A PROJECT REPORT

on

DIABETES PREDICTION USING MACHINE LEARNING

Submitted in partial fulfillment of the requirements for the award of degree of

BACHELOR OF TECHNOLOGY
in
COMPUTER SCIENCE ENGINEERING

by

19WH1A0586	**T. NAGAJYOTHI**
19WH1A05B2	**G. PRAVALIKA**
19WH1A0580	**T. SHARANYA**

Under the esteemed guidance of
Dr. S. Ashok
Assistant Professor

Department of Computer Science Engineering

BVRIT HYDERABAD
College of Engineering for Women
(Approved by AICTE, New Delhi and Affiliated to JNTUH, Hyderabad)
Accredited by NBA and NAAC with A Grade
Bachupally, Hyderabad – 500090

June - 2023

DECLARATION

We hereby declare that the work presented in this project entitled **"Diabetes Prediction using Machine Learning"** submitted towards completion of Project work in IV Year of B.Tech of CSE at **BVRIT HYDERABAD College of Engineering for Women,** Hyderabad is an authentic record of our original work carried out under the guidance of **Dr. S. Ashok, Assistant Professor, Department of CSE.**

T. Nagajyothi
(19WH1A0586)

G. Pravalika
(19WH1A05B2)

T. Sharanya
(19WH1A0580)

BVRIT HYDERABAD
College of Engineering for Women
(Approved by AICTE, New Delhi and Affiliated to JNTUH, Hyderabad)
Accredited by NBA and NAAC with A Grade
Bachupally, Hyderabad – 500090

Department of Computer Science Engineering

CERTIFICATE

This is to certify that the major project entitled "**Diabetes Prediction using Machine Learning**" is a bonafide work carried out by **Ms. T. Nagajyothi (19WH1A0586), Ms. G. Pravalika (19WH1A05B2), Ms. T. Sharanya (19WH1A0580)** in partial fulfillment for the award of B.Tech degree in **Computer Science & Engineering** , **BVRIT HYDERABAD College of Engineering for Women, Bachupally, Hyderabad**, affiliated to Jawaharlal Nehru Technological University Hyderabad, Hyderabad under my guidance and supervision. The results embodied in the project work have not been submitted to any other University or Institute for the award of any degree or diploma.

Internal Guide
Dr. S. Ashok
Assistant Professor, CSE

Head of the Department
Dr. E. Venkateswara Reddy
Professor, CSE

External Examiner

ACKNOWLEDGEMENT

We would like to express our sincere thanks to **Dr.K.V.N.Sunitha**, **Principal, BVRIT HYDERABAD College of Engineering for Women,** for her support by providing the working facilities in the college.

Our sincere thanks and gratitude to **Dr. E. Venkateswara Reddy, Head, Department of CSE, BVRIT HYDERABAD College of Engineering for Women,** for all timely support and valuable suggestions during the period of our project.

We are extremely thankful to our Internal Guide, **Dr. S. Ashok, Assistant Professor, CSE, BVRIT HYDERABAD College of Engineering for Women,** for his constant guidance and encouragement throughout the project.

Finally, we would like to thank our Major Project Coordinator, all Faculty and Staff of CSE department who helped us directly or indirectly. Last but not least, we wish to acknowledge our **Parents** and **Friends** for giving moral strength and constant encouragement.

T. Nagajyothi (19WH1A0586)
G. Pravalika (19WH1A05B2)
T. Sharanya (19WH1A0580)

LIST OF CONTENTS

ABSTRACT

The diabetes is one of the lethal diseases in the world. The aim of the project is to develop a system which might predict the diabetic risk level of a patient with a better accuracy. Diabetes is seen in all age groups these days and they are attributed to lifestyle, genetic, stress and age factor. Whatever maybe the reasons for the diabetes, the outcome could be severe if left unnoticed. Currently various methods are being used to predict diabetes and diabetes inflicted diseases. Diabetes can be controlled if it is predicted earlier.

Many complications occur if diabetes remains untreated and unidentified. The tedious identifying process results in visiting 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.

Model development is based on categorization methods as KNN, SVM, Random forest and XGBoost algorithms.

LIST OF FIGURES

1. INTRODUCTION

All around there are numerous ceaseless infections that are boundless in evolved and developing nations. One of such sickness is diabetes. Diabetes is a metabolic issue that causes blood sugar by creating a significant measure of insulin in the human body or by producing a little measure of insulin. Diabetes is perhaps the deadliest sickness on the planet. It is not just a malady yet, also a maker of different sorts of sicknesses like a coronary failure, visual deficiency, kidney ailments and nerve harm, and so on.

Causes of Diabetes

Genetic factors are the main cause of diabetes. It is caused by at least two mutant genes in the chromosome 6, the chromosome that affects the response of the body to various antigens. Viral infection may also influence the occurrence of type 1 and type 2 diabetes. Studies have shown that infection with viruses such as rubella, Coxsackievirus, mumps, bepatitis B virus, and cytomegalovirus increase the risk of developing diabetes.

Types of Diabetes

Type 1

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 cause of type 1 diabetes and there are currently no known methods of prevention.

Type 2

Type 2 diabetes means that the cells produce a low quantity of insulin or the body cannot use the insulin correctly. This is the most common type of diabetes, thus affecting 90% of persons diagnosed with diabetes. It is caused by both genetic and the manner of living.

To achieve this goal this project work will do early prediction of Diabetes in human body or a patient for a higher accuracy through applying, various machine learning techniques. Machine learning techniques provide better result

for prediction by constructing models from datasets collected from patients. In this work we will use Machine learning classification and ensemble techniques on a dataset to predict diabetes which are KNN, SVM, Random forest and XGBoost algorithms are used.

1.1 Problem statement:

Diabetes is a most common disease caused by a group of metabolic disorders. It is also known as Diabetic mellitus. It affects the organs of the human body. It can be controlled by predicting this disease earlier. If diabetes patient is untreated for a long time, it may lead to increase blood sugar. Machine Learning algorithms and statistics are used to predict the disease with the help of current and past data. Machine learning techniques help the doctors to predict early stage for diabetes.

1.2 Motivation:

The main motivation of doing this research is to present a diabetes disease prediction model for the prediction of occurrence of diabetes disease. Further, this research work is aimed towards identifying the best algorithm for identifying the possibility of diabetes disease in a patient. Various attributes are taken into consideration and using machine learning algorithms, prediction is done. This will provide researchers and medical practitioners to establish a better health environment.

1.3 Objectives:

This project predicts people with diabetes disease prediction by extracting the patient medical history that leads to a fatal diabetes disease from a dataset that includes patient's medical history such as Pregnancies, Glucose, Blood Pressure etc. Our main objective to design an interactive application, in which user can give input to generate the output whether the person is affected with diabetes or not.

1.4 Methodology

To predict diabetes disease, large collection of patient's medical history is required. Various methodologies used are KNN, SVM, Random forest and XGBoost . In System Architecture, the description of these algorithms are explained in detailed manner.

1.4.1 Dataset

The dataset collected is available on Kaggle. It consists of several medical analyst variables and one target variable. The dataset consists of several independent variables and one dependent variable i.e., the outcome. This dataset contains 8 medical attributes of 768 patients that helps us detecting if the patient is at risk of getting a diabetes disease or not and it helps us classify patients that are at risk of having a diabetes disease and that who are not at risk.
This dataset gives us the much-needed information i.e. the medical attributes such as Pregnancies, Blood Pressure, Age , Skin Thickness etc. of the patient that helps us in detecting the patient that is diagnosed with diabetes disease or not.

