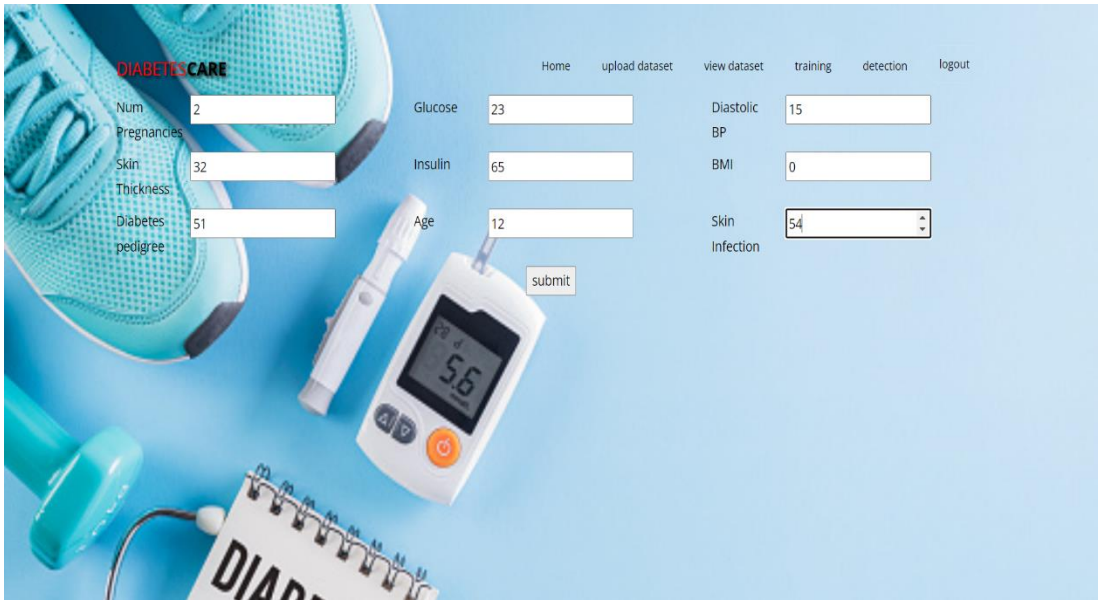


Figure 6.13: *Prediction Page*

Figure 6.14: *Result Page*

Description: After entering the details of the patient we can predict whether the person is having stroke or not.

CHAPTHER 7
CONCLUSION AND FUTURE ENHANCEMENT

7.1 CONCLUSION

In this system, we studied the current state of the art in order to predict and detect diabetes disease. Diabetes is a chronic disease and must be diagnosed earlier before it could reach a dangerous state. Various Supervised learning algorithms are applied such as SVM, Decision tree, etc. In conclusion, decision tree-based classifiers have the potential to detect the diabetes in early stage. It is clear that the model improves the accuracy and precision of diabetes prediction when combined with an unsupervised learning method such as K-Means. K-Mean and SVM have also diagnosed and evaluated diabetes based on accuracy.

7.2 FUTURE ENHANCEMENT

Using linear discriminant analysis model, we obtained efficient results with improved prediction accuracy. As future research, we are trying to know how many members will get diabetes and percentage of diabetes occurred on patient.

REFERENCES

[1] Chen, M., Malook, T., Rehman, A. U., Muhammad, Y., Alshehri, M. D., Akbar. (2021). Block chain-Enabled healthcare system for detection of diabetes. Journal of Information Security and Applications, 58, 102771.

[2] Kumari, S., Kumar, D., & Mittal, M. (2021). An ensemble approach for classification and prediction of diabetes mellitus using soft voting classifier. International Journal of Cognitive Computing in Engineering,40-46.

[3] Aftab, S., Alanazi, S., Ahmad, M., Khan, M. A., Fatima, A., & Elmitwally, N. S. (2021). Cloud-based diabetes decision support system using machine learning fusion.

