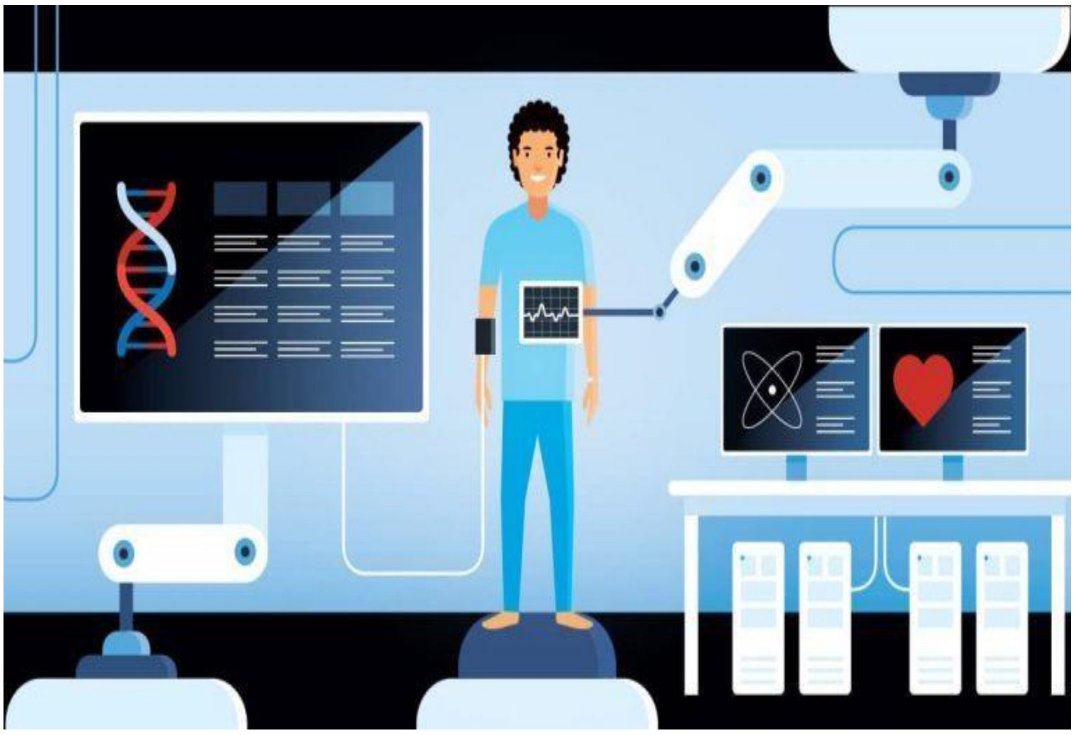
ABSTRACT

Diabetes is caused due to an increase in blood sugar level, and it is considered as one of the deadliest and chronic diseases. If it is untreated or unidentified there will be a chance of occurring many complications. Many complications occur if diabetes remains untreated and unidentified. The tedious identifying process results in the insisting of a patient to a diagnostic center and consulting doctor. But the rise in machine learning approaches solves this critical problem. The motive of this study is to design a model which can prognosticate the likelihood of diabetes in patients with maximum accuracy. Therefore, five machinelearning classification algorithms namely Logistic Regression, Decision Tree, SVM, K- Nearest Neighbor and Random Forest are used in this experiment to detect diabetes at an early stage. Experiments are performed on PIMA Indian diabetes which is sourced from Kaggle machine learning repository. The performances of these two algorithms are evaluated on various measures like Precision, Accuracy and Recall. Accuracy is measuredover correctly and incorrectly classified instances. Results obtained show Random Forest outperforms with the highest accuracy of 86% compared with Logistic Regression algorithm. The model is connecting to web interface for user interaction using Streamlit Diabetes mellitus or simply diabetes is a disease caused due to the increased level of blood glucose. Various traditional methods, based on physical and chemical tests, are available for diagnosing diabetes. However, early prediction of diabetes is quite challenging task for medical practitioners due to complex interdependence on various factors as diabetes affects human organs such as kidney, eye, heart, nerves, foot etc. Machine learning is an emerging scientific field in data science dealing with the ways in which machines learn from experience. The aimof this project is to develop a system which can perform early prediction of diabetes for a patient with a higher accuracy by combining the results of different machine learning techniques. This project aims to predict diabetes via three different supervised machine learning methods including Ensemble learning is a data analysis technique that combines multiple techniques into a single optimal predictive system to evaluate bias and variation, andto improve predictions. To provide all the details of diabetes in a single platform including information of various types of diabetes, causes of diabetes, wellness advice and common medicines used. Diabetes data, which included 17 variables, were gathered from the UCI repository of various datasets.

TABLE OF CONTENTS

Chapter1 INTRODUCTION

Diabetes mellitus or diabetes is one of the incurable chronic diseases caused by lack or absence of a hormone called insulin. It is an essential hormone produced by the pancreas that allows the cells to absorb glucose (blood sugar) from food supplies to provide them the necessary energy. The presence of high blood sugar levels in the blood is known as Hyperglycemia in medical terms. This situation can occur for two main reasons: when the body cannot make insulin required by the blood cells the body cannot respond to insulin properly. The body needs insulin so glucose in the blood can enter the cells of the body whereit can be used for energy. However, if the body fails to utilize glucose to produce energy, it builds up in the blood resulting in hyperglycemia. Fig.1.1,This can cause serious health problems such as diabetic ketoacidosis, nonketotic hyperosmolar, cardiovascular disease, stroke etc. According to the World Health Organization, diabetes is one of the leading causes of death worldwide and about 422 million people worldwide have diabetes. Indeed, it caused the deaths of 1.6 million



people in 2016.

Fig 1.1 Diabetes prediction using machine learning.

There are two main types of diabetes, type1 and type 2. The diabetes type 1 spans 5 to 10% of all diabetes cases. This type of diabetes appears most often during childhood or adolescence and is characterized by the partial functioning of the pancreas. At the beginning, type 1 diabetes does not develop any symptoms, as the pancreas remains partially functional. The disease only becomes apparent when 80-90% of pancreatic insulin-producing cells are already destroyed. The diabetes type 2 presents 90% of all diabetes cases. This type of diabetes is characterized by chronic hyperglycemia and the body's inability to regulate bloodsugar levels, which causes a too high glucose (sugar) level in the blood. This disease usually occurs in older adults and affects more obese or overweight people.

In medicine, doctors and current research confirm that if the disease is discovered at an early stage, the chances of recovery will be greater. With the continuous advancement of technology, machine learning and deep learning techniques have become very useful in early prediction and disease analysis. Among these techniques, Support Vector Machine (SVM), K- Nearest Neighbor (KNN), Decision Tree, the Random Forest (RF) and Logistic Regression are used in this research to predict diabetes. Recently, several research have focused on predicting diabetes using machine learning and deep learning techniques. For instance, authors have proposed a deep learning-based method for diabetes data classification by using the Deep Neural Network (DNN) method. The proposed system was experimented on Pima Indians Diabetes data set. The proposed system was good classification accuracy, precision and recall using Random Forest.

Genetic factors are the main cause of diabetes. It is caused by at least two mutant genes in chromosome 6, the chromosome that affects the response of the body to various antigens. Viral infection may also influence the occurrence of type 1 and type 2 diabetes. Diabetes is one of the most noxious diseases in the world. Diabetes is caused because of obesity, high blood glucose level. It affects the hormone insulin, resulting in abnormal metabolism of carbs and improving the level of sugar in the blood. Approximately 53 million adults (20- 79 years) are living with diabetes. The total number of people living with diabetesis projected to rise to 643 million by 2030 and 783 million by 2045.

The types of Diabetes are Type 1 ,Type 2 and Type 3. Type 1 diabetes means that the immune system is compromised, and the cells fail to produce insulin in sufficient amounts. There are no eloquent studies that prove the causes of type 1 diabetes and there are currently no known methods of prevention. Type 2diabetes means that the cells produce a low quantity of insulin, or the body can’t use the insulin correctly. This is the most common type of diabetes, thus affecting 90% of people diagnosed with diabetes. It is caused by both genetic factors and the manner of living. Type 3Gestational diabetes is a type of diabetes that develops during pregnancy. Gestational diabetes is usually diagnosed in the 24th to 28thweek of pregnancy.

Fig 1.2 Types of Diabetes

Diabetes can also cause vision problems.Fig1.2, It reduces blood glucose level in retina in aged diabetes patients. In future it makes cataracts to the diabetes peoples, and it caused poor vision very easily. Food control is very important, and another important part is physical activity. Diabetic patients must keep the habit of daily exercise. Like brisk walking, bicycling, swimming, housework, gardening etc. Type 2diabetes means that the cells produce a low quantity of insulin, or the body can't use the insulin correctly. This is the most commontype of diabetes, thus affecting 90% of people diagnosed with diabetes. It is caused by bothgenetic factors and the manner of living.

To implement a system (application software) that predicts the results of the diabetes and attempts to improve the accuracy. The main goal of the system is to obtain a high prediction accuracy. Also, to provide an easy-to-use user interface, so that the user can enter the values of each attribute (as input) and obtain a prediction.

Early-stage detection of diabetes is crucial for several reasons. Firstly, it allows for early intervention, which can prevent or delay the onset of complications. This includes making lifestyle changes such as eating a healthy diet, exercising regularly, and managing stress. By intervening early, people with diabetes can manage the condition more effectively, including taking medications as prescribed, monitoring blood sugar levels, and making lifestyle changes

This can reduce the risk of complications, such as nerve damage, kidney disease and heart disease. Secondly, early detection of diabetes can help prevent the condition from developing in people who are at high risk. This includes people who have a family history of diabetes, are overweight or obese, have high blood pressure or high cholesterol, or are over the age of 45. Early detection can lead to preventive measures, including lifestyle changes, and regular monitoring of blood sugar levels, which can prevent or delay the onset of diabetes. Thirdly, early detection of diabetes can lead to cost savings by reducing the need for expensive medical treatments and hospitalizations associated with diabetes-related complications. Early detection can help manage the condition effectively, preventing or delaying complications, thereby reducing healthcare costs. Fourthly, early detection of diabetes can improve the quality of life for people living with the condition. By managing the condition effectively, people with diabetes can reduce the impact of the condition on their daily lives and prevent complications, leading to a better quality of life. In summary, early-stage detection of diabetes is essential for preventing complications, managing the condition, preventing its onset in high-risk individuals, reducing healthcare costs, and improving the quality of life for people living with diabetes. In addition to the above, early detection of diabetes can also lead to improved overall health outcomes for individuals. By identifying the condition early on, people with diabetes can work with their healthcare providers to develop a comprehensive treatment plan that considers their unique needs and circumstances. Early detection of diabetes is, therefore, an important step in ensuring that individuals can maintain good health and quality of life over the long term. If you want to maintain a healthy diet and manage your blood sugar levels, it's important to focus on healthy carbohydrates, fiber-rich foods, heart-healthy fish, and "good" fats. These foods can help improve your overall health and wellbeing while reducing your risk ofdeveloping chronic diseases such as diabetes, heart disease, and stroke. Healthy carbohydrates, such as fruits, vegetables, whole grains, legumes, and low-fat dairy products, provide your body with essentialnutrients and energy without causing a rapid spike in blood sugar levels. These foods are also rich in fiber, which helps regulate digestion and blood sugar levels. Fiber-rich foods, such as vegetables, fruits, nuts, legumes, and whole grains, are an important part of a healthy diet. They help improve digestion, reduce cholesterol levels, and regulate blood sugar levels, making them an excellent choice for people with diabetes. ML is a type of artificial intelligence that uses algorithms to learn patterns in data and make predictions based on that learning. ML has been used in various fields, including healthcare, to analyze large amounts of data and identify patterns that are difficult or impossible for humans to detect.

Fig 1.3 Balanced diet to avoid diabetes

Heart-healthy fish, such as salmon, mackerel, tuna, and sardines, are rich in omega-3 fatty acids, which help reduce inflammation and improve heart health.Fig.1.3, Eating fish at least twice a week can help prevent heart disease and other chronic conditions. Finally, it's important to consume "good" fats in moderation. Foods containing monounsaturated and polyunsaturated fats, such as avocados, nuts, canola oil, olive oil, and peanut oil, can help lower cholesterol levels and improve heart health. However, it's important to keep in mind that all fats are high in calories, so it's important to consume them in moderation as part of a balanced diet.

Testing of diabetes requires visiting to hospitals where they need to give blood sample and wait for a long time to get results. As an alternative to this urine test can also detect diabetes but it involves the same procedure. We have developed a model to detect diabetes using the symptoms of a person. In this model, we have considered sixteen symptoms that a person will showcase while having diabetes. Using these symptoms as input we can be able to detect whether a person is diabetic or not. We have created a platform where people can check for early-stage detection of diabetes without visiting hospitals and even the platform has all the basic information about medicines, food habits and precautionary measures. The advantages of diabetic prediction using ML include early detection and prevention, personalized treatment plans, improved patient outcomes, disease monitoring, and resource allocation. Early detection and prevention can help individuals make lifestyle changes or take medication to reduce their risk of developing diabetes.

1.1 MOTIVATION

The motivation for developing this Early-Stage Diabetic Prediction System and Diabetic Information Platform is to improve the management and treatment of diabetes for individuals in the early stages of the disease. By utilizing machine learning algorithms to predict the likelihood of developing diabetes based on symptom data, healthcare professionals can intervene earlier and potentially prevent the onset of the disease or reduce its severity. Additionally, the creation of a comprehensive online platform for diabetic information, including medication and healthy lifestyle tips, can empower individuals to better manage their diabetes and improve their overall health outcomes. This platform can also serve as a resource for healthcare professionals to access the latest information and treatment options for diabetic patients.

The potential impact of this system is significant, as diabetes is a chronic disease that affects millions of people worldwide and can lead to serious health complications if not managed properly. By providing early detection and access to information and resources, this system has the potential to improve the quality of life for individuals with diabetes and reduce the burden on healthcare systems.

1.2 PROBLEM FORMULATION

The aim of this project is to develop a machine learning model that can accurately predict the likelihood of an individual developing diabetes based on their symptoms. The dataset used in this project consists of various features such as age, gender, polyuria, sudden weight loss, and other symptoms associated with diabetes. In addition, a web-based platform has been developed to provide individuals with diabetes-related information, including medications, healthy tips, and information about various types of diabetes. The challenge here is to collect and analyze large amounts of medical data to develop an accurate predictive model that can help in early- stage diabetic prediction and to create a user-friendly web platform to improve diabetes management and prevention.