S.No	Attribute Names	Description
1	Pregnancies	Number of times pregnant
2	Glucose	Plasma glucose concentration
3	Blood Pressure	Diastolic blood pressure

4	Skin Thickness	Triceps skin fold thickness (mm)
5	Insulin	2-h serum insulin
6	BMI	Body mass index
7	Diabetes pedigree function	Diabetes pedigree function
8	Age	Age of patient
9	Outcome	Class variable (0 or 1)

Table 1: Dataset Description

1.5 Proposed system:

The proposed system predicts diabetes disease. Initially the data is collected. Then the required data is preprocessed into the required format. The data is then divided into two parts training and testing data. The algorithms are applied and the model is trained using the training data. Using various machine learning algorithms (KNN, SVM, Random forest, XGBoost) the accuracy of the system is obtained by testing the system using the testing data. To improve the performance of the model, various evaluation metrics like confusion matrix, precision, recall and f1-score are used. The proposed framework will consider the high accuracy algorithm for predicting diabetes disease.

2. LITERATURE REVIEW

2.1 Title : Diabetes Disease Prediction Using Machine Learning Algorithms

Authors : Arwatki Chen Lyngdoh, Nurul Amin Choudhury, Soumen Moulik

Summary : The paper presented the Diabetes disease which causes an increase in blood glucose levels as result of the absence of the insulin. It is a hormonal disorder. Diabetes include risk factors such as age, BMI (Body Mass Index), glucose levels, Blood pressure (BP) play a crucial role in diabetes disease. They have used publicly available dataset named as Puma Indians Diabetes Database. Metrics such as Precision, Recall, F1-score and Accuracy are used in the analysis of diabetes prediction.

2.2 Title : Diabetes Prediction using Machine Learning Algorithms with Feature Selection and Dimensionality Reduction

Authors : Sivaranjani S, Ananya S, Aravinth J, Karthika R

Summary : There are two types of diabetes patients type-1 diabetes do not produce insulin and type – 2 diabetes do not respond properly to insulin. Major symptoms of these diabetes are urinate, thirsty frequently. There are more than 75% patients are of type – 2. There are several machine learning algorithms like decision tree, KNN to predict the diabetes. The data pre-processing is done such that the missing values are replaced by non zero means values of each parameter. Feature Selection is done to extract the most influencing parameters and classification is done. The model selects four most contributing features and the accuracy after step forward feature selection in different classifiers. The dimensionality reduction increases the test set accuracy of the classifiers. Dimensionality is not of high significance as the dataset is not very large.

2.3 Title : Diabetes Disease Prediction using Machine Learning on Big Data of Healthcare

Authors : Ayman Mir, Sudhir N.Dhage

Summary : The paper analyzed the trends in treatment of patients for diagnosis of a particular disease will help in making informed and efficient decisions to improve the overall quality of healthcare. Machine Learning is a very promising approach which helps in early diagnosis of disease and might help the practitioners in decision making for diagnosis. The classifiers build have been compared based on training time, testing time and accuracy value. The accuracy measures included TP – rate, FP – rate, precision, recall, F – measure.

2.4 Title : Early Prediction of Diabetes Mellitus Using Machine Learning

Authors : Gaurav Tripathi, Rakesh Kumar

Summary : Diabetes is a noxious disease that causes the organ pancreas unable to produce enough insulin which leads to increase in blood glucose levels in the body. Performance metrics are evaluated and Random forests gives the maximum accuracy of 87.66%. Classification techniques are widely used in pattern recognition or predictive analysis for classifying the data into different classes. The metrics are accuracy, sensitivity, precision, specificity, and F – score. These metrics depend on classification labels such as true positive, true negative, false positive and false negative. Missing values are replaced by class mean and for class imbalance problem over-sampling method is used.

2.5 Title : Diabetes Prediction using different Machine Learning Approaches

Authors : Priyanka Sonar, Professor K. JayaMalini

Summary : The paper presented with the unstructured and semi structured data like text, images, trees and the SVM algorithm works well. It gives good prediction and easy to implement. Difficult with dealing with Big data with complex model requires huge processing time. The training data set in Machine learning is used to train the model for carrying out abundant actions. Detailed features are fetched from the training set to train the model. These structures are therefore combined into the prototype. In sentiment analysis, single words or sequences of consecutive words are taken from the tweets. Therefore, if the training set is labeled correctly, then the model will be able to acquire something from the features. So for testing the model such type of data is used to check whether it is responding correctly or not.

3. REQUIREMENTS

3.1 Software Requirements

- Windows 10
- Python 3.8
- Google Colab
- Anaconda

3.2 Hardware Requirements

- Processor : Intel core I5
- Memory : RAM 8GB

3.3 Technologies Description

- **Google Colaboratory**

  Colab allows anybody to write and execute arbitrary python code through the browser, and is especially well suited to machine learning, data analysis and education.

- **Anaconda Prompt**

  Anaconda Prompt is a command line interface with Anaconda Distribution. prompt is a library for prompting input on the command line for Python 3.3+. It is pure Python code with no dependencies.

- **Libraries**

1. Numpy

NumPy is a powerful Python library widely used for scientific computing and data analysis. It provides a multidimensional

array object that allows efficient storage and manipulation of large numerical datasets.

2. Pandas

Pandas is a widely used Python library that simplifies data manipulation and analysis. It provides easy-to-use data structures, such as Data Frames, which are two-dimensional tabular structures capable of holding diverse data types.

3. Matplotlib

Matplotlib is a widely used Python library for creating high-quality visualizations and plots. It provides a comprehensive set of functions and tools for generating a wide range of plots, charts, histograms, and other visual representations of data.

4. Seaborn

Seaborn is a Python data visualization library that builds on top of Matplotlib, offering a high-level interface for creating visually appealing statistical graphics.

5. Sklearn

Sklearn, also known as Scikit-learn, is a popular Python library for machine learning tasks. It offers a comprehensive set of tools and algorithms for classification, regression, clustering, and dimensionality reduction. Built on top of NumPy and SciPy, scikit-learn provides a user-friendly API, making it accessible to both beginners and experienced practitioners.

4. DESIGN

4.1 Introduction

Our project can help predict the people who are likely to diagnose with a diabetes disease by help of their medical history. It recognizes who all are having any symptoms of diabetes such as change in insulin level or high blood pressure and can help in diagnosing disease with less medical tests and effective treatments, so that they can be cured accordingly.

The system consists of the following components:

- **Data Collection:** An organized dataset of individuals had been selected keeping in mind their history of diabetes and in accordance with other medical conditions.

- **Pre-processing:** Data pre-processing is an important step for the creation of a machine learning model. Initially, data may not be clean or in the required format for the model which can cause misleading outcomes. In pre-processing of data, we transform data into our required format. It is used to deal with noises, duplicates, and missing values of the dataset. Data pre-processing has the activities like importing datasets, splitting datasets, attribute scaling, etc. Preprocessing of the data is required for improving the accuracy of the model.

- **Algorithm Selection:** The model predicts whether the person is diabetic or not using variety of machine learning methods, such as KNN, SVM, Random forest and XGBoost.

- **Model training:** Training a machine learning (ML) model is a process in which a machine learning algorithm is fed with training data from which it can learn. The dataset is divided into training and testing data. The training dataset is used for prediction model learning and testing data

is used for evaluating the prediction model. For this project, 80% of training data is used and 20% of data is used for testing.

- **Model Evaluation:** On the testing set, all the chosen methods, will be assessed for accuracy, precision, recall and f1- score.

- **Comparative Analysis:** Based on their evaluation measures, the system will compare how well various machine learning algorithms perform in predicting diabetes disease. It is performed among algorithms and the algorithm that gives the highest accuracy is used for diabetes disease prediction.