[4] Khan, F. A., Zeb, K., Al-Rakhami, M., Derhab, A., & Bukhari, S. A. C. (2021). Detection and prediction of diabetes using data mining: a comprehensive review. IEEE Access, 9, 43711-43735.

[5] García-Ordás, M. T., Benavides, C., Benítez-Andrades, J. A., Alaiz-Moretón, H., & Garcia-Rodríguez, I. (2021). Diabetes detection using deep learning techniques with oversampling and feature augmentation. Computer Methods and Programs in Biomedicine, 202, 105968.

[6] Alshammari, R., Atiyah, N., Daghistani, T., & Alshammari, A. (2020). Improving Accuracy for Diabetes Mellitus Prediction by Using Deep net. Online Journal of Public Health Informatics, 12(1).

[7] Naz, H., & Ahuja, S. (2020). Deep learning approach for diabetes prediction using PIMA Indian dataset. Journal of Diabetes & Metabolic Disorders, 19(1), 391-403.

[8] Swarna, S. R., Boyapati, S., Dixit, P., & Agrawal, R. (2020, December). Diabetes prediction by using Big Data Tools and Machine Learning Approaches. In 2020 3rd International Conference on Intelligent Sustainable Systems (ICISS) (pp. 750-755). IEEE.

[9] Bhatia, M., Kaur, S., Sood, S. K., & Behal, V. (2020). Internet of the things-inspired healthcare system for urine-based diabetes prediction. Artificial Intelligence in Medicine, 107, 101913.

[10] Ayad, M., Kanaan, H., & Ayache, M. (2020, November). Diabetes disease prediction using artificial intelligence. In 2020 21st International Arab Conference on Information Technology (ACIT) (pp. 1-6). IEEE.

[11] Hasan, M. K., Alam, M. A., Das, D., Hossain, E., & Hasan, M. (2020). Diabetes prediction using ensembling of different machine learning classifiers. IEEE Access, 8, 76516-76531.

[12] Anand, A., & Shakti, D. (2015, September). Prediction of diabetes based on personal lifestyle indicators. In 2019 1st International Conference on Next Generation Computing Technologies (NGCT) (pp. 673-676). IEEE.

[13] VijiyaKumar, K., Lavanya, B., Nirmala, I., & Caroline, S. S. (2019, March). Random forest algorithm for the prediction of diabetes. In 2019 IEEE international conference on system, computation, automation and networking (ICSCAN) (pp. 1-5). IEEE.

[14] Gyuk, P., Vassányi, I., & Kósa, I. (2019). Blood glucose level prediction for diabetics based on nutrition and insulin administration logs using personalized mathematical models. Journal of Healthcare Engineering, 2019.

[15] Alam, T. M., Iqbal, M. A., Ali, Y., Wahab, A., Ijaz, S., Baig, T. I & Abbas, Z. (2019). A model for early prediction of diabetes. Informatics in Medicine Unlocked, 16, 100204.

[16] Rumbold, J. M., O'Kane, M., Philip, N., & Pierscionek, B. K. (2020). Big Data and diabetes: the applications of Big Data for diabetes care now and in the future. Diabetic Medicine, 37(2), 187-193.

[17] Bisandu, D. B., Datiri, D. D., Onokpasa, E., Thomas, G., Haruna, M. M., Aliyu, A., & Yakubu, J. Z. (2019). Diabetes prediction using data mining techniques. International Journal of Research and Innovation in Applied Science (IJRIAS), 4(6), 103-111.

[18] Sarwar, M. A., Kamal, N., Hamid, W., & Shah, M. A. (2018, September). Prediction of diabetes using machine learning algorithms in healthcare. In 2018 24th international conference on automation and computing (ICAC) (pp. 1-6). IEEE.

[19] Baig, M. M., GholamHosseini, H., Gutierrez, J., Ullah, E., & Linden, M. (2021). Early detection of prediabetes and T2DM using wearable sensors and internet-of-things-based monitoring applications. Applied Clinical Informatics, 12(01), 001-009.