1.3 OBJECTIVES

- To understand and analyze the diabetics diseases.

- To identify people who may be at risk of developing diabetes early on based on their symptoms.
- To develop a model that can accurately predict the presence or absence of diabetes based on symptom data.
- To develop a model that can process symptom data quickly and provide timely results to healthcare professionals.
- To identify the most important symptoms that contribute to the prediction of diabetes.
- To provide all the details of diabetes in a single platform including information of various types of diabetes, causes of diabetes, wellness advice and common medicines used.
- To evaluate the performance of the proposed model.

The main goal of this project is to develop a reliable, efficient, and interpretable machine learning model that can help healthcare professionals diagnose and manage diabetes at an early stage based on the symptoms of patients.

1.4 ADVANTAGES

- Early detection: The machine learning model can predict diabetes at an early stage, which can prevent the onset of severe complications.
- Personalized treatment: With the prediction model, the platform can provide personalized treatment recommendations based on the patient's symptoms and history.
- Improved patient outcomes: Early detection and personalized treatment can lead to better health outcomes and improve the quality of life for diabetic patients.
- Accessibility: The web-based platform can be easily accessible to everyone, providing them with relevant information and resources.
- Cost-effective: Early detection and prevention can be more cost-effective than treating advanced stages of diabetes.
- Health education: The platform can provide valuable information to people about the different types of diabetes, healthy habits, and available treatment options.

Overall, the combination of early-stage diabetic prediction using machine learning and a separate platform for diabetic diseases on the web can significantly improve patient outcomes and provide accessible and personalized healthcare.

1.5 DISADVANTAGES

- Lack of access to technology: Individuals who do not have access to technology or who are not comfortable using it may not be able to benefit from the system.
- Limited availability of data: The accuracy of the prediction model is highly dependent on the quality and quantity of data available. If there is limited data available or if the data is not representative of the population, the accuracy of the model may suffer.
- Privacy concerns: The platform may collect sensitive information aboutindividuals, such as their health status and medical history. If the platform does not have adequate security measures in place, this information could be compromised.
- Bias in the model: If the training data used to build the prediction model is biased or incomplete, the model may not accurately represent the population and may result in inaccurate predictions.
- Reliability: The accuracy of the prediction models may be affected by the quality and completeness of the input data, as well as the algorithms used to make predictions. Overall, while these drawbacks must be addressed, the potential benefits of early-stage diabetic prediction and a comprehensive diabetes information platform make it a promising area for future research and development.

1.6 APPLICATIONS

1. Early detection and prevention: ML models can be used to predict the risk of developing diabetes in individuals before they exhibit symptoms or develop complications. This can enable early detection and intervention, such as lifestyle modifications, medication, and regular monitoring, to prevent or delay the onset of diabetes.

2. Personalized treatment plans: ML models can be used to develop personalized treatment plans for patients with diabetes based on their clinical and demographic characteristics. This can help improve patient outcomes by optimizing treatment regimens and reducing the risk of complications.

3. Disease monitoring: ML models can be used to monitor the disease progression of patients with diabetes by tracking changes in their blood glucose levels and other clinical variables over time. This can enable healthcare professionals to adjust treatment plans as needed and identify potential complications before they become severe.

4. Resource allocation: ML models can be used to allocate healthcare resources more efficiently by identifying high-risk populations and targeting interventions to those populations. This can help improve patient outcomes and reduce healthcare costs by avoiding unnecessary hospitalizations and emergency department visits.

5. Patient engagement and education: ML models can be used to provide personalized feedback and recommendations to patients based on their data, helping them better understand their disease and make more informed decisions about their health.

1.7 Chapter Summary

The chapter on early-stage diabetic prediction using machine learning highlights the importance of detecting diabetes at an early stage, as it can prevent severe complications and help patients manage their condition more effectively. The chapter discusses the use of machine learning algorithms to predict diabetes based on symptoms and risk factors. Aseparate platform has been created to provide patients with information on various types of diabetes, medicines used, and healthy tips. However, the chapter also discusses some of the challenges and limitations of using machine learning for diabetes prediction, including the need for large datasets and potential biases in data collection. Overall, the chapter underscores the potential of machine learning to improve early-stage diabetes detection and management, while also highlighting the need for further research and development in this area.

Chapter 2

EXISTING METHOD

Diabetic prediction model achieves higher accuracy, sensitivity, and specificity compared to other models, demonstrating its effectiveness in predicting diabetes. In this literature review, we will discuss the current state-of-the-art research on Diabetic prediction.

2.1 LITERATURE REVIEW

Ahmed, *et.al.,* [1] proposed titled "Prediction of diabetics empowered with fused Machine Learning" explores the use of machine learning techniques for predicting diabetes. The study utilizes a fused machine learning approach, which combines the strengths of several machine learning algorithms to achieve higher accuracy in diabetic prediction. The authors utilize a dataset consisting of various features related to diabetes such as age, BMI, blood pressure, and glucose levels, among others. The authors also employ several preprocessing techniques to prepare the data before feeding it into the machine learning models. The study utilizes several machine learning models such as logistic regression, decision trees, and k-nearest neighbors (KNN), among others, to predict the presence of diabetes. The results of the study show that the fused machine learning approach outperforms individual machine learning models and achieves high accuracy in diabetic prediction. The study contributes to thegrowing body of research in the area of diabetic prediction using machine learning techniquesand highlights the importance of using a fused approach to improve the accuracy of the prediction.

Daliya *et.al.,* [2] proposed an optimized multivariable regression model for the predictive analysis of diabetic disease progression. The study aimed to identify the significant risk factors that contribute to the progression of diabetes and to develop a model that can predict the disease progression accurately. The authors used thedata of 1000 patients with diabetes, including various clinical and laboratory parameters such as age, BMI, HbA1c, blood pressure, and lipid profile. The proposed model was based on the multiple linear regression algorithm, and a stepwise variable selection approach was used to identify the most significant risk factors. The results showed that the proposed model achieved an accuracy of 85%, which outperformed other traditional regression models. The authors concluded that their model could be an effective tool for predicting the disease progression of diabetic patients, and it could assist physicians in making informed decisions regarding patient
management and treatment. The proposed model was based on the multiple linear regression

algorithm, and a stepwise variable selection approach was used to identify the most significant risk factors.

Jiajia Song, *et.al.,* [3] They identified various machine learning techniques used for diabetes prediction, such as decision trees, support vector machines, artificial neural networks, and random forests. The review also highlighted the importance of feature selection and data preprocessing techniques in improving the accuracy of the prediction models. The authors concluded that machine learning models can achieve high accuracy in diabetes prediction when proper feature selection, data preprocessing, and model selection techniques are employed. They also noted that the integration of multimodal data, such as clinical, genetic, and lifestyle data, can further improve the accuracy of the prediction models.Overall, this literature review provides valuable insights into the current state of machine learning applications for diabetes prediction and emphasizes the need for further research in this area.

Hruaping Zhou, *et.al.,* [4] proposes an enhanceddeep neural network (DNN) model for predicting diabetes. The authors suggest that traditional machine learning models have limitations in accurately predicting diabetes due to their inability to extract high-level features from complex datasets. To address this issue, they propose an enhanced DNN model that combines convolutional neural networks (CNNs) and long short-term memory (LSTM) networks to capture both spatial and temporal features from the input data. The model is trained and evaluated on a public diabetes dataset and compared with several state-of-the-art machine learning models. The experimental results show that the proposed model achieves higher accuracy, sensitivity, and specificity compared to other models, demonstrating its effectiveness in predicting diabetes. The authors conclude that the enhanced DNN model has potential in predicting diabetes and could be useful in clinical practice for early diagnosis and management of diabetes.

MD. Kamrul Hasan, *et.al.,* [5] The authors proposed an ensemble-based approach for diabetes prediction using various machine learning classifiers, including K-Nearest Neighbor (KNN), Decision Tree (DT), and Support Vector Machine (SVM). They utilized the Pima Indians Diabetes dataset, which contains various diagnostic measures for diabetes, to train and evaluate the performance of their proposed model. The authors evaluated the performance of their model using various performance metrics such as accuracy, sensitivity, specificity, F1-score, The results showed that the ensemble model outperformedthe individual machine learning algorithms in terms of accuracy, sensitivity, specificity, and F1-score. The ensemble model achieved an accuracy of 79.17%, sensitivity of 79.21%, specificity of

79.03%, and F1-score of 76.13%, which was significantly higher than the individual algorithms.

Yahyauoi, *et.al.,* [6] proposed the focuses on developing a decision support system for predicting diabetes using machine learning and deep learning techniques. The authors identified the importance of early prediction and prevention of diabetes complications, and how machine learning and deep learning techniques can be leveraged for the same. The paper discusses different machine learning and deep learning algorithms and techniques that can be used for diabetes prediction and compares their performance based on metrics such as accuracy, sensitivity, and specificity. The authors then propose a hybrid model that combines different machine learning and deep learning techniques to improve the accuracy of diabetes prediction. The proposed model was evaluated using various performance metrics, and the results demonstrated its superior performance compared to individual machine learning or deep learning techniques. The paper also discusses the potential implications and benefits of using such a system, including the ability to monitor and manage diabetes effectively, reduce the risk of complications, and improve the overall quality of life for diabetic patients.

R. Raj,*et.al.,* [7] focuses on developing an intelligent diabetes detection system using machine learning techniques. The authors conducted a comprehensive literature review on the existing approaches for diabetes detection using machine learning and identified the limitations of these approaches. The paper proposes a novel intelligent diabetes detection system that utilizes an ensemble of machine learning algorithms such as Naive Bayes, Decision Tree,Random Forest, and Support Vector Machines to achieve higher accuracy in diabetes detection. The authors used the Pima Indians Diabetes dataset and conducted experiments to evaluatethe performance of the proposed system. The results show that the proposed system outperforms the individual machine learning algorithms and achieves an accuracy of 96.66%in diabetes detection. This study provides a new direction for the development of intelligent diabetes detection systems using machine learning techniques.

S. T. Mir,*et.al.,*[8] discusses various machine learning algorithms, such as decision trees, artificial neural networks, support vector machines, and random forests, and compares their performance in predicting diabetes. The authors also highlight the importance of feature selection and feature engineering in improving the accuracy of diabetes prediction models. The paper also identifies several future directions in the field of diabetes prediction using machine learning, such as the use of deep learning algorithms, the integration of multiple datasources, and the development of personalized diabetes prediction models. Overall, the paper provides a valuable summary of the current state of research in diabetes prediction using machine learning and suggests several promising avenues for future research.

B. V. Gowtham, *et.al.,* [9] proposed provides a comprehensive review of various machine learning techniques that have been used for predicting diabetes. The authors have discussed the various types of diabetes and the factors that contribute to the development of diabetes. The paper reviews a variety of machine learning models such as logistic regression, decision trees, support vector machines, and neural networks that have been used for predicting diabetes. The authors have compared the performance of these models and discussed the strengths and weaknesses of each model. Additionally, the paper highlights the importance of feature selection in diabetes prediction and provides a detailed discussion of the different feature selection techniques that can be used in machine learning models. The authors also discuss the role of big data analytics in diabetes prediction and highlight some of the challenges associated with diabetes prediction using machine learning techniques. Overall, the paper provides a valuable resource for researchers and practitioners working in the field of diabetes prediction using machine learning techniques.

S. S. Shetty, *et.al.,* [10] the paper provides a comprehensive literature survey of various machine learning techniques used for diabetes prediction. The authors highlight the importance of early detection and management of diabetes and how machine learning algorithms can aid in achieving this goal. The paper presents a comparative analysis ofvarious classification algorithms such as Decision Trees, Random Forest, Support Vector Machines, Naive Bayes, and Artificial Neural Networks, and evaluates their performance on publicly available datasets such as Pima Indian Diabetes, Diabetes 130-US hospitals, and National Health and Nutrition Examination Survey (NHANES) datasets. The authors discuss the strengths and weaknesses of each algorithm and provide insights into the most effective techniques for diabetes prediction. Additionally, the paper discusses the limitations of current studies and highlights the need for further research in this field. Overall, the paper provides a valuable contribution to the literature on diabetes prediction using machine learning techniques.

2.2 CHAPTER SUMMARY

The presented literature review is comprised of ten research papers that discuss the use of machine learning diabetic prediction using symptom-based datasets. Each paper proposes a different machine learning approach to accurately predict diabetes. This chapter provides assistance in the process of developing the prototype by archiving all of the papers that were utilized in this survey for future reference. The findings of this literature review may guide future research and development of early-stage diabetic prediction models using machine learning.

Chapter 3

IMPLEMENTATION

3.1 METHODOLOGY

Diabetic Prediction Using Machine Learning Techniques is a complex process that involves several steps. Data collection is the first step, followed by pre-processing and feature engineering, which help to ensure the accuracy of the data and improve the performance of the machine learning model. Model selection is crucial, as the selected model must be able to handle the specific problem and the available data. The model is trained and validated to ensure that it can generalize to new data and evaluated using metrics such as Mean square root error, Root mean square error, and R-squared. Finally, the model is deployed in a production environment to make predictions on new data.

3.2 BLOCK DIAGRAM

This methodology can be used by prediction of diabetes , and other and make informed decisions about diabetics and Non-diabetics .Fig.3.1,The use of machine learning can also help to automate the process of house price prediction, reduce human error, and provide moreaccurate and timely predictions. However, the success of the methodology depends on the quality of the data, the relevance of the features selected, and the accuracy of the machine learning model.