- **Result Visualization:** An user interface application is created to predict whether the person is diabetic or not.

4.2 Architecture

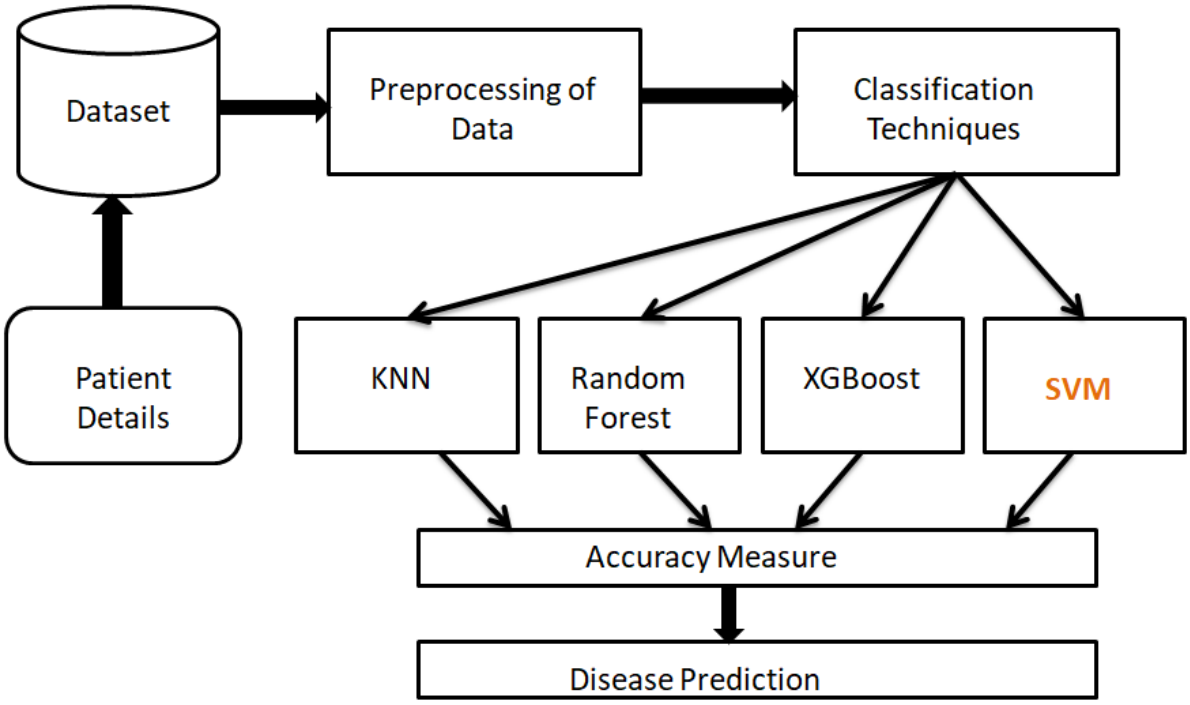

Figure 4.2 Architecture

The aim of 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. The algorithms like k nearest neighbors, Support Vector Machine and Decision Tree are used. The accuracy of the model using each of the algorithms is calculated. Then the one with good accuracy is taken as the model for predicting the diabetes.

5. SYSTEM ARCHITECTURE

5.1 Supervised Learning:

Supervised machine learning uses training datasets to achieve desired results. These dataset contains input and the correct output that helps the model to learn faster. For example, you want to train a machine to help you predict how long it will take you to drive home from your workplace.

Supervised learning in machine learning allows you to collect data or produce a data output from the previous experience. It also helps you to optimize performance using experience. Supervised machine learning helps you to solve various types of real-world computational problems.

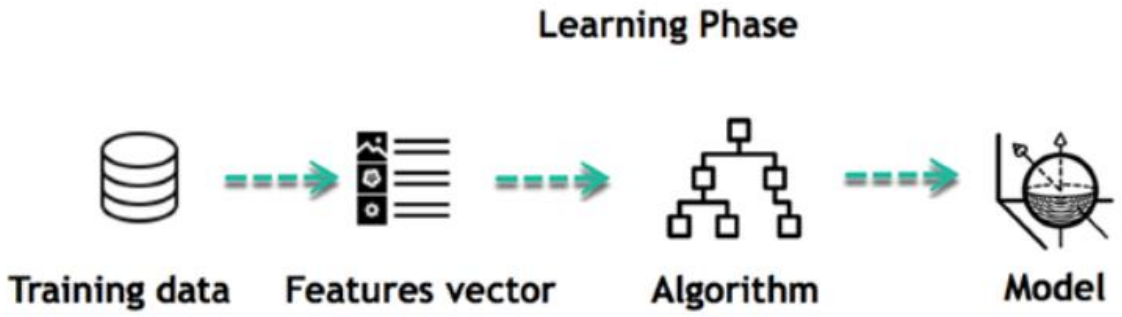

Figure 5.1 Supervised learning

5.2 Types of Supervised Machine learning Techniques

5.2.1 Regression

Regression technique predicts a single output value using training data. For example, you can use regression to predict the house price from training data. The input variables will be locality, size of a house, etc.

The output of the Regression always have a probabilistic interpretation, and the algorithm can be regularized to avoid overfitting.

5.2.2 Classification

Classification means to group the output inside a class. If the algorithm tries to label input into two distinct classes, it is called binary classification. Selecting between more than two classes is referred to as multi class classification. For example, determining whether or not someone will be a defaulter of the loan.

Classification tree perform very well in practice.

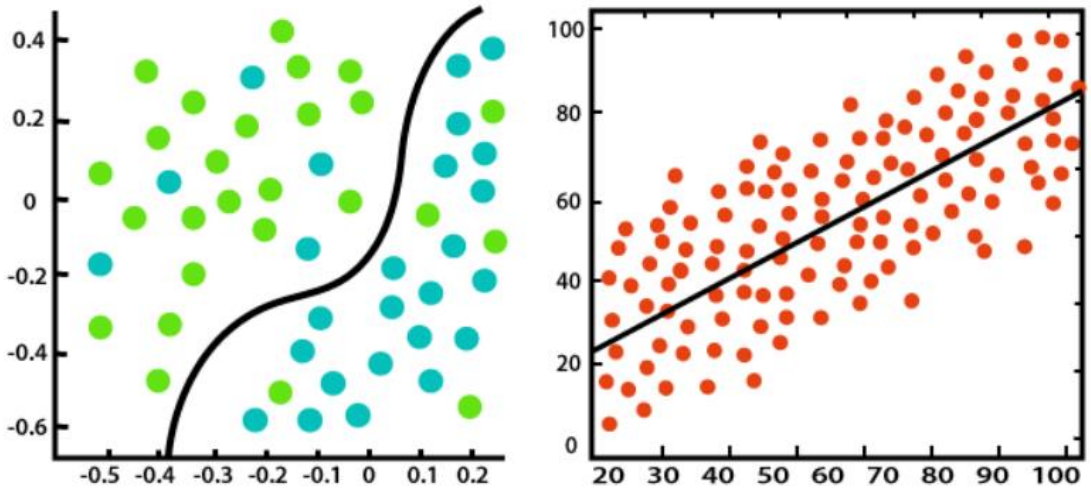

Figure 5.2 Classification and Regression

5.3 Supervised Algorithms used in this model

5.3.1 K-Nearest Neighbors

The k-nearest neighbors algorithm, also known as KNN or k-NN, is a non-parametric, supervised learning classifier, which uses proximity to make classifications or predictions about the grouping of an individual data point. While it can be used for either regression or classification problems, it is typically used as a classification algorithm, working off the assumption that similar points can be found near one another.

