DIABETES PREDICTION USING MACHINE LEARNING

IT8611

A MINI-PROJECT REPORT

Submitted by

T. SELVA SATHISH **312019205028**

B. SHARATH **312019205030**

in partial fulfilment for the award of the

degree of

BACHELOR OF TECHNOLOGY

IN

INFORMATION TECHNOLOGY

JEPPIAAR SRR ENGINEERING COLLEGE, PADUR

ANNA UNIVERSITY: CHENNAI 600 025

JUNE 2022

ANNA UNIVERSITY: CHENNAI 600 025

BONAFIDE CERTIFICATE

Certified that this project report **" DIABETES PREDICTION USING MACHINE LEARNING "** is the bonafide work of **T. SELVA SATHISH(312019205028) AND B.SHARATH(312019205030)** who carried out the project work under by supervision.

HEAD OF THE DEPARTMENT
Mr. S.RAMAKRISHNAN M.E.,
Assistant Professor,
Information Technology,
Jeppiaar SRR Engineering
College,

INTERNAL GUIDE
Mr. S.RAMAKRISHNAN M.E.,
Assistant Professor,
Information Technology,
Jeppiaar SRR Engineering
College,

Submitted for the examination held on_____.

INTERNAL EXAMINER **EXTERNAL EXAMINER**

ACKNOWLEGEMENT

We take this opportunity to express our profound gratitude and deep regard to our beloved **Founder Chairman (Late)Col. Dr. JEPPIAAR M.A., B.L., Ph.D.,** for enlightening our lives and showering heavenly blessings forever.

We also express our heartfelt thanks to our **Chairman and Managing Director Dr. REGEENA JEPPIAAR B.E., M.B.A., Ph.D.,** for her kind cooperation and keen interest for the success of the project.

We are immensely happy to accord the warmth of gratitude to our **Director Mr. MURLI SUBRAMANIAN** for being the beacon in all our endeavours.

We express our profound gratitude to our **Principal Dr. M. Sasikumar M.Tech., Ph.D** for bringing out novelty in all executions.

We express our thanks to our **Head of the Department Mr. S. RAMAKRISHNAN M.E.,** for his valuable suggestions and guidance for the development and completion of this project.

We are highly thankful to our project **Internal Guide Mr. S. RAMAKRISHNAN M.E.,** for guidance and encouragement in carrying out this project work.

We are much obliged to all our teaching and non-teaching staff members for their valuable information and constructive criticism that immensely contributed to the development of the project.

Above all, we wish to avail this opportunity to express a sense of gratitude and love to our beloved parents and friends for their moral support and constant strength at various stages of our project

TABLE OF CONTENTS

CHAPTER 1

ABSTRACT

Diabetes is an illness caused because of high glucose Level in a human body. Diabetes should not be ignored if it is untreated then Diabetes may cause some major issues in a person like: heart related problems, kidney problem, blood pressure, eye damage and it can also affects other organs of human body. Diabetes can be controlled if it is predicted earlier.To achieve this goal this project work we will do early prediction of Diabetes in a 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.

Keywords: Diabetes, Machine Learninng Prediction, Dataset.

INTRODUCTION

Machine learning is a sub-domain of computer science which evolved from the study of pattern recognition in data, and also from the computational learning theory in artificial intelligence. It is the first-class ticket to most interesting careers in data analytics today[1]. As data sources proliferate along with the computing power to process them, going straight to the data is one of the most straightforward ways to quickly gain insights and make predictions.

Diabetes mellitus is the most common disease worldwide and keeps increasing everyday due to changing lifestyle, unhealthy food habits and over weight problemsThere were studies handle in prediction diabetes mellitus through physical and chemical tests, are available for diagnosing diabetes.Data science methods have the potential to benefit other scientific fields by shedding new light on common questions.

1.1 PROJECT OBJECTIVE

- The objective of the study is classify Indian PIMA dataset for diabetes.
- This is proposed to achieve through machine learning classification algorithm.
- Our objective is to design an interactive application, in which user can give inputs to arrive the prediction

1.2 PROJECT DESCRIPTION

This project work we will do early prediction of Diabetes in a 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.