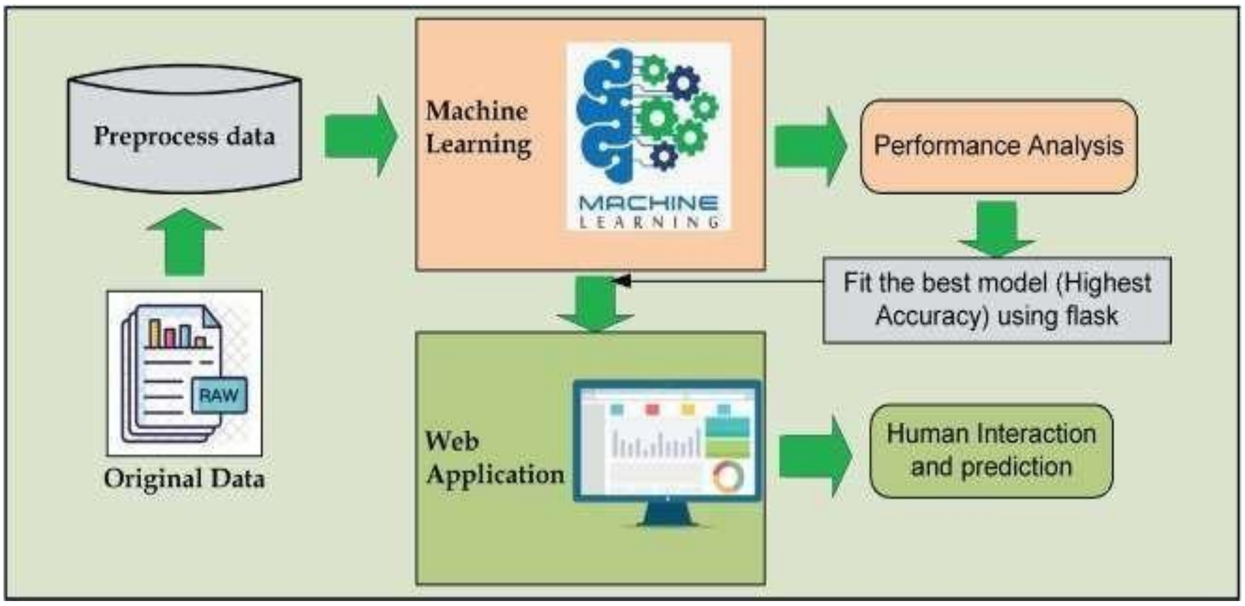


Fig 3.1 Block Diagram for Diabetes

Therefore, it is important to carefully design and implement each step of the methodology to ensure accurate predictions and informed decision-making.

3.3 FLOWCHART

Diabetic Prediction Using Machine Learning Techniques is a complex process that involves several steps.Fig3.2,Data collection is the first step, followed by pre-processing and feature engineering, which help to ensure the accuracy of the data and improve the performance of the machine learning model. Model selection is crucial, as the selected model must be able to handle the specific problem and the available data.

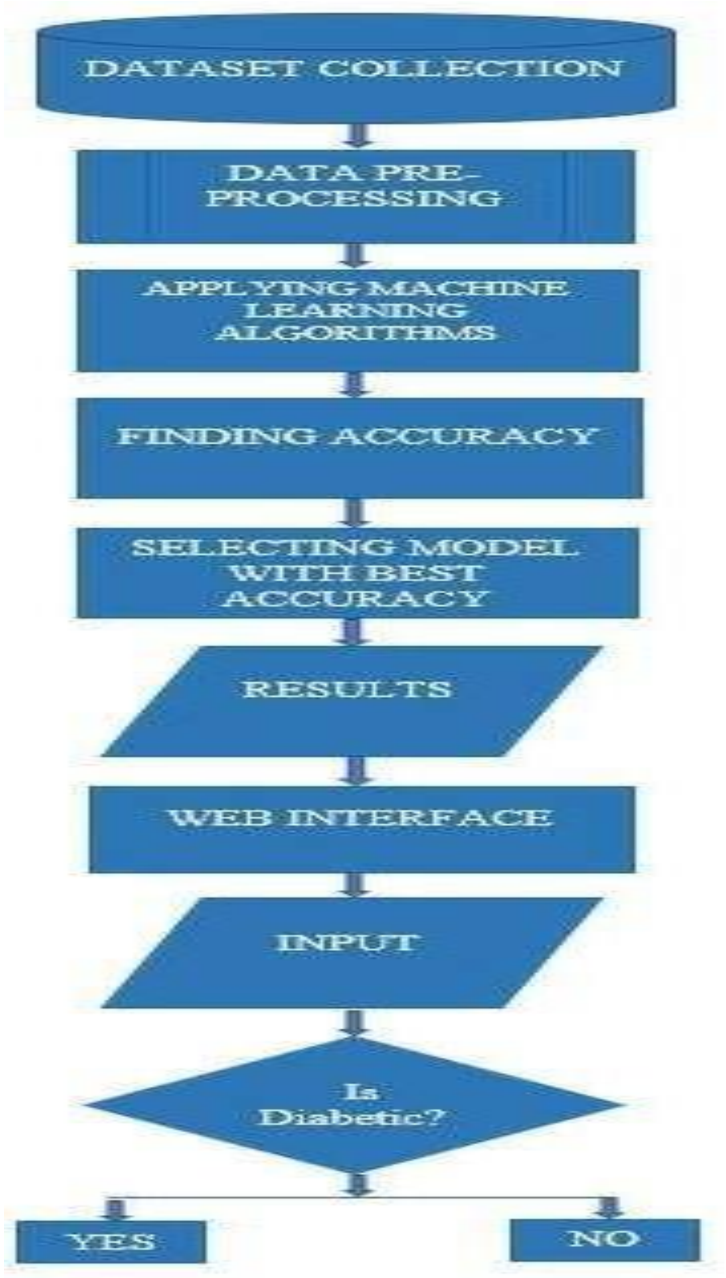


Fig 3.2 Flow Chart

The model is trained and validated to ensure that it can generalize to new data and evaluated using metrics such as Mean square root error, Root mean square error, and R-squared. Finally, the model is deployed in a production environment to make predictions on new data.

3.4 WORKING

3.4.1 System Design

Systems design is the process of defining elements of a system like modules, architecture, components and their interfaces and data for a system based on the specified requirements.It is the process of defining, developing, and designing systems which satisfies the specific needs and requirements of a business or organization.

3.4.2 Use case diagram.

A use case consists of a user and processor where user is used to provide the input to the system and processor is used to process the input data and provide output. The flow is shown in the above diagram fig.3.3.

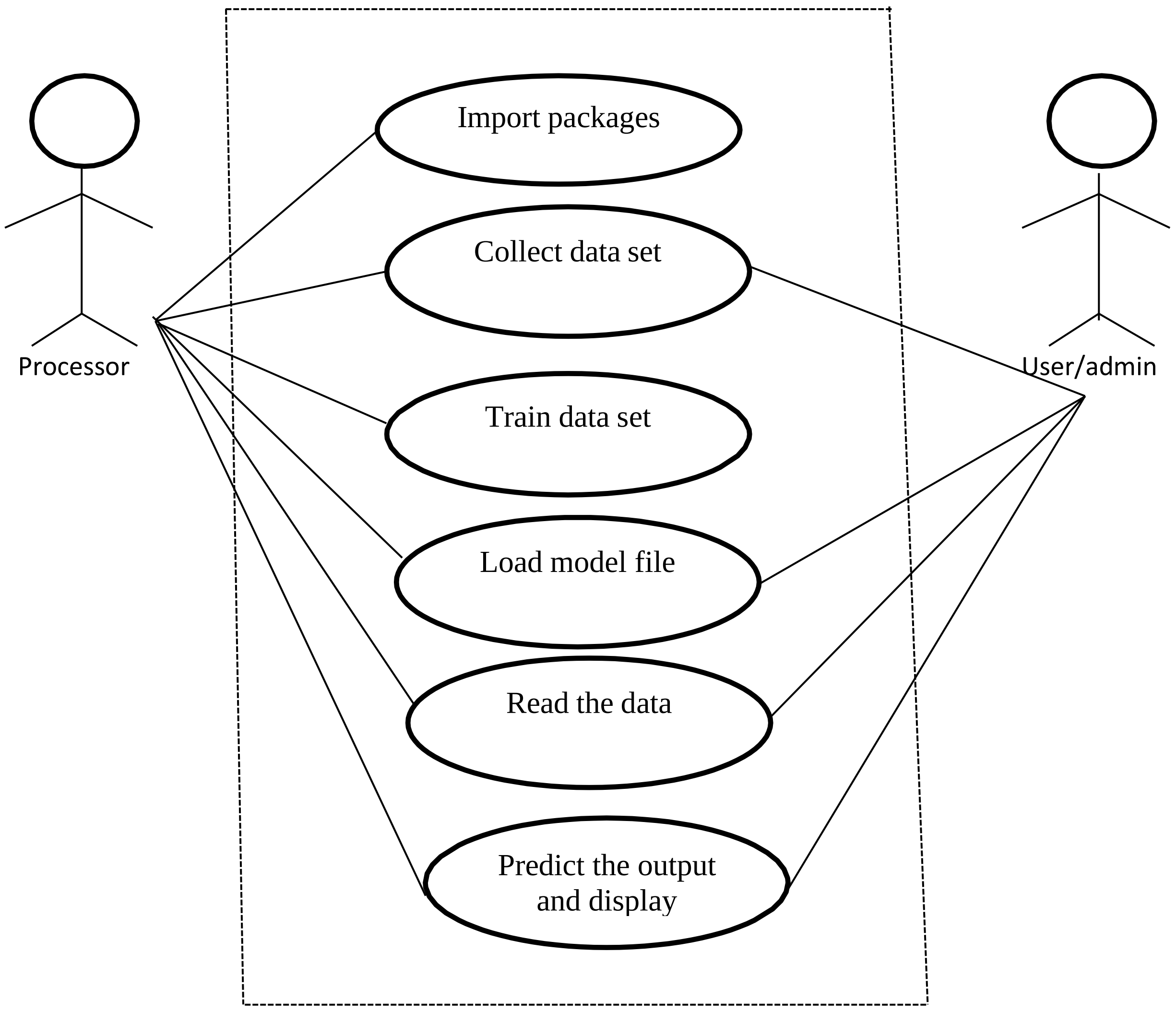


Fig. 3.3 Use Case Diagram

First user has to run the system and run the code, model and library packages are imported and loaded. After the run of code output is displayed according to the data input provided.

3.4.3 Activity diagram

An activity diagram is a behavioral diagram i.e., it depicts the behavior of a system. An activity diagram portrays the control flow from a start point to a finish point showing the various decision paths that exist while the activity is being executed. Firstly, we import all the library package necessary to run the code and for supporting the to code to run error free. As soon as codeis run it provides the desired output.

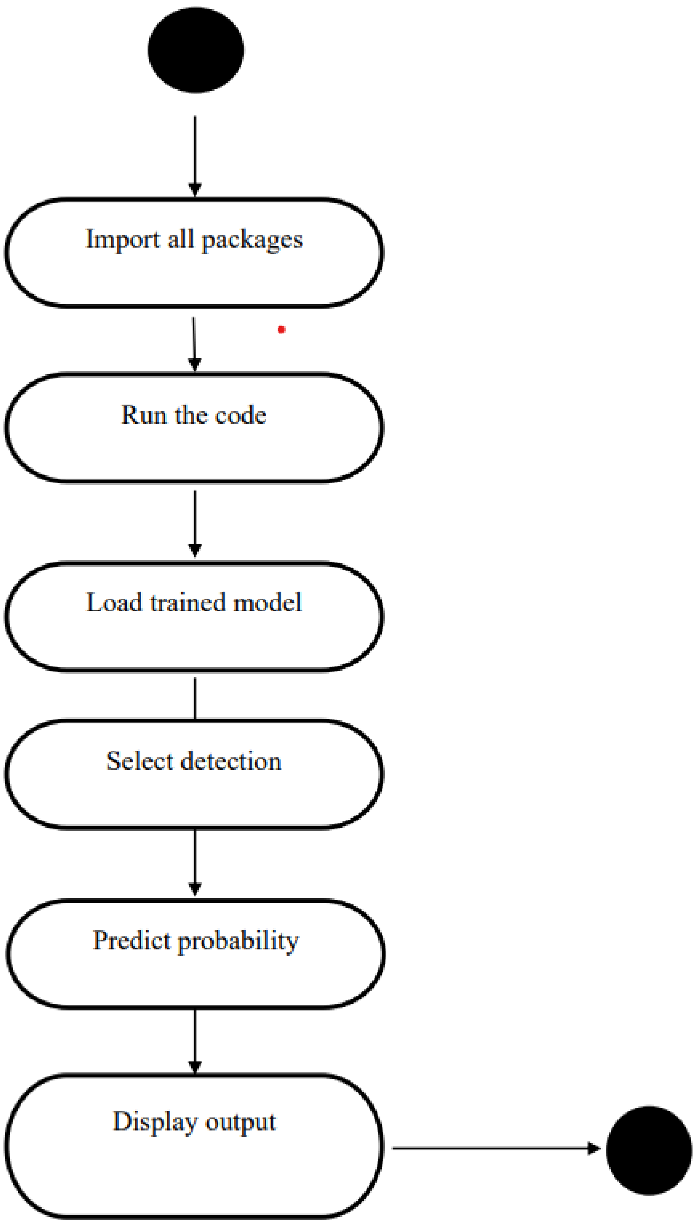


Fig. 3.4 Activity diagram

An activity diagram portrays the control flow from a start point to a finish point showing the various decision paths that exist while the activity is being executed. Firstly, we import all the library packages necessary to run the code and for supporting the to code to run error free. As soonas the code is run it provides the desired output.

3.4.5 PROPOSED METHOD FOR NEW ARCHITECTURE

To propose a new architecture for diabetes prediction, you can follow these general steps:

Define the problem: Clearly specify the problem you aim to address with your proposed architecture. In this case, it is diabetes prediction.

Review existing literature: Conduct a thorough literature review to understand the current state-of-the-art methods and architectures used for diabetes prediction. Identify their strengths, limitations, and areas for improvement.

Identify data requirements: Determine the type and amount of data required for training and evaluating your proposed architecture. This may include medical records, patient demographics, laboratory test results, lifestyle information, and other relevant features.

Design the architecture: Based on your problem understanding and literature review, design a new architecture that addresses the limitations of existing methods. Consider different components such as input representation, feature extraction, model structure (e.g., deep learning, ensemble methods, etc.), and output interpretation.

Data preprocessing: Preprocess the collected data to ensure it is in a suitable format for training your proposed architecture. This may involve steps such as data cleaning, feature scaling, handling missing values, and feature engineering.

Implement the architecture: Write code to implement your proposed architecture using a suitable programming language and framework. Popular choices for deep learning architectures include Python with libraries like TensorFlow, Flask, or Streamlit.