KNN is a reasonably simple classification technique that identifies the class in which a sample belongs by measuring its similarity with other nearby points. Though it is elementary to understand, it is a powerful technique for identifying the class of an unknown sample point.

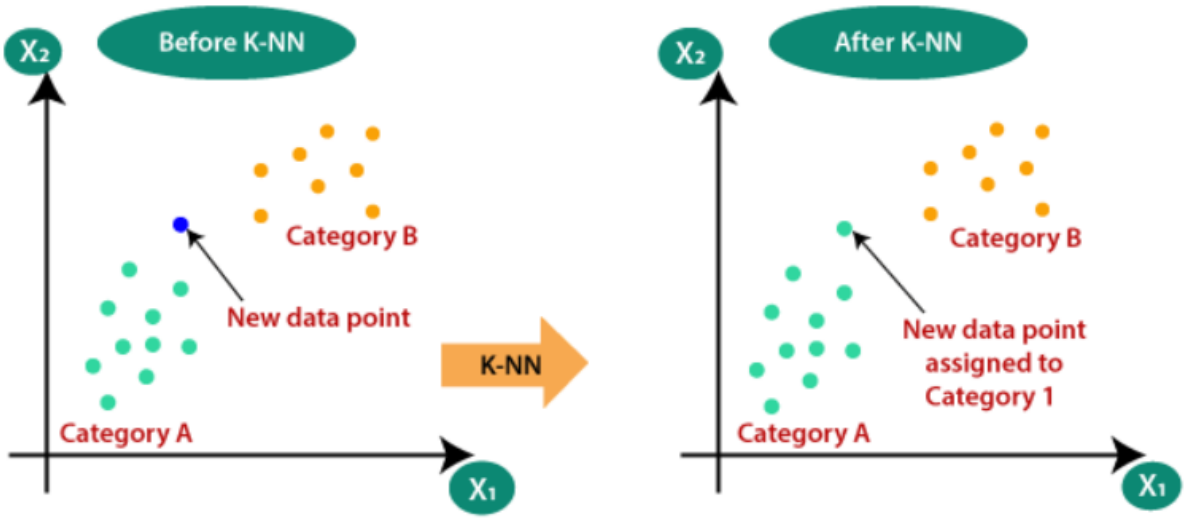

Figure 5.3.1 KNN

5.3.2 Random Forest

Every decision tree has high variance, but when we combine all of them together in parallel then the resultant variance is low as each decision tree gets perfectly trained on that particular sample data and hence the output does not depend on one decision tree but multiple decision trees.

Random Forest is a popular machine learning algorithm that belongs to the supervised learning technique. It can be used for both Classification and Regression problems in ML. It is based on the concept of ensemble learning, which is a process of combining multiple classifiers to solve a complex problem and to improve the performance of the model. Random forests are created from subsets of data, and the final output is based on average or majority ranking; hence the problem of overfitting is taken care of. It randomly selects observations, builds a decision tree, and takes the average result. It doesn't use any set of formulas.

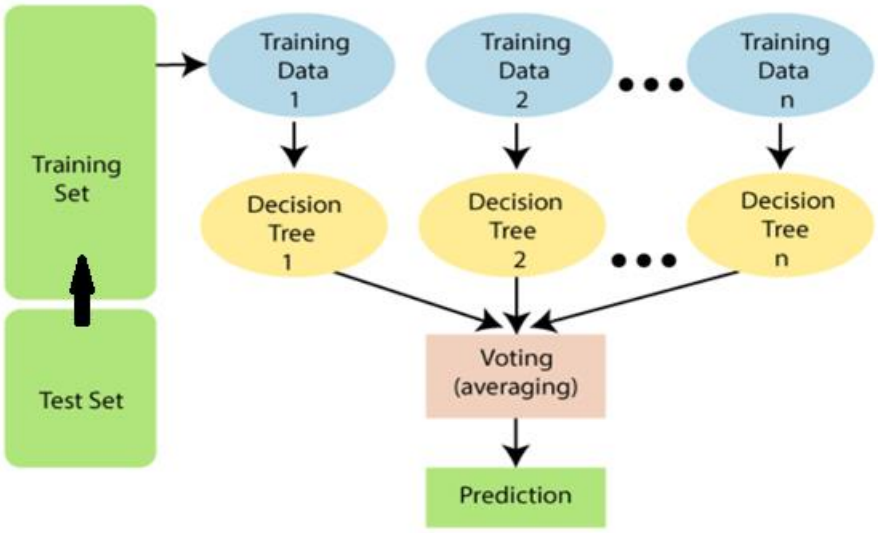

Figure 5.3.2 Random Forest

5.3.3 XGBoost

XGBoost is an implementation of Gradient Boosted decision trees. In this algorithm, decision trees are created in sequential form. Weights play an important role in XGBoost. Weights are assigned to all the independent variables which are then fed into the decision tree which predicts results.

XGBoost, which stands for Extreme Gradient Boosting, is a scalable, distributed gradient-boosted decision tree (GBDT) machine learning library. It provides parallel tree boosting and is the leading machine learning library for regression, classification, and ranking problems.

It's vital to an understanding of XGBoost to first grasp the machine learning concepts and algorithms that XGBoost builds upon: supervised machine learning, decision trees, ensemble learning, and gradient boosting.

A Gradient Boosting Decision Trees (GBDT) is a decision tree ensemble learning algorithm similar to random forest, for classification and regression. Ensemble learning algorithms combine multiple machine learning algorithms to obtain a better model.

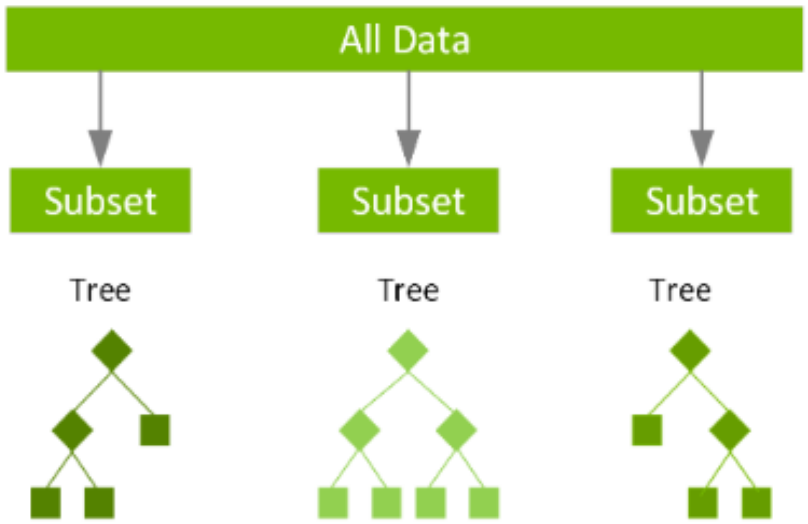

Figure 5.3.3 XGBoost

5.3.4 Support Vector Machine

Support Vector Machine or SVM is one of the most popular Supervised Learning algorithms, which is used for the Classification as well as Regression problems. However, primarily, it is used for Classification problems in Machine learning.

The goal of the SVM algorithm is to create the best line or decision boundary that can segregate n-dimensional space into classes so that we can easily put the new data point in the correct category in the future. This best decision boundary is called a hyperplane.

SVM chooses the extreme points/vectors that help in creating the hyperplane. These extreme cases are called as support vectors, and hence algorithm is termed as Support Vector Machine.