CHAPTER 2

SYSTEM REQUIREMENTS

2.1 HARDWARE REQUIREMENTS

- Processor : Any processor above 500 mhz
- Ram : 4 GB
- Hard Disk : 4 GB
- Input device : Standard Keyboard&Mouse
- Output device Monitor: VGA and High Resolution

2.2 SOFTWARE REQUIREMENTS

- Operating System : Windows 7 or higher
- Programming : Python 3.6 or higher
- Python Libraries : Numpy, Pandas , Matplotlib , Sklearn , Pickle

CHAPTER 3

SOFTWARE SPECIFICATION

3.1 SPECIFICATION

A software requirements specification (SRS) is a document that describes what the software will do and how it will be expected to perform. It also describes the functionality the product needs to fulfill all stakeholders (business, users) needs. A software requirements specification is the basis for your entire project. It lays the framework that every team involved in development will follow. It's used to provide critical information to multiple teams - development, quality assurance, operations, and maintenance. This keeps everyone on the same page.

Using the SRS helps to ensure requirements are fulfilled. And it can also help you make decisions about your product's lifecycle - for instance when to retire a feature.

3.2 FUNCTIONAL REQUIREMENTS

Functional requirements may involve calculations,

technical details, data manipulation and processing, and other specific functionality that define what a system is supposed to accomplish. Behavioral requirements describe all the cases where the system uses the functional requirements, these are captured in use cases.

3.3 ABOUT THE DATASET

This dataset is originally from the National Institute of Diabetes and Digestive and Kidney Diseases. It is provided courtesy of the Pima Indians Diabetes Database and is available on Kaggle. Here is the link to the dataset. It consists of several medical predictor variables and one target variable, Outcome. Predictor variables include the number of pregnancies the patient has had, their BMI, insulin level, age, and so on. The dataset has 7 columns as shown below;

Glucose – Plasma glucose concentration a 2 hours in an oral glucose tolerance test

BloodPressure – Diastolic blood pressure (mm Hg)

SkinThickness – Triceps skinfold thickness (mm)

Insulin – 2-Hour serum insulin (mu U/ml)

BMI – Body mass index (weight in kg/(height in m)^2)

DiabetesPedigreeFunction – Diabetes pedigree function

Age – Age (years)

3.4 MODULES

Systems design is the process of defining the architecture, modules, interfaces, and data for a system to satisfy specified requirements. Systems design could be seen as the application of systems theory to product development. This chapter gives the overall view of the module's description and the proposed architecture of the project.

3.4.1 ALGORITHMS

Decision Trees

A decision tree is built by repeatedly asking questions to the partition data. The aim of the decision tree algorithm is

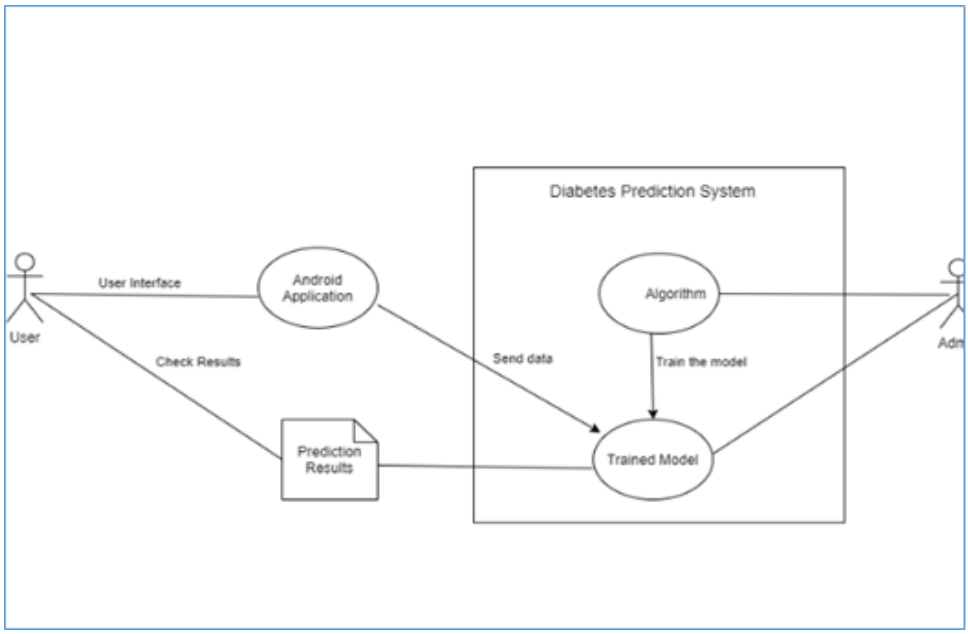
to **increase the predictiveness at each level of partitioning so that the model is always updated with information about the dataset.**