[20] Manna, S., Maity, S., Munshi, S., & Adhikari, M. (2018, September). Diabetes prediction model using cloud analytics. In 2018 International Conference on

Advances in Computing, Communications and Informatics (ICACCI) (pp. 30-36). IEEE.

WEBSITES

[1] https://www.python.org/

[2] https://en.wikipedia.org/wiki/Machine_learning

[3] https://realpython.com/tutorials/flask/

BIBLIOGRAPHY

[1] Alpaydin, E. (2020). Introduction to machine learning. MIT press.

[2] Chopra, D., & Khurana, R. (2023). Introduction to Machine Learning with Python. Bentham Science Publishers.

[3] Barry, P. (2016). Head first Python: A brain-friendly guide. O Reilly Media,Inc.

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:05/Issue:03/March-2023 Impact Factor- 7.868 www.irjmets.com

MACHINE LEARNING FOR DIABETES PREDICTION

Mr M Manivannan [*1], T Balaji[*2], K M Thanuja[*3],G Janani[*4], D Himanth[*5], R Poojith[*6]

[*1] Associate Professor, Department of Computer Science and Engineering, Siddartha Institute of Science and Technology, Puttur, Andhra Pradesh, India.

[*2,3,4,5,6] Student, Department of Computer Science and Engineering, Siddartha Institute of Science and Technology, Puttur, Andhra Pradesh, India.

DOI : https://www.doi.org/10.56726/IRJMETS35053

ABSTRACT

Nowadays Poor lifestyle, diet, and work pressure lead the diabetes disease which may cause several fatal health issues like heart attack, strokes, kidney failure, etc. Diabetes mellitus can be effectively managed when caught early with high accuracy. Machine Learning approaches are very effective in the early detection and prediction of diabetes. The goal of this system is to offer the inclusive examination of the diagnosis of diabetes by supervised and unsupervised ML algorithms. Unsupervised learning techniques such as Support vector machine (SVM), Linear discriminant analysis (LDA), and Decision tree-based algorithms such as AdaBoost have predicted diabetes with high accuracy. Unsupervised learning techniques such as K-Means are also useful in attribute selection and outlier detection from large diabetes datasets. K- Means and SVM have also diagnosed and estimated diabetes with high accuracy as a combination of supervised and unsupervised machine learning techniques.

Keywords: Diabetes prediction, Machine Learning, Decision tree, XGBoost, Linear discriminant analysis, K-means, and Support Vector Machine.

I. INTRODUCTION

Diabetes is one of the most dangerous chronic diseases that could lead to others serious complicating diseases. Diabetes diseases are also called as diabetes mellitus, which describes a set of metabolic disease. Diabetes leads to many other kinds of diseases and that are- heart attack, blindness, kidney diseases and so on. Diabetes is also called as Diabetes Mellitus is a chronic disease and is considered as one of the deadliest diseases.

Diabetes disease can be categorized as Type 1 or Type 2. If the pancreas does not create an adequate amount of insulin in the body, is called as Type 1. In Type 2, the body either cannot effectively use the insulin that it produces or an inadequate amount of insulin is released into the bloodstream Type 1 disease generally occurs in children and adolescents, but it can occur in older people. Type 2 diabetes is normally milder compare to people have type 2 diabetes. Type 1 diabetes can be cured by inserting insulin into the fatty tissue under the skin of the patient. However, Type 2 diabetes can be cured by having a healthy diet, weight, and exercising.

Many diseases can be prevented if diabetes can be diagnosed in the early stages. Early diagnosis and prediction of disease is possible due to recent technological development of IoT, Artificial Intelligence (AI) and Blockchain in the current healthcare system. AI presented a paradigm shift in diabetes care from conservative management approaches to construct the targeted data-driven precision care. IoT offers connected environment to the smart healthcare system. ML and deep learning are AI based techniques. ML has a potential of improving efficiency and decrease the cost of treatment in the healthcare system. Various texts are available for diagnosis and prediction of diabetes based on data mining and ML. Data mining and ML methods are equally important to their specific objective. Data mining techniques are useful to extract rules and pattern from the vast amount of diabetes data set, while ML is significant to learn and automate the machine along with pattern recognition. Several ML techniques are used to form digital support in diabetes care. These include support vector machine (SVM), Decision Tree (DT), random forest (RF), classification and regression trees, Logistic Regression (LR) k-nearest neighbor (KNN), neural network, K-Mean, Principle Component Analysis (PCA) based algorithm for better diabetes care. Various texts have been available for automatic diabetes detection, prediction and management via ML and AI.