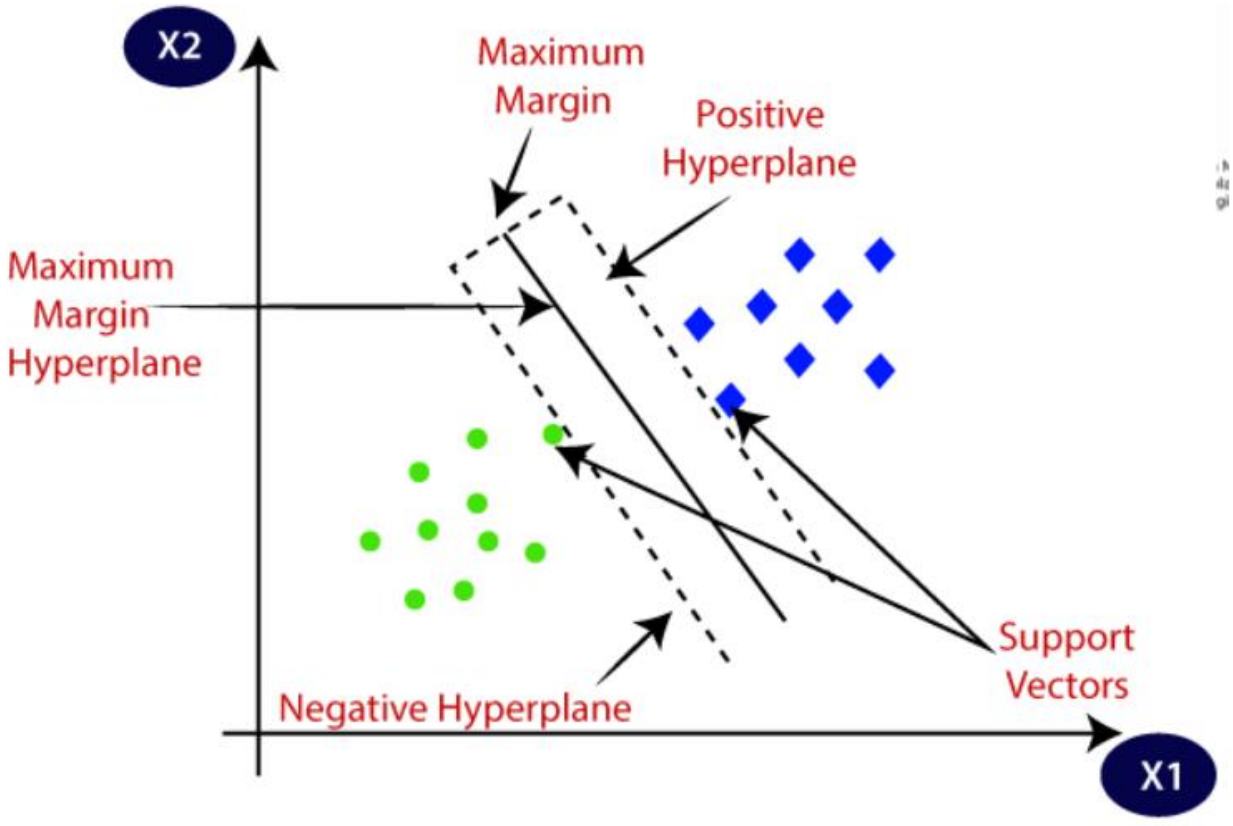

Figure 5.3.4 SVM

6. IMPLEMENTATION

6.1 Coding:

Git link : https://github.com/Nagajyothi-586/Major-Project

6.2 Training dataset Screenshots:

The dataset is taken from Kaggle which consists of 9 attributes and 2 classes (Yes-1, No-0) with Test size – 154 and Train size – 614.

	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

Figure 6.2 Sample Dataset

6.3 Evaluation Metrics

- **Confusion Matrix:**

The confusion matrix is a matrix used to determine the performance of the classification models , which aim to predict a categorical label for each input instance. The matrix displays the number of true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN) produced by the model on the test data.

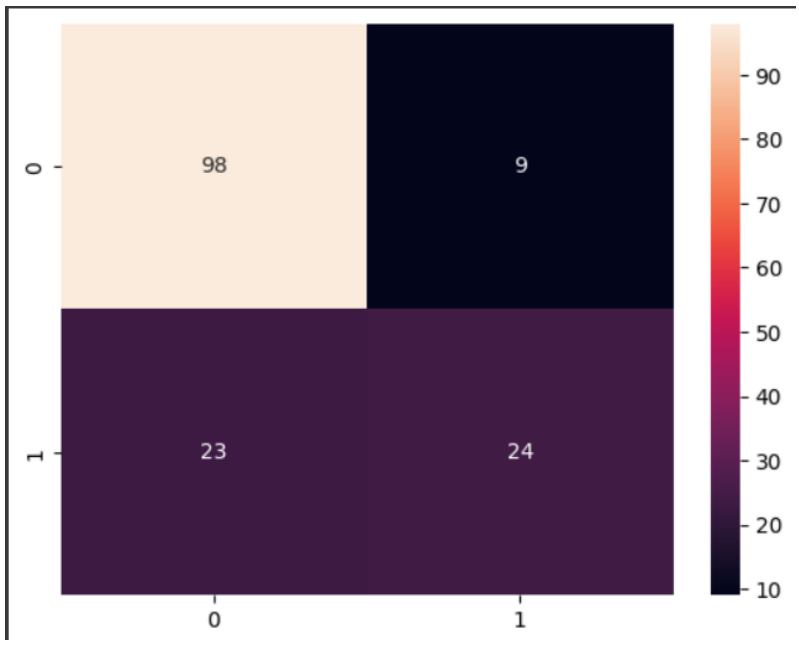

Figure 6.3 Confusion Matrix

- **Accuracy:**

  It defines how often the model predicts the correct output.

$$\textbf{Accuracy :} \quad \frac{TP+TN}{TP+FP+FN+TN}$$

- **Recall:**

  It corresponds to the ratio of the number of correctly classified positive items to the number of actual positive items.

$$\textbf{Recall :} \quad \frac{TP}{TP+FN}$$

- **Precision:**

  It refers to the ratio of true positive and the total positives predicted.

$$\textbf{Precision :} \quad \frac{TP}{TP+FP}$$

- **F1-Score:**

  It relates precision and recall metrics to obtain a quality measure that balances the relative importance of these two metrics.