Even though it is a **Supervised Machine Learning algorithm**, it is used mainly for **classification rather than regression**. In a nutshell, the model takes a particular instance, traverses the decision tree by comparing important features with a conditional statement. As it descends to the left child branch or right child branch of the tree, depending on the result, the features that are more important are closer to the root. The good part about this machine learning algorithm is that **it works on both continuous dependent and categorical variables.**

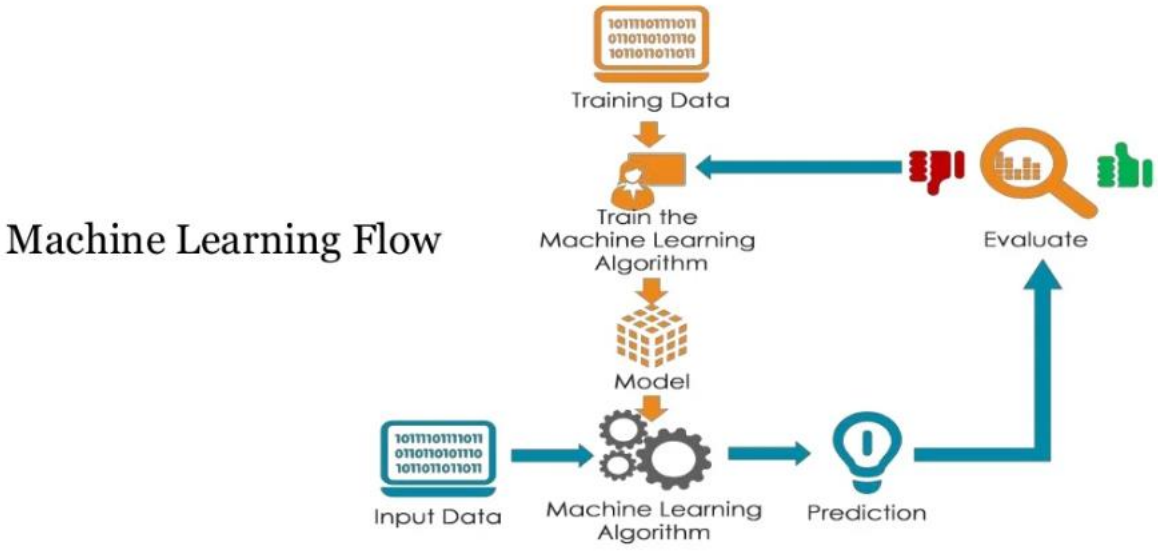
CHAPTER 4

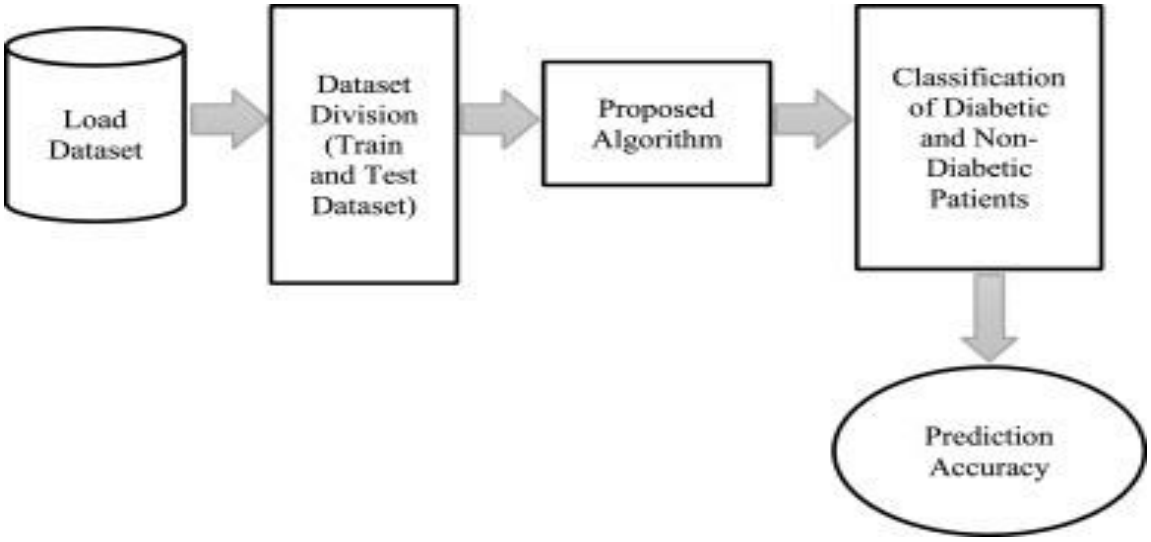
DIAGRAMATICAL REPRENSENTATION