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:05/Issue:03/March-2023 Impact Factor- 7.868 www.irjmets.com

II. RELATED WORK

The implementation of machine learning algorithms is a bit complex to build due to the lack of information about data visualization. More Mathematical calculations are used in the existing system for model building this may takes a lot of time and complexity. To overcome all these problems, we use machine learning packages available in the sci-kit-learnn library. Proposed several machine learning models to classify attacks are not, but none have adequately addressed this misdiagnosis problem. That can be used for this purpose is Steven Multi-Parameterer Prediction of Diabetes Empowered With Fused Machine Learning. Also, similar studies that have proposed models for the evaluation of such tumors mostly do not consider the heterogeneity and the size of the data Therefore, we propose a machine learning-based approach which combines a new technique of pre-processing the data for features transformation, SVM, LDA, Decision tree,. ML algorithm gives the best accuracy techniques to eliminate the bias and the deviation of instability.

III. SYSTEM ARCHITECTURE

.

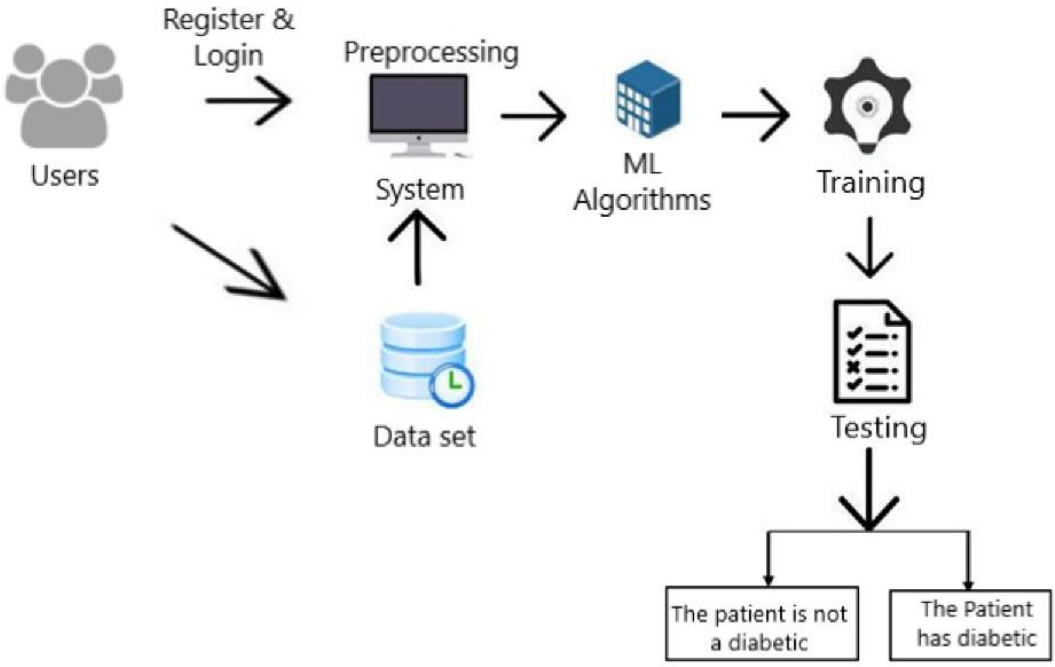


Figure 1: System Architecture

IV. MODULES

Our project has two main modules such as,

1. User
2. System

1. USER

Register:

Users can register for the Diabetes Care application here.

Login:

After registering, the user can access his portal.