Train and validate the model: Split your dataset into training and validation sets. Train your proposed architecture on the training set and evaluate its performance on the validation set. Experiment with hyperparameters, loss functions, and optimization techniques to improve the model's performance.

Evaluate and compare: Assess the performance of your proposed architecture using appropriate evaluation metrics such as accuracy, precision, recall, F1-score, or area under the receiver operating characteristic curve (AUC-ROC). Compare the results with existing methods to demonstrate the effectiveness of your approach.

3.3.8 MODEL BUILDING

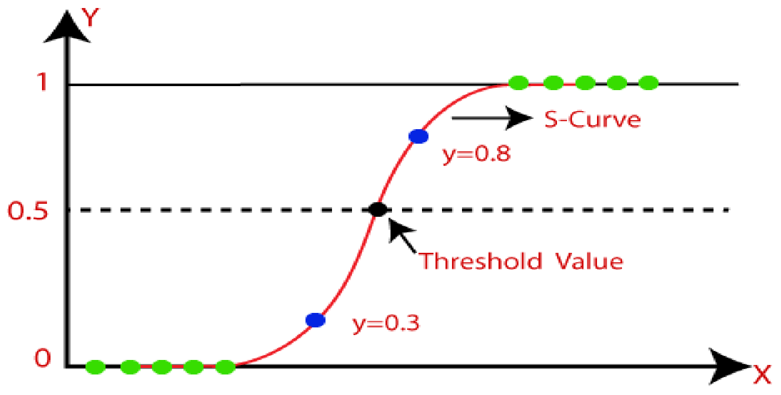
The goal is to develop a predictive model that can make accurate predictions on new data. Algorithm selection involves choosing an appropriate machine learning algorithm basedon the problem being addressed. This can include supervised learning algorithms such as linear regression, logistic regression, decision trees, and neural networks, or unsupervised learning algorithms such as clustering and dimensionality reduction.

a. Logistic Regression

Logistic regression is one of the most popular Machine Learning algorithms, which comes under the Supervised Learning technique. It is used for predicting the categorical dependent variable using a given set of independent variables. Logistic regression predicts the output of a categorical dependent variable. Therefore, the outcome must be a categorical or discrete value. It can be either Yes or No, 0 or 1, true or False, etc. but instead of giving the exact value as 0 and 1, it gives the probabilistic values which lie between 0 and 1.

Logistic Regression is much similar to Linear Regression except that how they are used. Linear Regression is used for solving Regression problems, whereas Logistic regression is used for solving the classification problems. In Logistic regression, instead of fitting a regression line, we fit an "S" shaped logistic function, which predicts two maximum values (0 or 1). The curve from the logistic function indicates the likelihood of something such as whether the cells are cancerous or not, a mouse is obese or not based on its weight, etc. Logistic Regression is a significant machine learning algorithm becauseit has the ability to provide probabilities and classify new data using continuous and discrete datasets. Logistic Regression can be used to classify the observations using different types of data and can easily determine the most effective variables used for the classification. The below image is showing the logistic function:

Fig 3.7 Logistic Regression



Logistic Function (Sigmoid Function):

The sigmoid function is a mathematical function used to map the predicted values to probabilities. It maps any real value into another value within a range of 0 and 1. The value of the logistic regression must be between 0 and 1, which cannot go beyond this limit, so it forms a curve like the "S" form. The S-form curve is called the Sigmoid function or the logistic function. In logistic regression, we use the concept of the threshold value, which defines the probability of either 0 or 1. Such as values above the threshold value tends to 1, and a value below the threshold values tends to 0.

Assumptions for Logistic Regression:

- The dependent variable must be categorical in nature.
- The independent variable should not have multi-collinearity.

Logistic Regression Equation:

The Logistic regression equation can be obtained from the Linear Regression equation. The mathematical steps to get Logistic Regression equations are given below:

- We know the equation of the straight line can be written as:

$$y = b_0 + b_1x_1 + b_2x_2 + b_3x_3 + \ldots\ldots\ldots\ldots\ldots + b_nx_n$$

- In Logistic Regression y can be between 0 and 1 only, so for this let's divide the above equation by (1-y):

$$y/1-y;\ 0 \text{ for } y=0, \text{ and infinity for } y=1$$

- But we need range between -[infinity] to +[infinity], then take logarithm of the equation it will become:

$$\text{Log}[y/1-y] = b_0 + b_1x_1 + b_2x_2 + b_3x_3 + \ldots + b_nx_n$$

The above equation is the final equation for Logistic Regression.

b. DECISION TREE

Decision tree learning or induction of decision trees is one of the predictive modelling approaches used in statistics, data mining and machine learning. It uses a decision tree (asa predictive model) to go from observations about an item (represented in the branches) to conclusions about the item's target value (represented in the leaves). Tree models where the target variable can take a discrete set of values are called classification trees; in these tree structures, leaves represent class labels and branches represent conjunctions of features that lead to those class labels. Decision trees where the target variable can take continuous values (typically real numbers) are called regression trees. Decision trees are among the most popular machine learning algorithms given their intelligibility and simplicity.[1]

In decision analysis, a decision tree can be used to visually and explicitly represent decisions and decision making. In data mining, a decision tree describes data (but the resulting classification tree can be an input for decision making). This page deals with decision treesin data mining

Decision tree is the most powerful and popular tool for classification and prediction. A Decision tree is a flowchart like tree structure, where each internal node denotes a test on an attribute, each branch represents an outcome of the test, and each leaf node (terminal node) holds a class label.

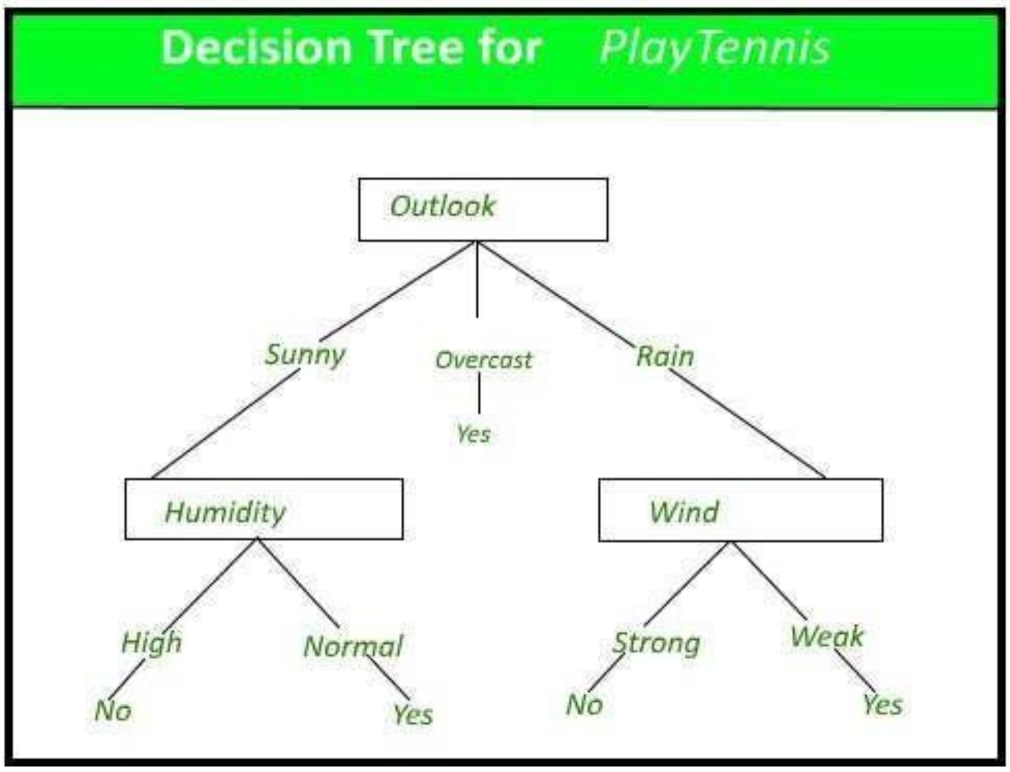


Fig 3.8 Decision Tree

Construction of Decision Tree: A tree can be *"learned"* by splitting the source set into subsets based on an attribute value test. This process is repeated on each derived subset in a recursive manner called *recursive partitioning*. The recursion is completed when the subset at a node all has the same value of the target variable, or when splitting no longer adds value to the predictions. The construction of decision tree classifier does not require any domain knowledge or parameter setting, and therefore is appropriate for exploratory knowledge discovery. Decision trees can handle high dimensional data. In general, decision tree classifierhas good accuracy. Decision tree induction is a typical inductive approach to learn knowledgeon classification.

c. RANDOM FOREST

Random forest or random decision forest are ensemble method for classification, regression and other tasks that operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees. Random decision forests correct for decision trees' habit of overfitting to their training set.

First, Random Forest algorithm is a supervised classification algorithm. We can see it from itsname, which is to create a forest by some way and make it random. There is a direct relationship between the number of trees in the forest and the results it can get: the larger the number of trees, the more accurate the result. But one thing to note is that creating the forest is not the same as constructing the decision with information gain or gain index approach. The author gives 4 links to help people who are working with decision trees for the first time to learn it and understand it well. The decision tree is a decision support tool. It uses a tree- like graph to show the possible consequences. If you input a training dataset with targets and features into the decision tree, it will formulate some set of rules. These rules can be used to perform predictions. The author uses one example to illustrate this point: suppose you want to predict whether your daughter will like an animated movie, you should collect the past animated movies she likes, and take some features as the input. Then, through the decision tree algorithm, you can generate the rules. You can then input the features of this movie and see whether it will be liked by your daughter. The process of calculating these nodes and forming the rules is using information gain and Gini index calculations.

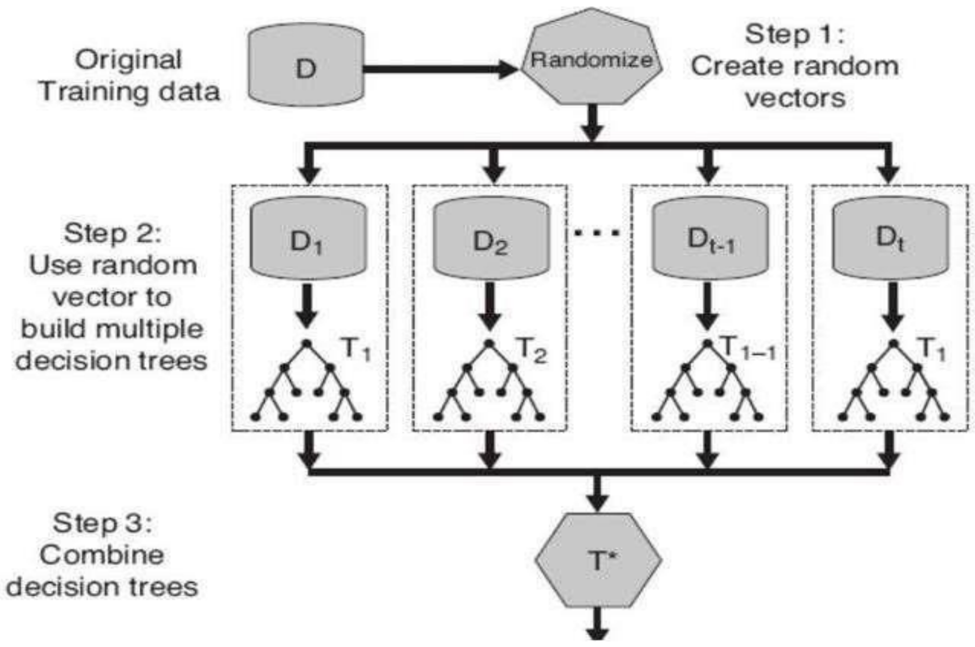

Original Training data
D
Randomize
Step 1: Create random vectors
Step 2: Use random vector to build multiple decision trees
D_1
D_2
D_{t-1}
D_t
T_1
T_2
T_{1-1}
T_1
Step 3: Combine decision trees
T*

Fig 3.9 Random Forest

The difference between Random Forest algorithm and the decision tree algorithm is that in Random Forest, the process es of finding the root node and splitting the feature nodes will run randomly.

Advantages of Random Forest:

1. Random forest can solve both type of problems that is classification and regression and does a decent estimation at both fronts.
2. One of benefits of Random Forest which exists me most is, the power of handle large data sets with higher dimensionality. It can handle thousands of input variables and identity most significant variables, so it is considered as one of the dimensionality reduction methods. Further, the model outputs importance of variable, which can be a very handy feature.
3. It has an effective method for estimating missing data and maintains accuracy when large proportion of the data are missing.
4. It has methods for balancing errors in data sets where classes are imbalanced.

Random forest involves sampling of the input data with replacement called as bootstrap sampling. Here one third of data is not used for training and can be used to testing. These are called the OUT OF BAG samples. Error estimated on these output bag samples is known as out of bag error. Study of error estimates by out of bag, gives evidence to show that the out of bag estimate is as accurate as using a test set of the same size as the training set. Therefore, using the out of bag error estimate removes the need for a set aside test set.

d. SUPPORT VECTOR MACHINE

Support Vector Machine (SVM) is a popular machine learning algorithm used for classification, regression, and outlier detection. The main objective of SVM is to find the hyperplane that separates the data into different classes in the best possible way. In a binary classification problem, SVM algorithm creates a boundary between the two classes by maximizing the margin or distance between the closest data points of each class. The data points closest to the boundary are called support vectors. SVM can also handle multi-class classification problems by using one-vs-all or one-vs-one strategies.

SVM can handle both linear and nonlinear datasets by using different kernel functions such as linear, polynomial, radial basis function (RBF), and sigmoid. The kernel function transforms the data into a higher dimensional space where it is easier to separate the classes. Support Vector Machine (SVM) is a popular machine learning algorithm used for classification, regression, and outlier detection. The main objective of SVM is to find the hyperplane that separates the data into different classes in the best possible way. In a binary classification problem, SVM algorithm creates a boundary between the two classes by maximizing the margin or distance between the closest data points of each class. The data points closest to the boundary are called support vectors. SVM can also handle multi-class classification problems by using one-vs-all or one-vs-one strategies.