$$F1 : \frac{2*(precision*recall)}{precision + recall}$$

6.4 Accuracy Comparison

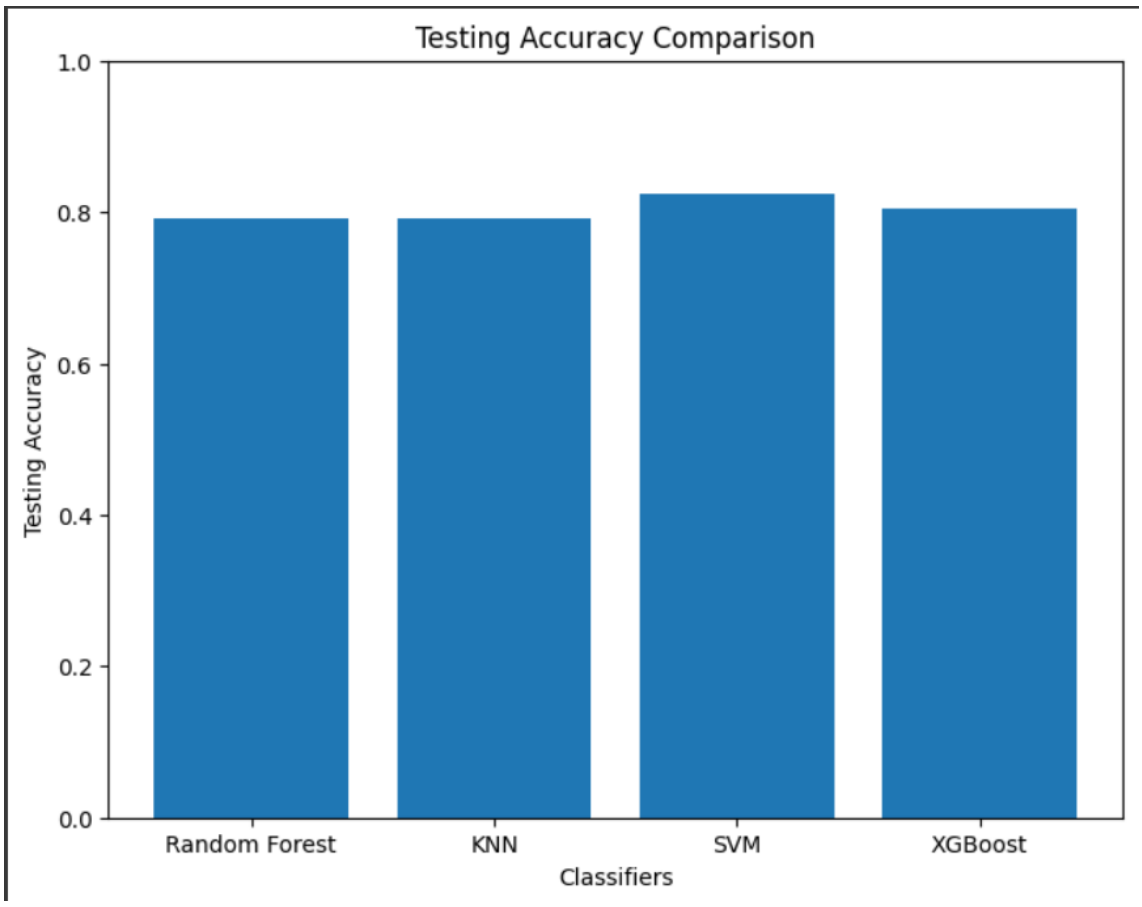

Figure 6.4 Comparison of Algorithms

6.5 Evaluation Metrics Comparison

S.No	Algorithm	Accuracy(%)	Precision	Recall	F1-Score
1	KNN	79	0.72	0.51	0.60
2	Random Forest	79	0.72	0.68	0.70
3	XGBoost	80	0.68	0.66	0.67
4	SVM	82	0.76	0.61	0.68