Upload dataset:

Here is where the user uploads his dataset.

View data:

The user can view the dataset which he was uploaded

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:05/Issue:03/March-2023 Impact Factor- 7.868 www.irjmets.com

Select Model:

To create a model that predicts disease with better accuracy, this module will help the user.

View accuracy:

Here user views the accuracy of the selected model.

Input Values:

The user must provide input values for certain fields in order to get results.

View Results:

The user views the generated results from the model based on input values.

Logout: After completing all processes, the user can log out from this page.

2. SYSTEM

Working on heart dataset:

The system checks for data whether it is available or not and loads the data in a CSV file.

Training the data:

After pre-processing, the data will be spilt into two parts as train and test data before training with the given algorithms.

Generate accuracy:

We train the machine learning algorithms to improve model accuracy.

Generate Results:

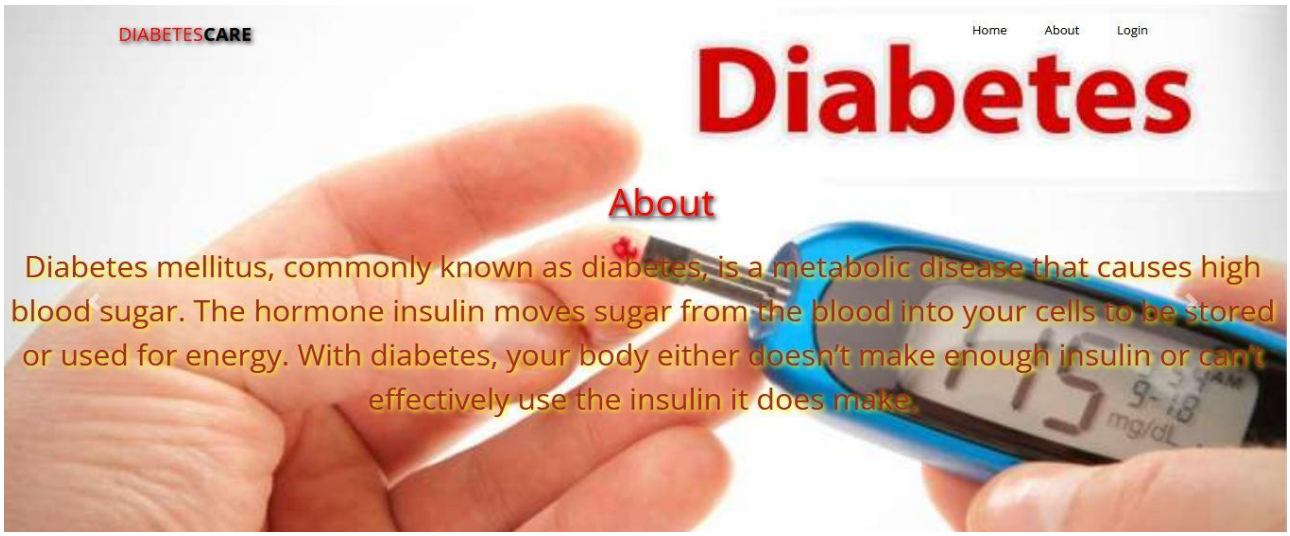
We use a machine learning method to train the algorithm and determine whether or not the patient has diabetes.

V. RESULTS

Home Page:

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:05/Issue:03/March-2023 Impact Factor- 7.868 www.irjmets.com

About Page:



Register page:

Login Page:

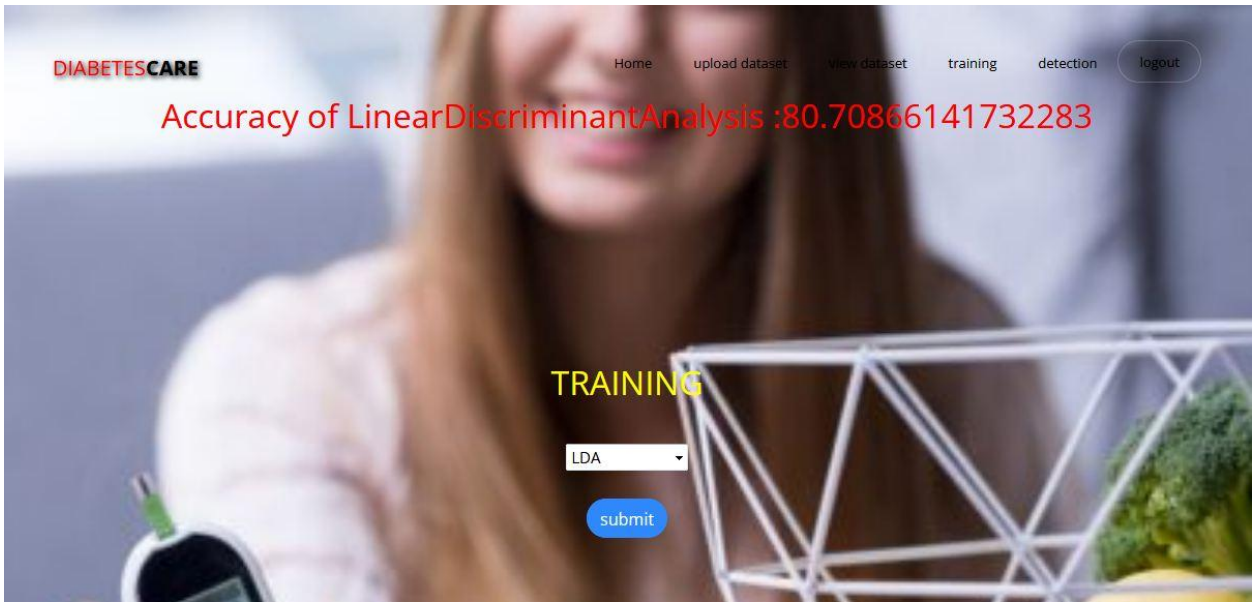
e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:05/Issue:03/March-2023 Impact Factor- 7.868 www.irjmets.com

View data set Page:

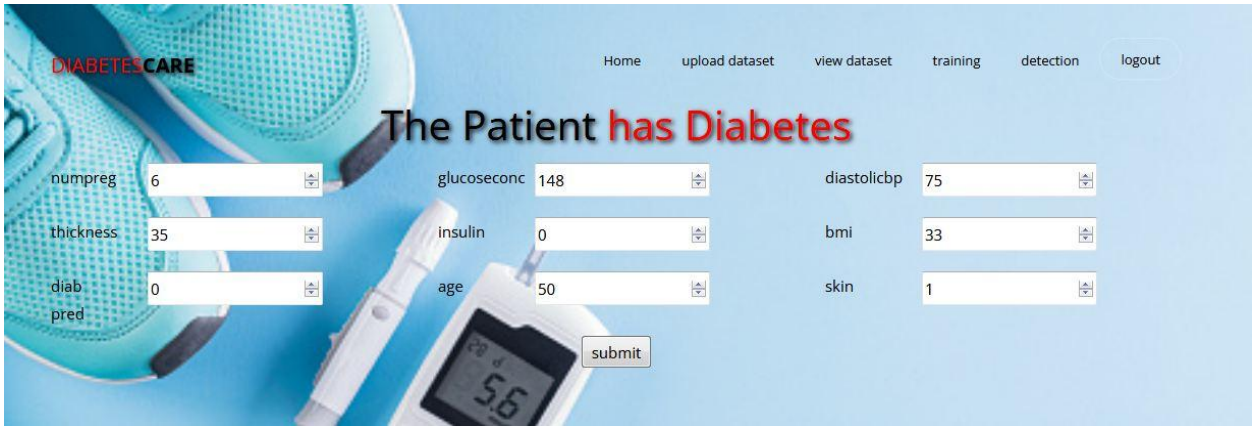
DIABETESCARE Home upload dataset view dataset training detection logout