SVM can handle both linear and nonlinear datasets by using different kernel functions such as linear, polynomial, radial basis function (RBF), and sigmoid. The kernel function transforms the data into a higher dimensional space where it is easier to separate the classes.

SVM has several advantages such as:

- It can handle high-dimensional datasets and large sample sizes efficiently.
- It is effective in cases where the number of features is larger than the number of samples.
- It can handle both linear and nonlinear datasets using kernel functions.
- It has a regularization parameter that helps prevent overfitting.

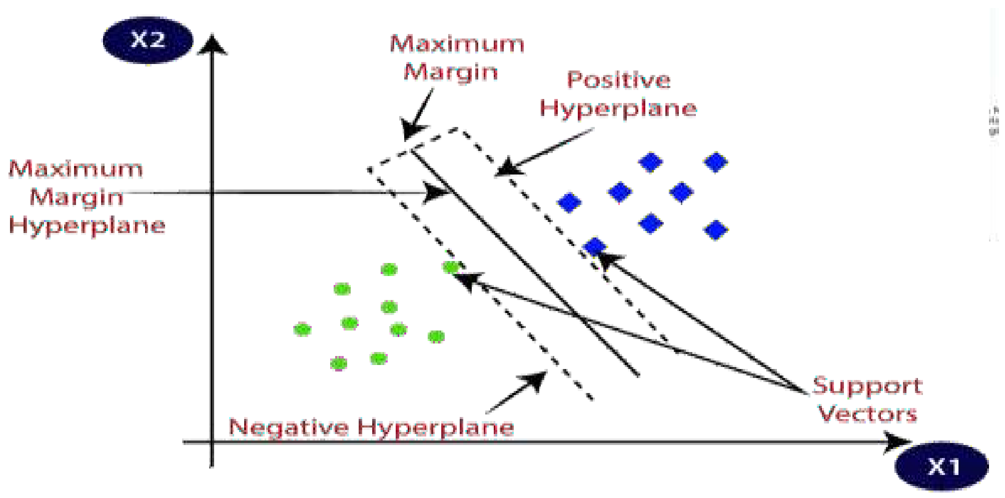


Fig 3.10 Support Vector Machine

However, SVM also has some limitations such as:

- It can be sensitive to the choice of kernel function and hyperparameters.
- It can be computationally expensive for large datasets.
- It may not perform well when the classes are heavily overlapped or imbalanced.

e. K-NEAREST NEIGHBOUR

The k-nearest neighbors (KNN) algorithm is a popular machine learning technique used for classification and regression. It works by finding the k closest data points in the training set to a given input data point and using their labels to make a prediction. To measure the distance between data points, various distance metrics can be used, such as Euclidean distance, Manhattan distance, and Minkowski distance. The optimal value of k can be determined through techniques such as cross-validation, and the choice of k can affect the bias-variance trade-off.

KNN has both strengths and weaknesses. Its simplicity and versatility make it easy to understand and apply to a wide range of problems, but it also requires a large amount of training data and can be computationally complex during inference. Variations of KNN include weighted KNN, which gives more weight to closer neighbors, and KNN with kernel functions, which applies a kernel function to the distance metric. Implementing KNN involves preparing and preprocessing the data, calculating distances between data points, and making predictions using the algorithm. Examples of real-world applications of KNN include image classification, sentiment analysis, and personalized recommendations. By understanding the various aspects of KNN and its applications, one can effectively use the algorithm in practice.

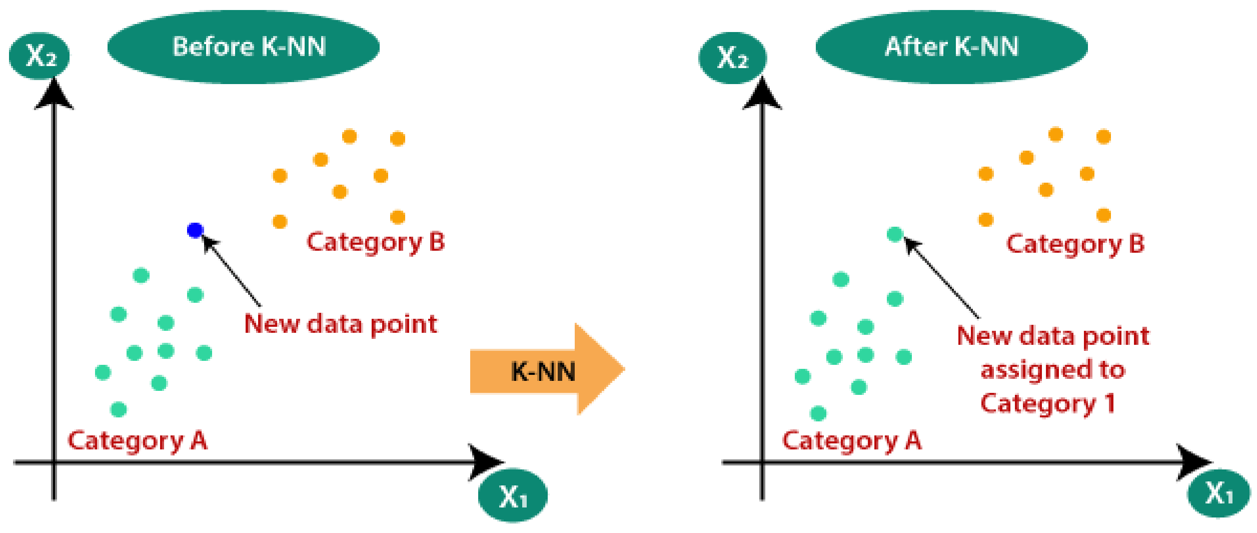


Fig 3.11 K-Nearest neighbour

3.4.9 EVALUATING MODEL

Evaluating models helps to determine the accuracy and effectiveness of the model in making predictions. There are several methods for evaluating a model, and the choice of method depends on the type of problem and the data being used. This refers to the process of assessing its performance on a specific task using set of evaluation metrics After training , usethe testing data to evaluate the performance of the Machine learning model .This involves calculating metrics such as accuracy, Precision, recall.

A. Confusion Matrix

A confusion matrix is a performance evaluation tool used in machine learning and statistics to assess the accuracy of a classification model. It is particularly useful when dealing with supervised learning problems where the data has predefined labels or classes.

The confusion matrix is a square matrix that summarizes the predictions of a model by comparing them with the actual labels of the data. It is called a "confusion" matrix because it helps identify instances where the model is confused or misclassifies certain data points.

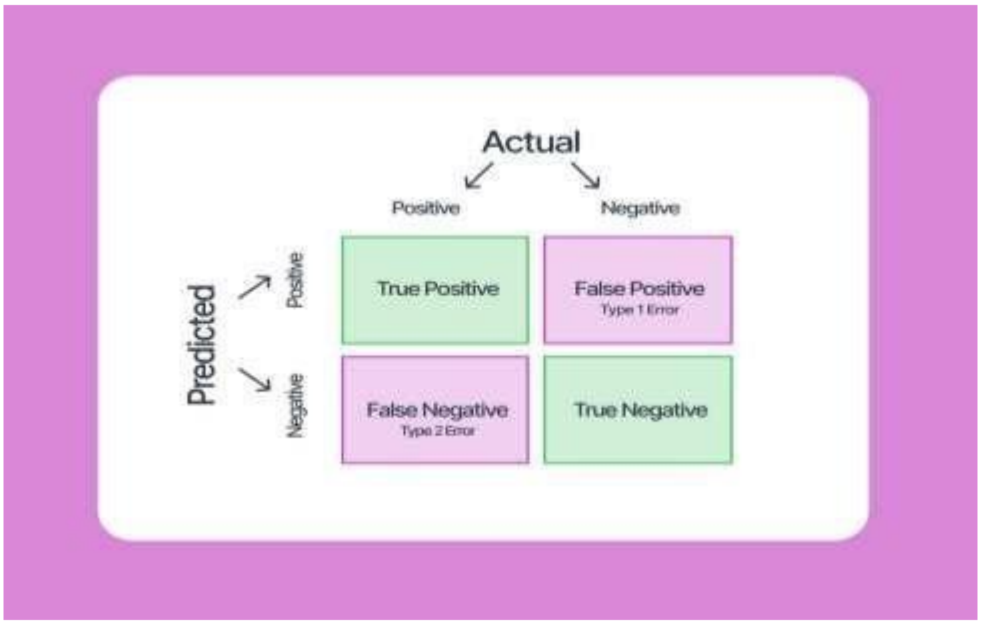


Fig 3.12 Confusion Matrix

Here's a brief explanation of the elements in a confusion matrix:

3.4.2 True Positives (TP): These are the cases where the model predicted a positive class, and the actual label was also positive.

3.4.3 True Negatives (TN): These are the cases where the model predicted a negative class, and the actual label was also negative.

3.4.4 False Positives (FP): These are the cases where the model predicted a positive class, but the actual label was negative (a type I error).

3.4.5 False Negatives (FN): These are the cases where the model predicted a negative class, but the actual label was positive (a type II error).

a) Accuracy:

The proportion of correctly classified instances over the total number of instances

Accuracy = (TP + TN) / (TP + TN + FP + FN)

TP -True Positive

TN- true negatives

FN-false Negatives

FP -False positives

b) Precision:

It is implied as the measure of correctly identified positive case (Tp) from all the predictive positive cases (Tp + Fp).

Precision = TP/ (TP + FP)

c) Recall:

It is implied as the measure of correctly identified positive case(Tp) from all the actual positive cases.

Recall = TP/(TP + FN)

3.4.10 CONNECTING TO WEB INTERFACE

Connecting a web application to Python and utilizing a pickled machine learning model can be accomplished using the Streamlit web framework. Streamlit is a lightweight and flexible Python web framework that provides tools and libraries for building web applications. To start, the pickled machine learning model can be loaded into the Streamlit application using the pickle library. Next.This route can take in the required input data for the machine learning model prediction and pass it through the loaded model. The prediction output can then be returned to the web application as a response. Streamlit also provides various methods to interact with HTML templates, allowing for easy integration of the model prediction results into the web application's user interface. By utilizing Streamlit and pickled machine learning models, web developers can easily incorporate machine learning-based predictions into their applications**.**

Streamlit:

Streamlit is an open-source Python library that allows you to create web applications for data science and machine learning projects. It simplifies the process of building interactive web interfaces by providing a straightforward API for displaying data, visualizations, and interactive widgets

3.5 CHAPTER SUMMARY

This chapter gives a complete step of building a model using machine learning algorithms. We have used five algorithms logistic regression, decision tree, random forest, support vector machine and K-Nearest Neighbor. After evaluating each model, we compared various metrices like precision, accuracy, and recall. Based on these metrices we have found that random forest has best accuracy and finally connected to the web interface for userinteraction using Streamlit.

Chapter 4

COMPONENTS REQUIRED

4.1 SYSTEM REQUIREMENTS

System Requirements Specification (SRS) (also known as a Software Requirements Specification) is a document or set of documentation that describes the features and behavior of a system or software application.

4.1.1 HARDWARE REQUIREMENTS:

System	:	Intel i3 2.1 GHZ
Memory	:	8 GB
Hard Disk	:	200 GB.

4.1 2 SOFTWARE REQUIREMENTS:

Operating System	:	Windows 10
Language	:	Python, Java, Html, CSS, Java Script.
Tool	:	Jupiter Notebook, Flask, Visual Studio.

4.1 3 FUNCTIONAL REQUIREMENTS:

- Device must be enabled or disabled by user.
- Device should be able to load the model correctly.
- Device should be able to extract the features of the form model file.
- Device should be able to recommend the appropriate output.
- Device should work without any several lagging.

4.1.4 NON-FUNCTIONAL REQUIREMENTS:

The non-functional requirements are divided into usability, reliability, performance, Supportability, and safety.

a. Usability

The system must be easy to learn for both users of the device and helpers who are theusers of the GUI interface.

b. Reliability

The reliability of the device essentially depends on the software tools (numpy, sklearnetc.) and hardware tools (computer etc.) used for the system development

c. Performance

- Csv data loading through program and live streaming makes performance measures crucial.
- For desired performance, image capturing, transferred data size, speed of connection, response time, processing speed must be considered.
- System should work real-time which means there should be an acceptable timedelay such as max 4-5 seconds between request and response.
- Data processing should be optimized so it should not take time more than 2 seconds. Program should not process every frame and should determine whether process or not the frame.

4.1.5 Libraries

a. Pandas - Pandas is a Python library used for working with data sets. It has functions for analyzing, cleaning, exploring, and manipulating data. Pandas allows us to analyzebig data and make conclusions based on statistical theories. Pandas can clean messy data sets, and make them readable and relevant.

b. NumPy - NumPy is a Python library used for working with arrays. It also has functions for working in domain of linear algebra, Fourier transform, and matrices. In Python we have lists that serve the purpose of arrays, but they are slow to process. NumPy aims to provide an array object that is up to 50x faster than traditional Python lists. The array object in NumPy is called ND array, it provides a lot of supporting functions that make working with ND array very easy. Arrays are very frequently usedin data science, where speed and resources are very important.

c. Sklearn - Scikit-learn is probably the most useful library for machine learning in Python. The sklearn library contains a lot of efficient tools for machine learning and statistical modelling including classification, regression, clustering and dimensionality reduction. Scikit- learn is a free machine learning library for Python. It features various algorithms like support vector machine, random forests, and k- neighbours, and it also supports Python numerical and scientific libraries like NumPy and SciPy

d. Matplotlib - Matplotlib is a popular open-source data visualization library for Python. It provides a wide range of customizable charts, graphs, and plots for visualizing data in various formats, including line plots, scatter plots, histograms, bar plots, pie charts, and many others.