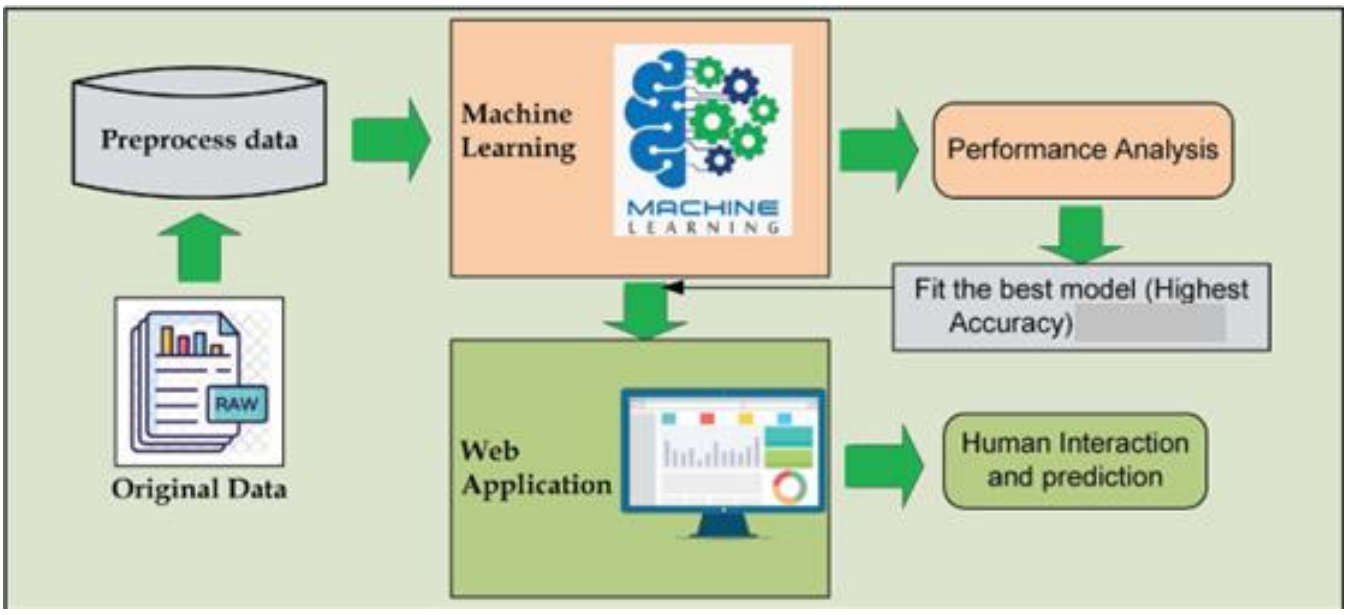
4.1 USE CASE DIAGRAM



4.2 DATA FLOW DIAGRAM




Load Dataset
Dataset Division (Train and Test Dataset)
Proposed Algorithm
Classification of Diabetic and Non-Diabetic Patients
Prediction Accuracy


Preprocess data
Machine Learning
MACHINE LEARNING
Performance Analysis
Fit the best model (Highest Accuracy)
RAW
Original Data
Web Application
Human Interaction and prediction

CHAPTER 5

IMPLEMENTATION

Project implementation (or project execution) is the phase where visions and plans become reality. This is the logical conclusion, after evaluating, deciding, visioning, planning, applying for funds and finding the financial resources of a project.