num_preg	glucose_conc	diastolic_bp	thickness	insulin	bmi	diab_pred	age	skin	diabetes
6	148	72	35	0	33.6	0.627	50	1.379	True
1	85	66	29	0	26.6	0.35100000000000003	31	1.1426	False
8	183	64	0	0	23.3	0.672	32	0.0	True
1	89	66	23	94	28.1	0.16699999999999998	21	0.9062	False
0	137	40	35	168	43.1	2.2880000000000003	33	1.379	True
5	116	74	0	0	25.6	0.201	30	0.0	False
3	78	50	32	88	31.0	0.248	26	1.2608	True
10	115	0	0	0	35.3	0.134	29	0.0	False
2	197	70	45	543	30.5	0.158	53	1.7730000000000001	True

Training page:



Detection Page:



e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:05/Issue:03/March-2023 Impact Factor- 7.868 www.irjmets.com

VI. CONCLUSION

In this system, we studied the current state of the art in order to predict and detect diabetes disease. Diabetes is a chronic disease and must be diagnosed earlier before it could reach a dangerous state. Various Supervised learning algorithms are applied such as SVM, Decision tree, etc. In conclusion, decision tree-based classifiers have the potential to detect the diabetes in early stage. It is clear that the model improves the accuracy and precision of diabetes prediction when combined with an unsupervised learning method such as K-Means. we used the sci-kit-learn library to improve accuracy and above 80% accuracy has been achieved.

VII. REFERENCES

[1] Chen, M., Malook, T., Rehman, A. U., Muhammad, Y., Alshehri, M. D., Akbar, A. & Khan, M. A. (2021). Block chain-Enabled healthcare system for detection of diabetes. Journal of Information Security and Applications, 58, 102771.

[2] Kumari, S., Kumar, D., & Mittal, M. (2021). An ensemble approach for classification and prediction of diabetes mellitus using soft voting classifier. International Journal of Cognitive Computing in Engineering, 2, 40-46.

[3] Aftab, S., Alanazi, S., Ahmad, M., Khan, M. A., Fatima, A., & Elmitwally, N. S. (2021). Cloud-based diabetes decision support system using machine learning fusion.

[4] Khan, F. A., Zeb, K., Al-Rakhami, M., Derhab, A., & Bukhari, S. A. C. (2021). Detection and prediction of diabetes using data mining: a comprehensive review. IEEE Access, 9, 43711-43735.

[5] García-Ordás, M. T., Benavides, C., Benítez-Andrades, J. A., Alaiz-Moretón, H., & Garcia-Rodríguez, I. (2021). Diabetes detection using deep learning techniques with oversampling and feature augmentation. Computer Methods and Programs in Biomedicine, 202, 105968.

[6] Naz, H., & Ahuja, S. (2020). Deep learning approach for diabetes prediction using PIMA Indian dataset. Journal of Diabetes & Metabolic Disorders, 19(1), 391-403.

ANNAMACHARYA INSTITUTE OF TECHNOLOGY & SCIENCES :: TIRUPATI
(Autonomous)

Approved by AICTE, New Delhi & Permanent Affiliation to JNTUA, Anantapuramu.
Two B.Tech Programmes (CSE & ECE) are accredited by NBA, New Delhi.
Accredited by NAAC with 'A' Grade, Bangalore. Accredited by Institution of Engineers (India), KOLKATA.
A-grade awarded by AP Knowledge Mission. Recognized under sections 2(f) & 12(B) of UGC Act 1956.

ICETSM 2 K 2 3

Certificate

This is to Certify that Dr./Mrs./Ms./Mr. T BALAJI has contributed & Presented a Paper Titled Machine Learning for Diabetes Prediction in the 5th International Conference on "**Emerging Trends in Engineering & Technology, Science, Management**" - [ICETSM-2K23], held during 3rd - 4th April 2023 at Annamacharya Institute of Technology & Sciences, Tirupati

Convener **Co-Convener** **Principal**

ANNAMACHARYA INSTITUTE OF TECHNOLOGY & SCIENCES :: TIRUPATI
(Autonomous)

Approved by AICTE, New Delhi & Permanent Affiliation to JNTUA, Anantapuramu.
Two B.Tech Programmes (CSE & ECE) are accredited by NBA, New Delhi.
Accredited by NAAC with 'A' Grade, Bangalore. Accredited by Institution of Engineers (India), KOLKATA.
A-grade awarded by AP Knowledge Mission. Recognized under sections 2(f) & 12(B) of UGC Act 1956.