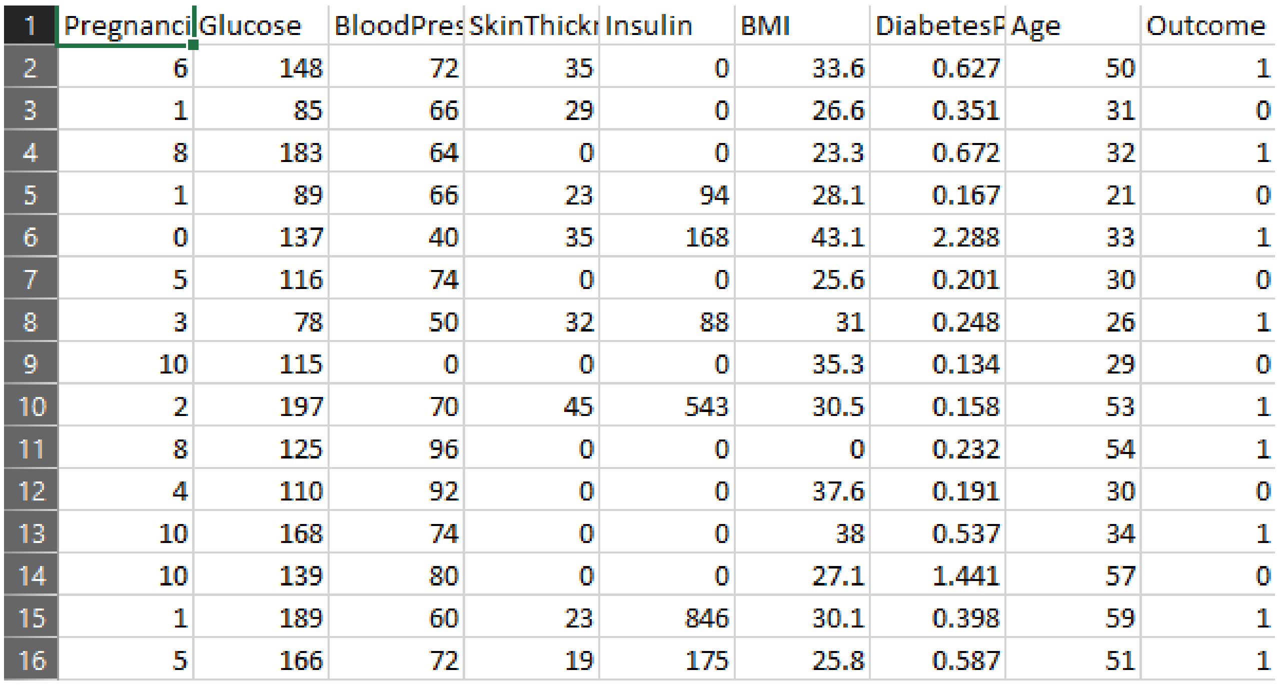
e. Seaborn - is a popular data visualization library for Python that is built ontop of Matplotlib. It provides a high-level interface for creating attractive and informative statistical graphics and visualizations with minimal coding.

4.2 CHAPTER SUMMARY

In implementation we have discussed system requirements, system design and steps included in creating our model. This chapter also provides implementation programming details with resulting output graphs and accuracy results. Machine learning techniques for model building start with collecting dataset followed by programming with importing various required libraries. Then we perform data pre- processing, correlating various features of attributes. Next, we apply ML algorithms in order to find accuracy using confusion matrix once we find the accuracy, we use flask to connect our backend ML model to web for user interface. This chapter includes programming codes of implementation.

Chapter 5 RESULT AND DICUSSION

5.1.1 Data Collection



1	Pregnanci	Glucose	BloodPres	SkinThickı	Insulin	BMI	DiabetesP	Age	Outcome
2	6	148	72	35	0	33.6	0.627	50	1
3	1	85	66	29	0	26.6	0.351	31	0
4	8	183	64	0	0	23.3	0.672	32	1
5	1	89	66	23	94	28.1	0.167	21	0
6	0	137	40	35	168	43.1	2.288	33	1
7	5	116	74	0	0	25.6	0.201	30	0
8	3	78	50	32	88	31	0.248	26	1
9	10	115	0	0	0	35.3	0.134	29	0
10	2	197	70	45	543	30.5	0.158	53	1
11	8	125	96	0	0	0	0.232	54	1
12	4	110	92	0	0	37.6	0.191	30	0
13	10	168	74	0	0	38	0.537	34	1
14	10	139	80	0	0	27.1	1.441	57	0
15	1	189	60	23	846	30.1	0.398	59	1
16	5	166	72	19	175	25.8	0.587	51	1

Fig. 5.1 Dataset

The Dataset is collected from Kaggle website. This is used to train the Machine Learning Model. The total number of samples present for house price prediction is 521 and we have 16 parameters. All these samples are one-dimensional data which is in the excel sheet of csv format. According to the data set, there is one target variable(dependent) and the others being independent(predictable) variables. The class column in the dataset is the dependent variable and the rest of the columns are independent variables. The 16 parameters are Age, Gender, Polyuria, Polydipsia, Sudden weakness, Polyphagia, Genital thrush, Visual blurring, Itching, Irritability, Delayed healing, Partial paresis, Muscle stiffness, Alopecia, Obesity, Class.

5.1.2 Import Libraries

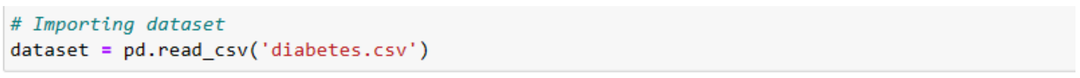
```python
#Load the libraries

import pandas as pd
import numpy as np
from pandas_profiling import ProfileReport
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import classification_report
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.svm import SVC
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import classification_report
from sklearn.neighbors import KNeighborsClassifier
from joblib import dump, load
from sklearn.metrics import accuracy_score
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, classification_report
```

Fig. 5.2 Importing the libraries.

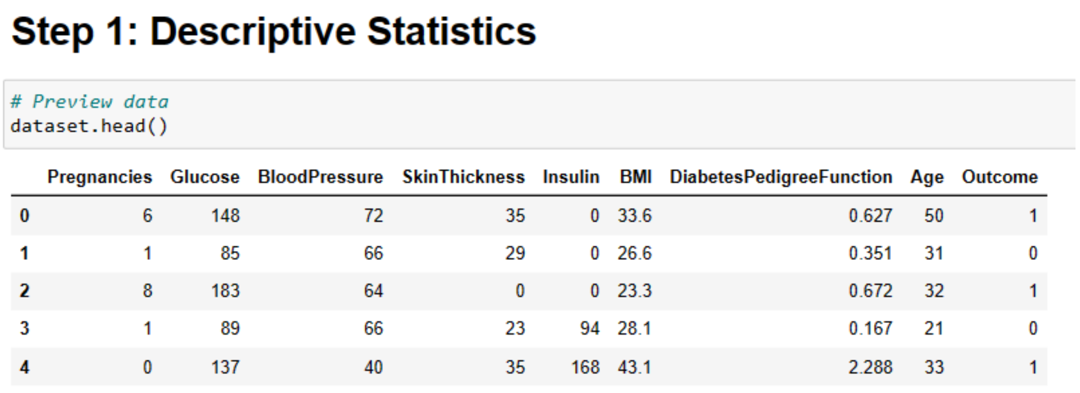
5.1.3 Load Dataset

read_csv() - is an important panda's function to read CSV files and do operations on them.

head () - To obtain the first n rows, use the head () function. This function retrieves the object's first n rows based on location. It is handy for rapidly determining whether your object contains the correct type of data.



```
# Importing dataset
dataset = pd.read_csv('diabetes.csv')
```



Step 1: Descriptive Statistics

```
# Preview data
dataset.head()
```

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

Fig. 5.3 Loading dataset

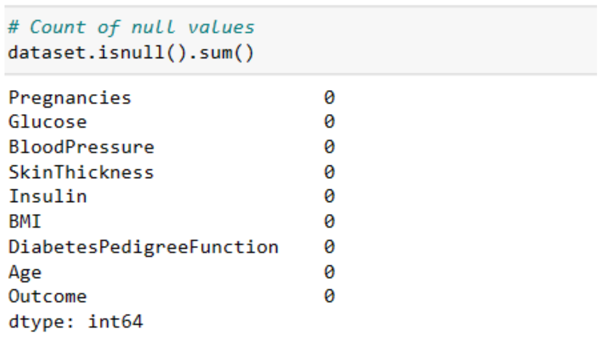
5.1.4 Preparing the Dataset

a. Checking for missing/null values.

b. Examining the information in the columns.

c. The fundamental statistics of the numeric column.

d. Imputing/encoding the data (In this case, we'll convert Yes/No at the end because 0and 1 in all the columns can be confusing).



```
# Count of null values
dataset.isnull().sum()
```

```
Pregnancies                 0
Glucose                     0
BloodPressure               0
SkinThickness               0
Insulin                     0
BMI                         0
DiabetesPedigreeFunction    0
Age                         0
Outcome                     0
dtype: int64
```

Fig. 5.4 Checking for null values

5.1.4 Data Visualization

Histogram: A histogram is a useful visualization tool to analyze the distribution of a continuous variable. While diabetes prediction typically involves classification tasks rather than continuous variables, you can still create a histogram to examine the distribution of certain features in your dataset that may be relevant to diabetes prediction

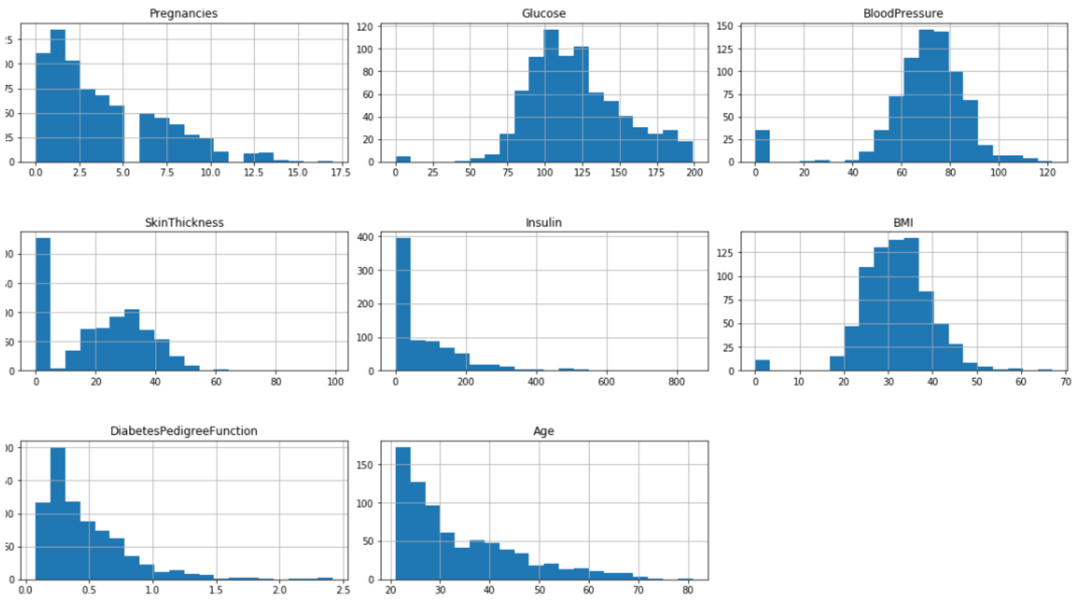
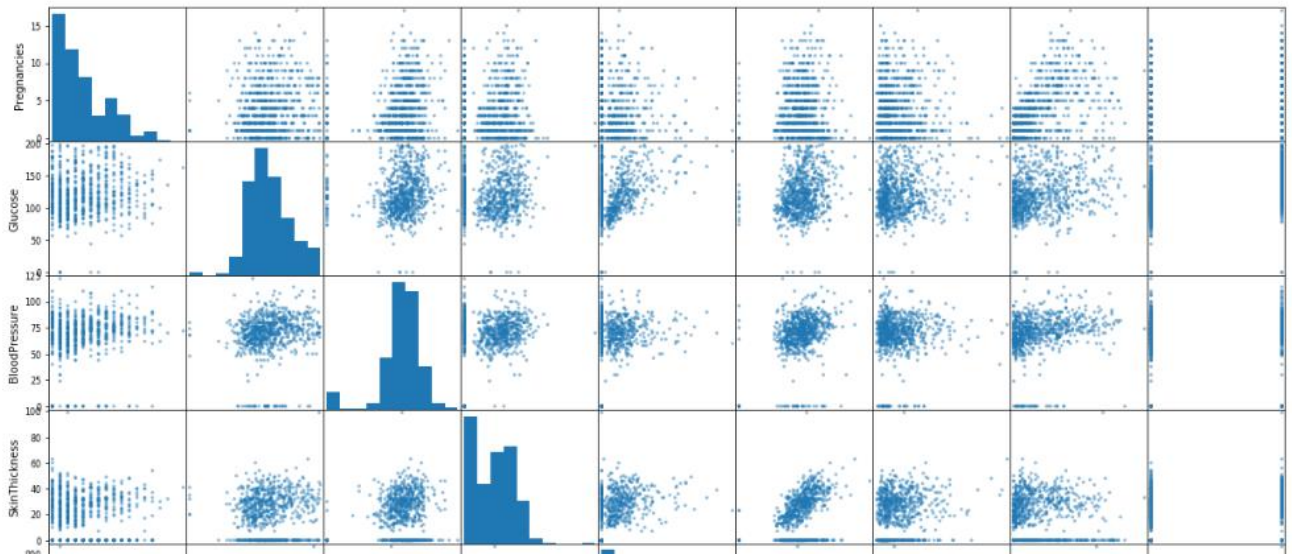


Fig. 5.7 Histogram Distribution

Scatter Plot Matrix:

A scatter plot matrix, also known as a pair plot, is a useful visualization tool for exploring the relationships between multiple variables in a dataset. While diabetes prediction typically involves classification tasks, you can still create a scatter plot matrix to examine the relationships between various features that may be relevant to diabetes prediction. Here's a step-by-step guide on how to create a scatter plot matrix for diabetes prediction

Fig 5.8 Scatter plot



i. Data Preprocessing:

```
dataset_new = dataset
```

```
# Replacing zero values with NaN
dataset_new[["Glucose", "BloodPressure", "SkinThickness", "Insulin", "BMI"]] = dataset_new[["Glucose", "BloodPressure", "SkinThic
```

```
# Count of NaN
dataset_new.isnull().sum()
```