Table 6.5 Table of Comparison

7. RESULTS

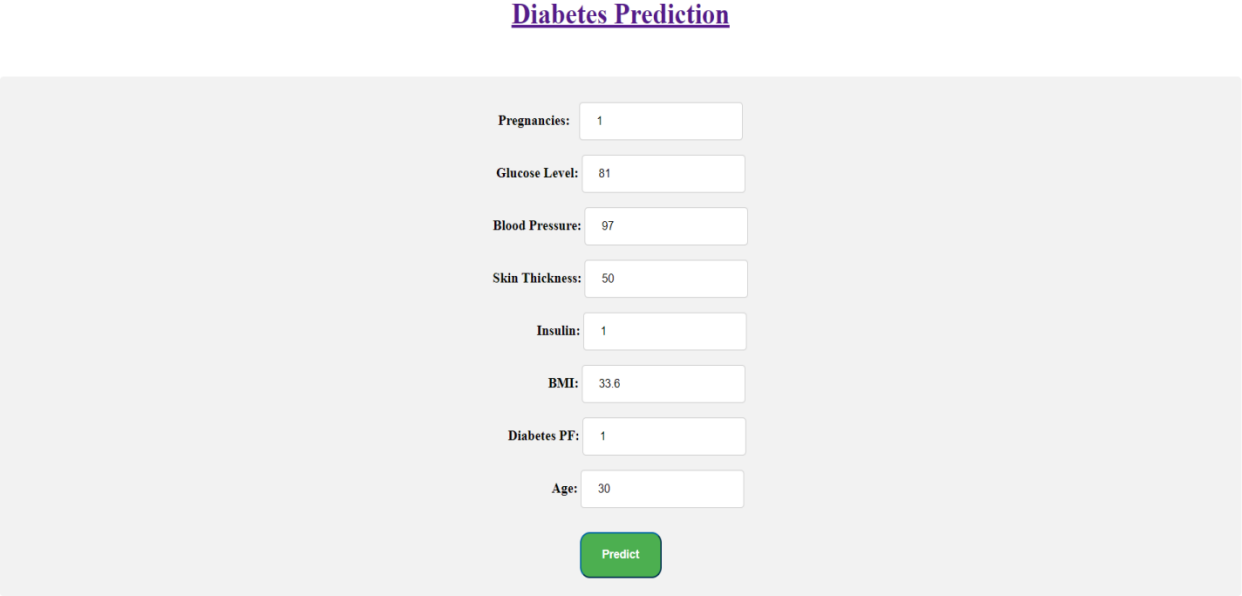

Figure 7.1 Input Details 1

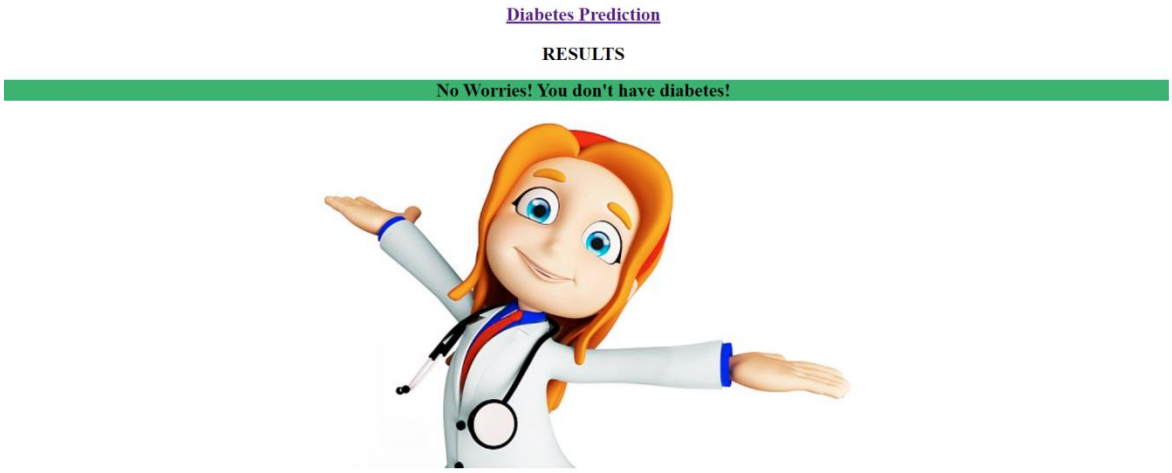

Figure 7.2 Results 1

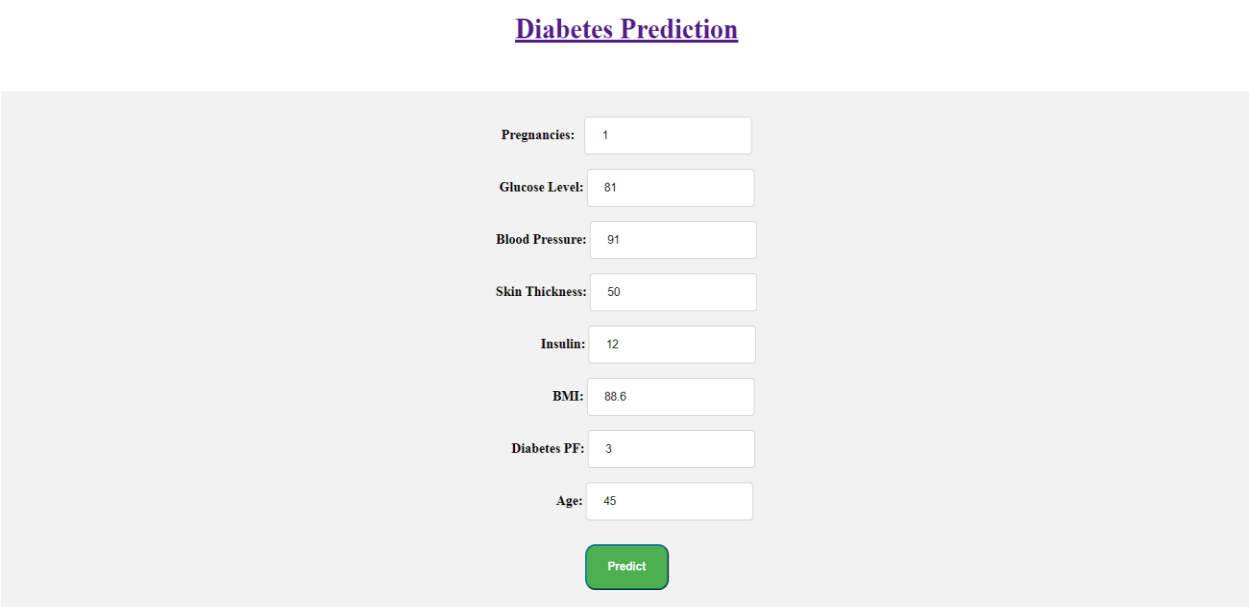

Figure 7.3 Input Details 2

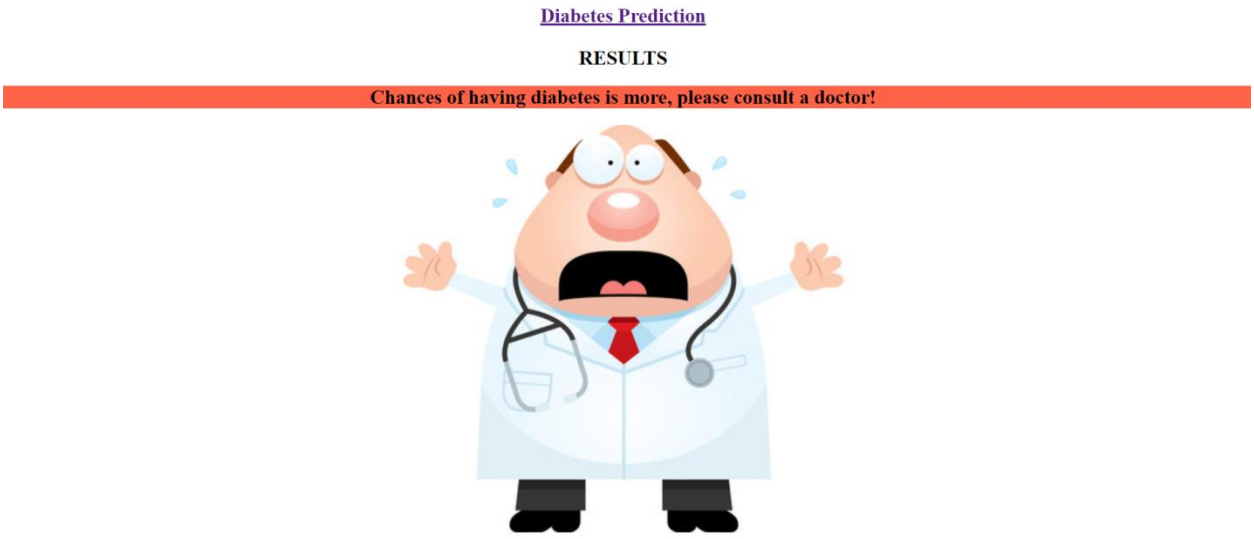

Figure 7.4 Results 2

8. CONCLUSION AND FUTURE SCOPE

- Diabetes disease prediction is a major challenge in the present modern life. This project aims to predict the disease on the basis of symptoms. With this application, the patient is able to find whether they are diabetic or not.
- With this application if the patient/user is away from reach of doctor, they can make use of the application in prediction of disease just by entering the values.
- For future scope, Deep Learning approaches can be used for better analysis of diabetes disease and for earlier prediction of diseases so that the rate of death cases can be minimized

9. REFERENCES

[1] Arwatki Chen Lyngdoh, Soumen Moulik (2021) , Diabetes Disease Prediction using Machine Learning Algorithms.

[2] Sivaranjan S,Ananya S, Aravinth J, Karthika R (2021), Diabetes Prediction using Machine Learning Algorithms with Feature Selection and Dimensionality Reduction.

[3] Ayman Mir, Sudhir N.Dhage (2019) , Diabetes Disease Prediction using Machine Learning on Big Data of Healthcare.

[4] Gaurav Tripathi , Rakesh Kumar (2020),Early Prediction of Diabetes Mellitus using Machine Learning.

[5] Priyanka Sonar, Prof. K. JayaMalini(2019), Diabetes Prediction using different Machine Learning Approaches.

[6] Aakansha Rathore and Simran Chauhan(2017), Detecting and Predicting Diabetes Using Unsupervised Learning.

[7] Deeraj Shetty, Kishor Rit, Sohail shaikh and Nikita Patil(2016), Diabetes Disease Prediction Using Data Mining.

[8] S.Yadav and S. Shukla(2016), Analysis of k-fold cross-validation over hold-out validation on colossal datasets for quality classification.

[9] A. Mir and S. N. Dhage(2018), Diabetes disease prediction using Machine Learning on Bigdata of Healthcare.

[10] D. Sissodia and D. S. Sissodia(2018), Prediction of Diabetes using Classification Algorithm

10. APPENDIX

IMPORTING LIBRARIES AND DATASET

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn import svm
from sklearn.metrics import accuracy_score
from xgboost import XGBClassifier
from sklearn.metrics import precision_score,recall_score,f1_score
dataset = pd.read_csv('diabetes.csv')
```

DATA PREPROCESSING

```
#head returns the first five rows of the dataset.
dataset.head()
#shape shows the number of rows and columns present in the dataset.
dataset.shape
```

#info returns information about the dataset.

```
dataset.info()
```

The describe() function is used to generate descriptive statistics that summarize the dataset.

```
dataset.describe()
```

#countplot used to represent the occurrence(counts) of the observation present in the categorical variable.

```
sns.countplot(x = 'Outcome',data = dataset)
```

The corr() method finds the correlation of each column in a DataFrame.

```
corr_matrix = dataset.corr()
```

A heatmap is a graphical representation of data where the individual values contained in a matrix are represented as colors. Darker colors indicate stronger correlations, while lighter colors indicate weaker correlations.

```
sns.heatmap(corr_matrix, annot = True)
```

```
plt.show()
```

isnull(). sum(). returns the number of missing values in the dataset.