ICETSM 2K23

Certificate

This is to Certify that Dr./Mrs./Ms./Mr. K M THANUJA

has contributed & Presented a Paper Titled Machine Learning for Diabetes Prediction in the 5th International Conference on "**Emerging Trends in Engineering & Technology, Science, Management**" - [ICETSM-2K23], held during 3rd - 4th April 2023 at Annamacharya Institute of Technology & Sciences, Tirupati

Convener **Co-Convener** **Principal**

ANNAMACHARYA INSTITUTE OF TECHNOLOGY & SCIENCES :: TIRUPATI
(Autonomous)

Approved by AICTE, New Delhi & Permanent Affiliation to JNTUA, Anantapuramu.
Two B.Tech Programmes (CSE & ECE) are accredited by NBA, New Delhi.
Accredited by NAAC with 'A' Grade, Bangalore. Accredited by Institution of Engineers (India), KOLKATA.
A-grade awarded by AP Knowledge Mission. Recognized under sections 2(f) & 12(B) of UGC Act 1956.

ICETSM 2 K 2 3

Certificate

This is to Certify that Dr./Mrs./Ms./Mr. G JANANI

has contributed & Presented a Paper Titled Machine Learning for Diabetes prediction in the 5th International Conference on "**Emerging Trends in Engineering & Technology, Science, Management**" - [ICETSM-2K23], held during 3rd - 4th April 2023 at Annamacharya Institute of Technology & Sciences, Tirupati

Convener **Co-Convener** **Principal**

ANNAMACHARYA INSTITUTE OF TECHNOLOGY & SCIENCES :: TIRUPATI
(Autonomous)

Approved by AICTE, New Delhi & Permanent Affiliation to JNTUA, Anantapuramu.
Two B.Tech Programmes (CSE & ECE) are accredited by NBA, New Delhi.
Accredited by NAAC with 'A' Grade, Bangalore. Accredited by Institution of Engineers (India), KOLKATA.
A-grade awarded by AP Knowledge Mission. Recognized under sections 2(f) & 12(B) of UGC Act 1956.

ICETSM 2 K 2 3

Certificate

This is to Certify that Dr./Mrs./Ms./Mr. D HIMANTH has contributed & Presented a Paper Titled Machine Learning for Diabetes prediction. in the 5th International Conference on "**Emerging Trends in Engineering & Technology, Science, Management**" - [ICETSM-2K23], held during 3rd - 4th April 2023 at Annamacharya Institute of Technology & Sciences, Tirupati

Convener **Co-Convener** **Principal**

ANNAMACHARYA INSTITUTE OF TECHNOLOGY & SCIENCES :: TIRUPATI
(Autonomous)

Approved by AICTE, New Delhi & Permanent Affiliation to JNTUA, Anantapuramu.
Two B.Tech Programmes (CSE & ECE) are accredited by NBA, New Delhi.
Accredited by NAAC with 'A' Grade, Bangalore. Accredited by Institution of Engineers (India), KOLKATA.
A-grade awarded by AP Knowledge Mission. Recognized under sections 2(f) & 12(B) of UGC Act 1956.

ICETSM 2 K 2 3

Certificate

This is to Certify that Dr./Mrs./Ms./Mr. R POOJITH has contributed & Presented a Paper Titled Machine Learning for Diabetes Prediction. in the 5th International Conference on "**Emerging Trends in Engineering & Technology, Science, Management**" - [ICETSM-2K23], held during 3rd - 4th April 2023 at Annamacharya Institute of Technology & Sciences, Tirupati

Convener **Co-Convener** **Principal**