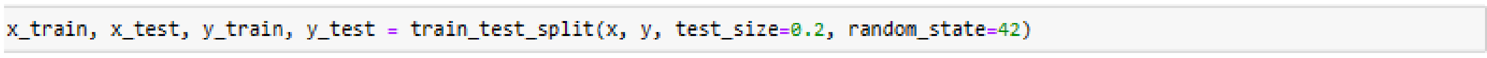
```
Pregnancies                   0
Glucose                       5
BloodPressure                35
SkinThickness               227
Insulin                     374
BMI                          11
DiabetesPedigreeFunction      0
Age                           0
Outcome                       0
dtype: int64
```

```
# Replacing NaN with mean values
dataset_new["Glucose"].fillna(dataset_new["Glucose"].mean(), inplace = True)
dataset_new["BloodPressure"].fillna(dataset_new["BloodPressure"].mean(), inplace = True)
dataset_new["SkinThickness"].fillna(dataset_new["SkinThickness"].mean(), inplace = True)
dataset_new["Insulin"].fillna(dataset_new["Insulin"].mean(), inplace = True)
dataset_new["BMI"].fillna(dataset_new["BMI"].mean(), inplace = True)
```

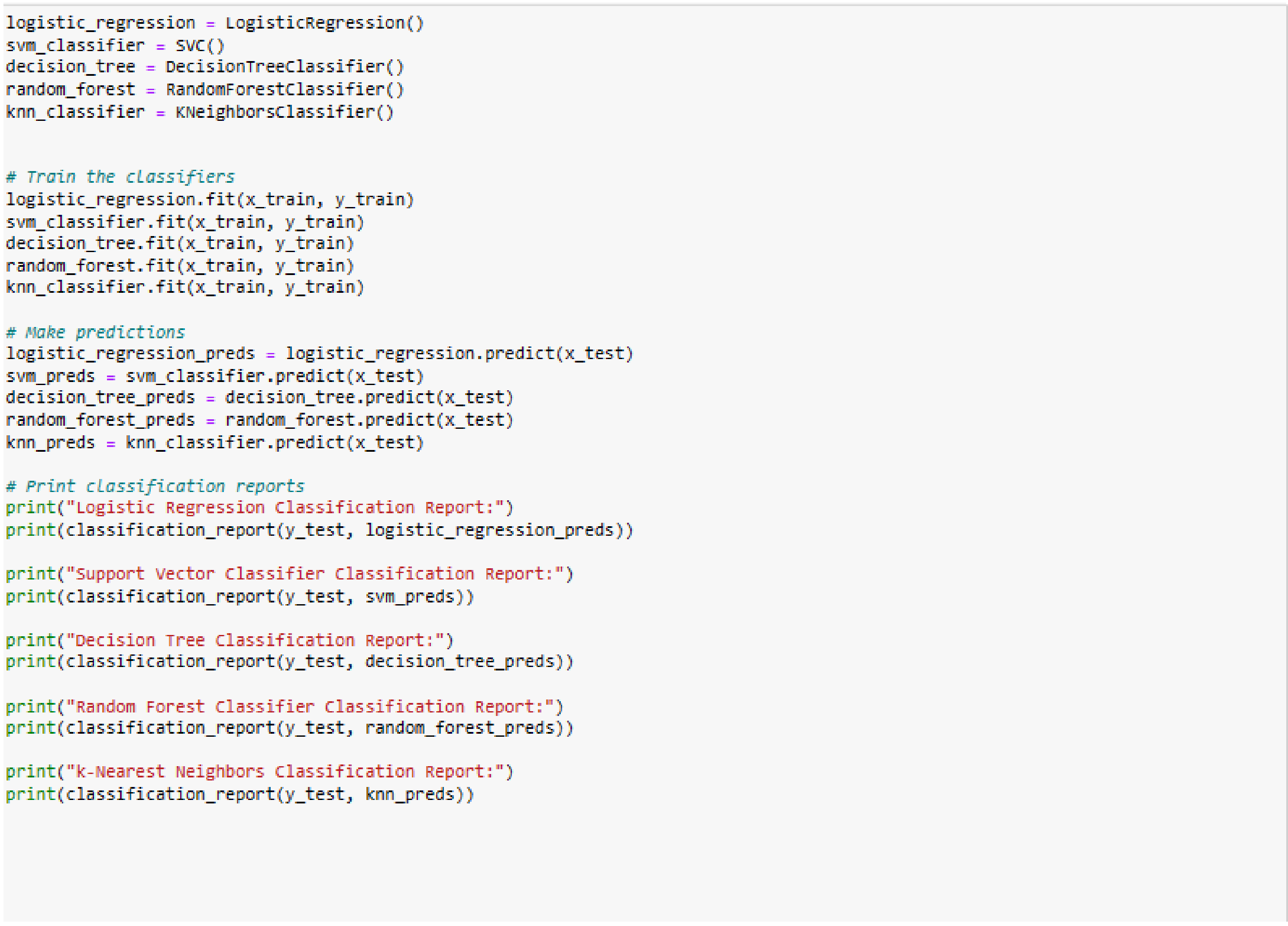
```
# Statistical summary
dataset_new.describe().T
```

	count	mean	std	min	25%	50%	75%	max
Pregnancies	768.0	3.845052	3.369578	0.000	1.00000	3.000000	6.000000	17.00
Glucose	768.0	121.686763	30.435949	44.000	99.75000	117.000000	140.250000	199.00
BloodPressure	768.0	72.405184	12.096346	24.000	64.00000	72.202592	80.000000	122.00
SkinThickness	768.0	29.153420	8.790942	7.000	25.00000	29.153420	32.000000	99.00
Insulin	768.0	155.548223	85.021108	14.000	121.50000	155.548223	155.548223	846.00
BMI	768.0	32.457464	6.875151	18.200	27.50000	32.400000	36.600000	67.10
DiabetesPedigreeFunction	768.0	0.471876	0.331329	0.078	0.24375	0.372500	0.626250	2.42

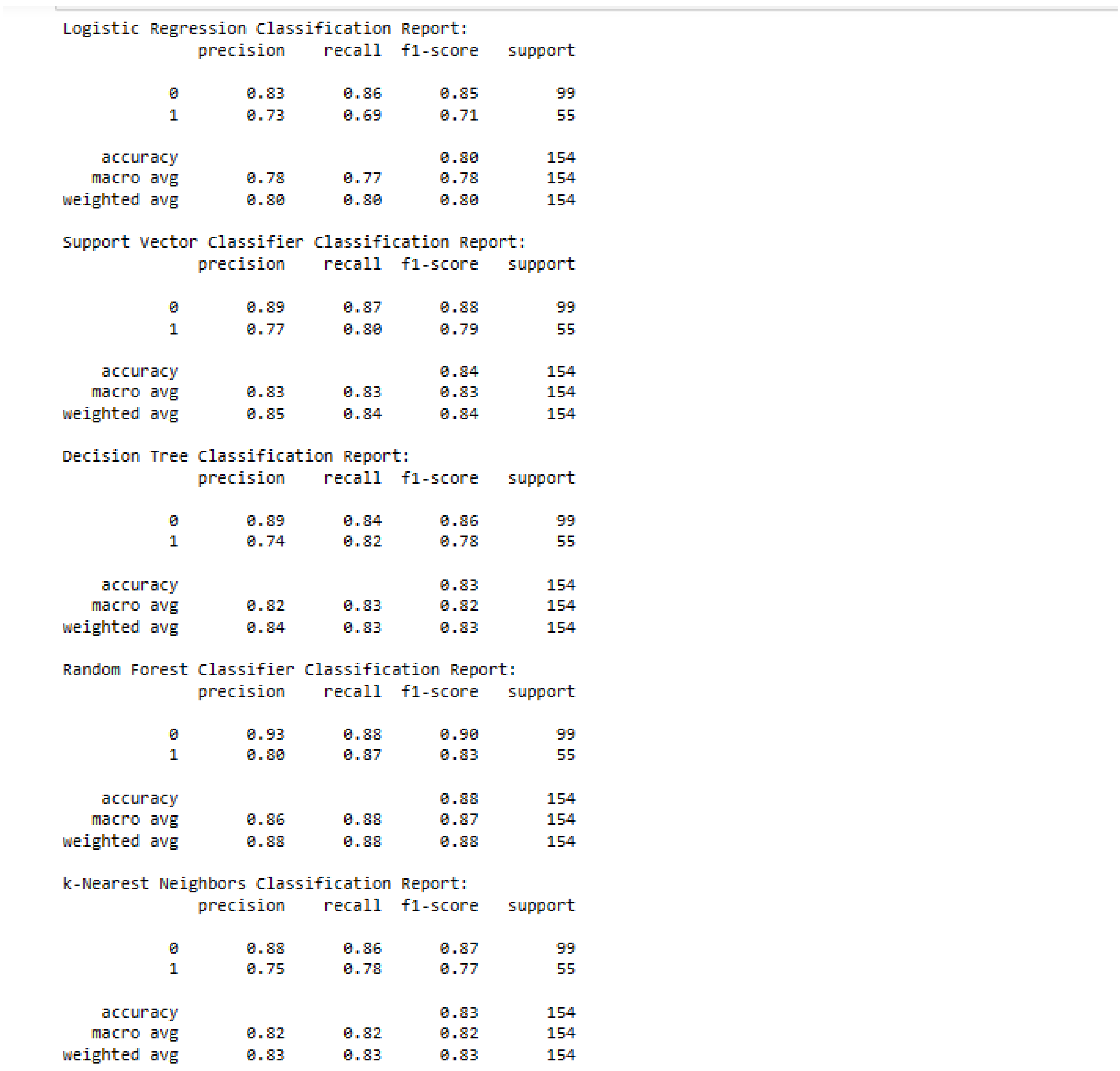
ii. Data Modelling



```
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2, random_state=42)
```

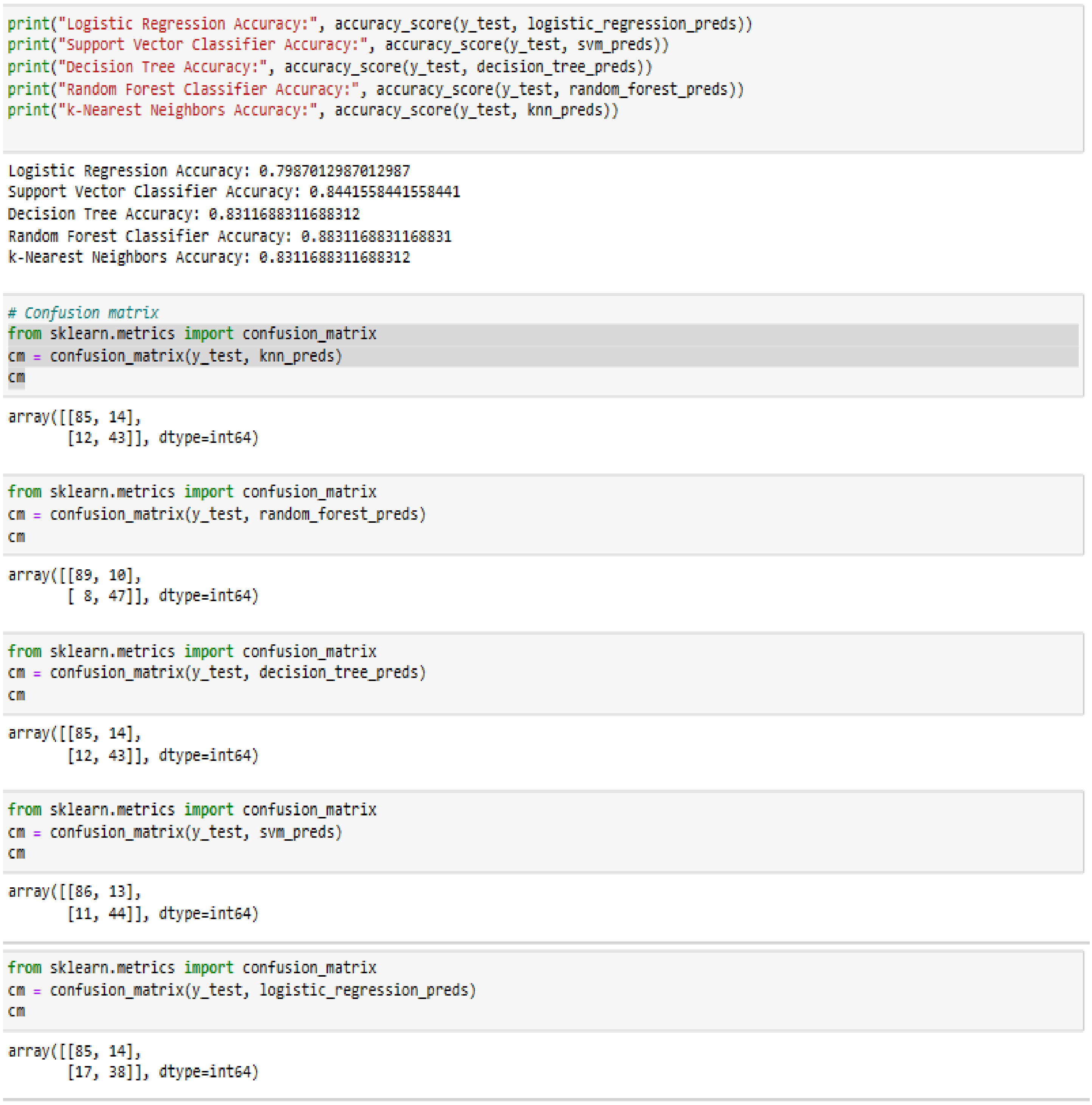


```
logistic_regression = LogisticRegression()
svm_classifier = SVC()
decision_tree = DecisionTreeClassifier()
random_forest = RandomForestClassifier()
knn_classifier = KNeighborsClassifier()


# Train the classifiers
logistic_regression.fit(x_train, y_train)
svm_classifier.fit(x_train, y_train)
decision_tree.fit(x_train, y_train)
random_forest.fit(x_train, y_train)
knn_classifier.fit(x_train, y_train)

# Make predictions
logistic_regression_preds = logistic_regression.predict(x_test)
svm_preds = svm_classifier.predict(x_test)
decision_tree_preds = decision_tree.predict(x_test)
random_forest_preds = random_forest.predict(x_test)
knn_preds = knn_classifier.predict(x_test)

# Print classification reports
print("Logistic Regression Classification Report:")
print(classification_report(y_test, logistic_regression_preds))

print("Support Vector Classifier Classification Report:")
print(classification_report(y_test, svm_preds))

print("Decision Tree Classification Report:")
print(classification_report(y_test, decision_tree_preds))

print("Random Forest Classifier Classification Report:")
print(classification_report(y_test, random_forest_preds))

print("k-Nearest Neighbors Classification Report:")
print(classification_report(y_test, knn_preds))
```



```
Logistic Regression Classification Report:
              precision    recall  f1-score   support

           0       0.83      0.86      0.85        99
           1       0.73      0.69      0.71        55

    accuracy                           0.80       154
   macro avg       0.78      0.77      0.78       154
weighted avg       0.80      0.80      0.80       154

Support Vector Classifier Classification Report:
              precision    recall  f1-score   support

           0       0.89      0.87      0.88        99
           1       0.77      0.80      0.79        55

    accuracy                           0.84       154
   macro avg       0.83      0.83      0.83       154
weighted avg       0.85      0.84      0.84       154

Decision Tree Classification Report:
              precision    recall  f1-score   support

           0       0.89      0.84      0.86        99
           1       0.74      0.82      0.78        55

    accuracy                           0.83       154
   macro avg       0.82      0.83      0.82       154
weighted avg       0.84      0.83      0.83       154

Random Forest Classifier Classification Report:
              precision    recall  f1-score   support

           0       0.93      0.88      0.90        99
           1       0.80      0.87      0.83        55

    accuracy                           0.88       154
   macro avg       0.86      0.88      0.87       154
weighted avg       0.88      0.88      0.88       154

k-Nearest Neighbors Classification Report:
              precision    recall  f1-score   support

           0       0.88      0.86      0.87        99
           1       0.75      0.78      0.77        55

    accuracy                           0.83       154
   macro avg       0.82      0.82      0.82       154
weighted avg       0.83      0.83      0.83       154
```