```
dataset.isna().sum()
```

TRAINING AND TESTING

#split the data into training and testing data

```
X = dataset.iloc[:, :-1].values
```

```
Y = dataset.iloc[: , -1].values
```

```
X[0]
```

#80% is training data and 20% is testing data

x_train , x_test , y_train, y_test = train_test_split(X,Y,test_size = 0.2,random_state = 0)

x_train.shape #returns no of rows present in training data

x_test.shape   #returns no of rows present in testing data

FEATURE SCALING

from sklearn.preprocessing import StandardScaler #feature scaling

sc = StandardScaler()

x_train = sc.fit_transform(x_train)

x_test = sc.transform(x_test)

KNN

from sklearn.neighbors import KNeighborsClassifier

knn = KNeighborsClassifier(n_neighbors = 25, metric = "minkowski")

knn.fit(x_train,y_train)

y_pred = knn.predict(x_test)

from sklearn.metrics import confusion_matrix

c_mat = confusion_matrix(y_test,y_pred)

sns.heatmap(c_mat, annot = True)

from sklearn.metrics import accuracy_score

knn_acc = accuracy_score(y_test,y_pred)

```
print(knn_acc)
```

Calculation of Precision, Recall and F1-score:

```
print('Precision: %.3f' % precision_score(y_test, y_pred))
print('Recall: %.3f' % recall_score(y_test, y_pred))
print('F1 Score: %.3f' % f1_score(y_test, y_pred))
```

SVM

```
classifier = svm.SVC(kernel='linear')
classifier.fit(x_train,y_train)
X_train_prediction = classifier.predict(x_test)
data_accuracy = accuracy_score(y_test,X_train_prediction)
print('Accuracy: ', data_accuracy)
```

Calculation of Precision, Recall and F1-score:

```
print('Precision: %.3f' % precision_score(y_test, X_train_prediction)) #svm
print('Recall: %.3f' % recall_score(y_test, X_train_prediction))
print('F1 Score: %.3f' % f1_score(y_test, X_train_prediction))
```

RANDOM FOREST

```
from sklearn.ensemble import RandomForestClassifier
rf = RandomForestClassifier(random_state = 42,max_depth=5)
rf.fit(x_train,y_train)
predictions = rf.predict(x_test)
```

```
accuracy = accuracy_score(y_test,predictions)

print(accuracy)
```

Calculation of Precision, Recall and F1-score:

```
print('Precision: %.3f' % precision_score(y_test, predictions)) #RandomForest

print('Recall: %.3f' % recall_score(y_test, predictions))

print('F1 Score: %.3f' % f1_score(y_test, predictions))
```

XGBOOST

```
model_1 = XGBClassifier(random_state = 42,
max_depth=5,learning_rate=0.001, n_estimators=1000)

model_1.fit(x_train,y_train)

x_pred_1 = model_1.predict(x_test)

xg_acc=accuracy_score(y_test,x_pred_1)

print(xg_acc)
```

Calculation of Precision, Recall and F1-score:

```
print('Precision: %.3f' % precision_score(y_test, x_pred_1)) #XGBoost

print('Recall: %.3f' % recall_score(y_test, x_pred_1))

print('F1 Score: %.3f' % f1_score(y_test, x_pred_1))
```

ACCURACY COMPARISON

```
classifiers = ['Random Forest', 'KNN','SVM' ,'XGBoost']

test_accuracy = [accuracy,knn_acc, data_accuracy,xg_acc]
```

```
plt.figure(figsize=(8, 6))
plt.bar(classifiers, test_accuracy)
plt.xlabel('Classifiers')
plt.ylabel('Testing Accuracy')
plt.title('Testing Accuracy Comparison')
plt.ylim(0, 1)
plt.show()
```

USER INTERFACE

```
import pickle
pickle.dump(knn, open('classifier.pkl','wb'))
pickle.dump(sc, open('sc.pkl','wb'))
import numpy as np #app.py
from flask import Flask, request, render_template
import pickle
app = Flask(__name__)
sc = pickle.load(open('sc.pkl','rb'))
model = pickle.load(open('classifier.pkl','rb'))
@app.route('/')
def home():
    return render_template('index.htm')
@app.route('/predict',methods=['POST'])
```

```
def predict():
    float_features = [float(x) for x in request.form.values()]
    final_features = [np.array(float_features)]
    pred = model.predict(sc.transform(final_features))
    return render_template('result.html',prediction = pred)
if __name__ == "__main__":
    app.run(debug = True)
#index page
<!DOCTYPE html>
<html>
<style>
input[type=number], select {
  width: 100%;
  padding: 12px 20px;
  margin: 8px 0;
  display: inline-block;
  border: 1px solid #ccc;
  border-radius: 4px;
  box-sizing: border-box;
}
input[type=submit] {
  width: 100%;
```

```
  background-color: #4CAF50;
  color: white;
  padding: 14px 20px;
  margin: 8px 0;
  border: none;
  border-radius: 4px;
  cursor: pointer;
}
div {
  border-radius: 5px;
  background-color: #f2f2f2;
  padding: 20px;
}
.registration {
        background: #4CAF50;
        color: white;
        border-style: outset;
        border-color: #0066A2;
        height: 50px;
        width: 100px;
        font: bold15px arial,sans-serif;
        text-shadow: none;
```

```
    border-radius: 12px;
    cursor: pointer;
}
</style>
<title> Diabetes </title>
<h1> <a href = '/'> <p style = "text-align:center;">Diabetes Prediction</p></a>
</h1>
<center><br>
<div>
<form action = "/predict"method="post">
<label for="Pregnancies"> <b>Pregnancies:   </b></label>
<input type = "number" step = "any" min="0" id = "Pregnancies"name =
"Pregnancies" placeholder= "Enter positive digit" required = "required"
style="width: 200px;"/> <br>
<label for="Glucose Level"> <b>Glucose Level: </b></label><input type =
"number" step = "any" min="0" id = "Glucose Level" name = "Glucose Level"
placeholder= "Enter positive digit" required = "required" style="width:
200px;"/> <br>
<label for="Blood Pressure"> <b>Blood Pressure: </b></label>
<input type = "number" step = "any"  min="0" name = "Blood Pressure"
placeholder= "Enter positive digit" required = "required" style="width:
200px;"/> <br>
<label for="Skin Thickness"> <b>Skin Thickness: </b></label>
<input type = "number" step = "any" min="0" name = "Skin Thickness"
placeholder= "Enter positive digit" required = "required" style="width:
200px;"/> <br>
```

<label for="Insulin">   Insulin: </label>

<input type = "number" step = "any" min="0" name = "Insulin" placeholder= "Enter positive digit" required = "required" style="width: 200px;"/>

<label for="BMI">   BMI: </label>

<input type = "number" step = "any" min="0" name = "BMI" placeholder= "Enter positive digit" required = "required" style="width: 200px;"/>

<label for="Diabetes PF">   Diabetes PF: </label>

<input type = "number" step = "any" min="0" name = "Diabetes PF" placeholder= "Enter positive digit" required = "required" style="width: 200px;"/>

<label for="Age">   Age: </label>

<input type = "number" step = "any" min="0" name = "Age" placeholder= "Enter positive digit" required = "required" style="width: 200px;"/>

<button type="submit" class = "registration"> Predict</button>

</form>

</html>

#result page

<!DOCTYPE html>

<html>

```
<body>

<center>

<h2> <a href='/'> <p style = "text-align:center;">Diabetes Prediction</p></a></h2>

<h2> RESULTS </h2>

{% if prediction == 1%}

<h2 style = "background-color:Tomato;">Chances of having diabetes is more, please consult a doctor! </h2>

<img src="https://t4.ftcdn.net/jpg/01/59/92/75/360_F_159927541_IO2QsQNT67ZbjVSShiwXu14JpSfuaGPj.jpg" alt="Paris" style="width:35%;"> <br>

{%elif prediction == 0%}

<h2 style = "background-color:MediumSeaGreen;">No Worries! You don't have diabetes! </h2>

<img src="https://i.dailymail.co.uk/i/pix/2015/08/29/00/2BBFC0F900000578-0-image-a-36_1440805913150.jpg" alt="Paris" style="width:50%;"> <br>

{% endif %}

</center>

</body>

</html>
```