iii. Model Evaluation:



```python
print("Logistic Regression Accuracy:", accuracy_score(y_test, logistic_regression_preds))
print("Support Vector Classifier Accuracy:", accuracy_score(y_test, svm_preds))
print("Decision Tree Accuracy:", accuracy_score(y_test, decision_tree_preds))
print("Random Forest Classifier Accuracy:", accuracy_score(y_test, random_forest_preds))
print("k-Nearest Neighbors Accuracy:", accuracy_score(y_test, knn_preds))
```

```
Logistic Regression Accuracy: 0.7987012987012987
Support Vector Classifier Accuracy: 0.8441558441558441
Decision Tree Accuracy: 0.8311688311688312
Random Forest Classifier Accuracy: 0.8831168831168831
k-Nearest Neighbors Accuracy: 0.8311688311688312
```

```python
# Confusion matrix
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test, knn_preds)
cm
```

```
array([[85, 14],
       [12, 43]], dtype=int64)
```

```python
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test, random_forest_preds)
cm
```

```
array([[89, 10],
       [ 8, 47]], dtype=int64)
```

```python
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test, decision_tree_preds)
cm
```

```
array([[85, 14],
       [12, 43]], dtype=int64)
```

```python
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test, svm_preds)
cm
```

```
array([[86, 13],
       [11, 44]], dtype=int64)
```

```python
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test, logistic_regression_preds)
cm
```

```
array([[85, 14],
       [17, 38]], dtype=int64)
```

Algorithm	**Accuracy**	**Precision**	**Recall**
Random Forest	86.36	0.91	0.87
KNN	83.11	0.88	0.86
Decision Tree	81.16	0.87	0.83
Support Vector Classifier	84.41	0.89	0.87
Logistic Regression	79.87	0.83	0.86

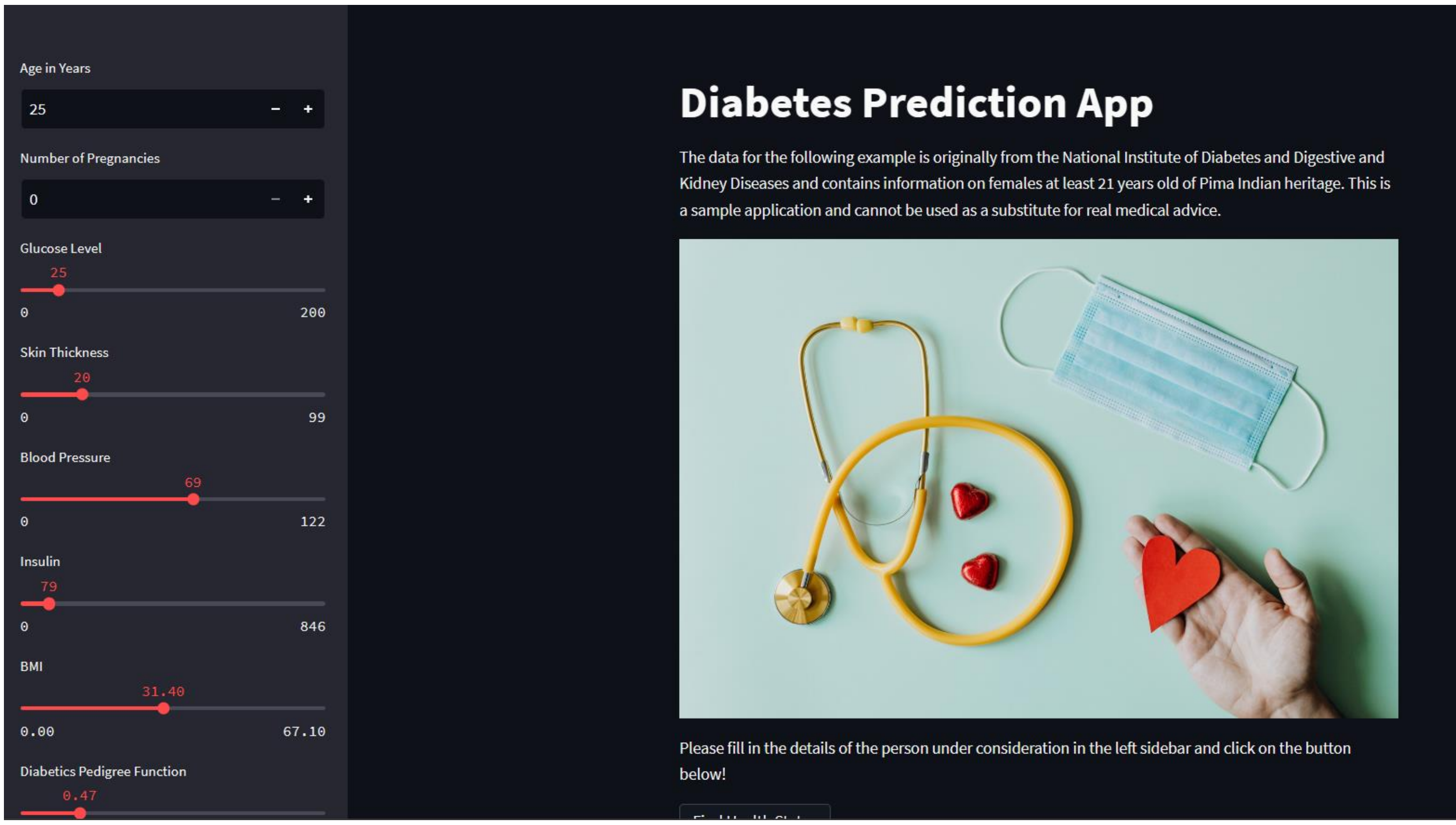
Table 5.1 Results

In the above table we can observe that KNN and random forest provide the highest accuracy compared to all the other algorithms. So, we consider the next two parameters in order to find the best algorithm. Random Forest highest precision and recall so we use Random Forest to produce best results.

5.4.6 Streamlit CODE:

```
import streamlit as st
import joblib
import pandas as pd
from PIL import Image

@st.cache(allow_output_mutation=True)
def load(scaler_path, model_path):
    sc = joblib.load(scaler_path)
    model = joblib.load(model_path)
    return sc , model

def inference(row, scaler, model, feat_cols):
    df = pd.DataFrame([row], columns = feat_cols)
    X = scaler.transform(df)
    features = pd.DataFrame(X, columns = feat_cols)
    if (model.predict(features)==0):
        return "This is a healthy person!"
    else: return "This person has high chances of having diabetics!"

st.title('Diabetes Prediction App')
st.write('The data for the following example is originally from the National Institute of Diabetes and Digestive and Kidney Diseases and contains information on females at least 21 years old of Pima Indian heritage. This is a sample application and cannot be used as a substitute for real medical advice.')
image = Image.open('data/diabetes_image.jpg')
st.image(image, use_column_width=True)
st.write('Please fill in the details of the person under consideration in the left sidebar and click on the button below!')

age =           st.sidebar.number_input("Age in Years", 1, 150, 25, 1)
pregnancies =   st.sidebar.number_input("Number of Pregnancies", 0, 20, 0, 1)
glucose =       st.sidebar.slider("Glucose Level", 0, 200, 25, 1)
skinthickness = st.sidebar.slider("Skin Thickness", 0, 99, 20, 1)
bloodpressure = st.sidebar.slider('Blood Pressure', 0, 122, 69, 1)
insulin =       st.sidebar.slider("Insulin", 0, 846, 79, 1)
bmi =           st.sidebar.slider("BMI", 0.0, 67.1, 31.4, 0.1)
```

```
dpf =           st.sidebar.slider("Diabetics Pedigree Function", 0.000, 2.420, 0.471, 0.001)

row = [pregnancies, glucose, bloodpressure, skinthickness, insulin, bmi, dpf, age]

if (st.button('Find Health Status')):
    feat_cols = ['Pregnancies', 'Glucose', 'BloodPressure', 'SkinThickness', 'Insulin', 'BMI', 'DiabetesPedigreeFunction', 'Age']

    sc, model = load('models/scaler.joblib', 'models/model.joblib')
    result = inference(row, sc, model, feat_cols)
    st.write(result)
```



Age in Years

25

Number of Pregnancies

0

Glucose Level

25

0 200

Skin Thickness

20

0 99

Blood Pressure

69

0 122

Insulin

79

0 846

BMI

31.40

0.00 67.10

Diabetics Pedigree Function

0.47

Diabetes Prediction App

The data for the following example is originally from the National Institute of Diabetes and Digestive and Kidney Diseases and contains information on females at least 21 years old of Pima Indian heritage. This is a sample application and cannot be used as a substitute for real medical advice.

Please fill in the details of the person under consideration in the left sidebar and click on the button below!

Fig 5.27 Diabetic Prediction

The diabetic detection page may 16 key symptoms checker that allows users to select symptoms they are having including age.Fig.5.27, The page features a user-friendly interface that displays the results of the machine learning algorithm in a clear and easy-to-understand way that is diabetes or no diabetes.

5.2 Chapter Summary

In implementation we have discussed system requirements, system design and steps included in creating our model. This chapter also provides implementation programming details with resulting output graphs and accuracy results. Machine learning techniques for model building start with collecting dataset followed by programming with importing various required libraries. Then we perform data pre- processing, correlating various features of attributes. Next, we apply ML algorithms in order to find accuracy using confusion matrix once we find the accuracy, we use flask to connect our backend ML model to web for user interface. This chapter includes programming codes of implementation.

Chapter 6

CONCLUSION & FUTURE SCOPE

6.1 Conclusion

The Early-stage Diabetic Prediction System is a comprehensive solution that utilizes machine learning techniques to predict the likelihood of an individual developing diabetes based on their symptoms. The system uses a dataset of symptoms associated with diabetes and applies various machine learning algorithms to identify patterns and relationships between the symptoms and the likelihood of developing diabetes. To provide a seamless user experience, a separate platform for diabetic diseases has been created, which includes a range of information such as medicines used, healthy tips, and information on various types of diabetes. This platform is accessible via a web interface, which allows users to accessinformation from anywhere, at any time. The Early-stage Diabetic Prediction System aims to improve the early detection of diabetes, which can lead to early intervention and treatment, thus reducing the risk of complications associated with diabetes. By providing individuals with access to relevant information and resources, we hope to empower them to take control of their health and make informed decisions about their lifestyle choices. In conclusion, our Diabetic Prediction

6.2 Future Scope

System and the accompanying platform for diabetic diseases is a comprehensive solution that utilizes machine learning techniques to predict the likelihood of developing diabetes and provides individuals with access to relevant information and resources. By empowering individuals to take control of their health, we aim to improve their overall well- being and quality of life. The use of machine learning in diabetic prediction can also help to reduce healthcare costs by enabling earlier detection and prevention of the disease. Byidentifying individuals at high risk of developing diabetes, healthcare providers can intervene early with lifestyle interventions and preventative measures that can help to delay or even prevent the onset of the disease. This can result in significant cost savings for healthcare systems, as well as improving the quality of life for patients. In addition to its potential impact on healthcare, the use of machine learning in diabetic prediction can also haveimportant societal implications. For example, it can help to reduce health disparities by identifying individuals at high risk of diabetes who may not have access to regular healthcare, and by providing personalized recommendations and interventions that are tailored to individual needs and circumstances.

REFRENCES

[1] Ahmed, "Prediction of diabetics empowered with fused Machine Learning", 2022 International Research Journal of Modernization in Engineering Technology and Science, Student, Department Of Computer Engineering, Pune Institute Of Computer Technology, Pune, India.

[2] Daliya, "An Optimized Multivariable Regression Model for Predictive Analysis of Diabetic Disease Progression", 2021 Department of Electronics and Communication Engineering, Amrita School of Engineering, Amrita Vishwa Vidyapeetham, Bengaluru, India.

[3] Jiajia Song, Chao Wang, and Wenzhuo Zhao, "Literature review on machine learning for diabetes prediction", 2021.

[4] Hruaping Zhou, Raushan Myrzashova and Ruiz Heng, "An enhanced deep neural network (DNN) model for predicting diabetes."

[5] MD. Kamrul Hasan and MD. Ashraful Alam, "Diabetes Prediction Using Ensembling of Different Machine Learning Classifiers", 2020.

[6] Yahyauoi, "Developing a decision support system for predicting diabetes using machine learning and deep learning techniques."

[7] R. Raj, "Intelligent Diabetes Detection System using Machine Learning Techniques", 2020.

[8] S. T. Mir, "Diabetes Prediction using Machine Learning: A Review and Future Directions", 2021.

[9] B. V. Gowtham, "A Comprehensive Study on Predicting Diabetes Using Machine Learning Techniques", 2021.

[10] S. S. Shetty, "A Comparative Analysis of Machine Learning Techniques for Predicting Diabetes", 2022.