MACHINE LEARNING CODE:-

```
import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

from sklearn import metrics

from sklearn.model_selection import train_test_split

from sklearn.linear_model import LogisticRegression

import pickle
```

```
diabetes=pd.read_csv(r"C:\Users\Admin\Desktop\trial\diabetes.csv”)

diabetes.columns

diabetes.isnull().any()

X=diabetes[['Glucose','BloodPressure','SkinThickness','Insulin','BMI','DiabetesPedigreeFunction', 'Age']]

X.shape

y=diabetes[['Outcome']]

y.shape

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=20)

X_train.head()

y_train.head()

dia=LogisticRegression()

dia.fit(X_train,y_train)
```

```
X_test

y_pred=dia.predict(X_test)

y_pred

diabetes['Outcome'].unique()

metrics.accuracy_score(y_pred,y_test)*100

dia.score(X_test,y_test)*100

X_train=X_train.values

X_train


##TO CHECK HIGH ACCURARY ALGORITHM

from sklearn.linear_model import LogisticRegression

LR=LogisticRegression(random_state=0,max_iter=3000)

LR.fit(X_train, y_train)

p1=LR.score(X_test,y_test)*100
```

```
print(p1)


from sklearn.ensemble import AdaBoostClassifier
ADA=AdaBoostClassifier()
ADA.fit(X_train, y_train)
p2=ADA.score(X_test,y_test)*100
print(p2)


from sklearn.ensemble import RandomForestClassifier
RF=RandomForestClassifier(max_features='auto',
n_estimators=200)
RF.fit(X_train, y_train)
p3=RF.score(X_test,y_test)*100
print(p3)
```

```
from sklearn.tree import DecisionTreeClassifier

DC=DecisionTreeClassifier()

DC.fit(X_train, y_train)

p4=DC.score(X_test,y_test)*100

print(p4)


from sklearn.naive_bayes import GaussianNB

GB=GaussianNB()

GB.fit(X_train,y_train)

p5=GB.score(X_test,y_test)*100

print(p5)


a=["LogisticRegression","AdaBoostClassifier","RandomFor
estClassifier","DecisionTreeClassifier","GaussianNB"]

b=[p1,p2,p3,p4,p5]
```

```
plt.figure(figsize=(15,6))

plt.bar(a,b)

plt.title("Accuracy Graph")

plt.xlabel("Algorithm")

plt.ylabel("percentage")

plt.show()


##FINAL PART

result=RF.predict([[84,82,31,125,38.2,0.233,23]])

result

result_perc=RF.predict_proba([[84,82,31,125,38.2,0.233,2
3]])

result_perc*100

max(result_perc[0])*100

if(result==1):
```

```
        print((max(result_perc[0])*100),"% You are having Diabetics")
    else:
        print((max(result_perc[0])*100),"% You are not having Diabetics")


    ##LOADING CODE INTO FILE
    file=open("dia.pkl","wb")
    pickle.dump(DC,file)
    file.close()
```

FRONT-END CODE:-

```
    from tkinter import*
    import pickle
    import sklearn
```

```
dia_pred=pickle.load(open("dia.pkl","rb"))

dia=Tk()

dia.title("Diabetes Prediction")

dia.geometry("750x900")

dia.configure(background="#d446cd")

gl_input=DoubleVar()

bp_input=DoubleVar()

sk_input=DoubleVar()

ins_input=DoubleVar()

bmi_input=DoubleVar()

dpf_input=DoubleVar()

age_input=DoubleVar()
```

```
    def prediction():
        gl=gl_input.get()
        bp=bp_input.get()
        sk=sk_input.get()
        ins=ins_input.get()
        bmi=bmi_input.get()
        dpf=dpf_input.get()
        age=age_input.get()
        if ((70<=gl<=350) and (80<=bp<=150) and (2<=sk<=29)
and (10<=ins<=300) and (12<=bmi<=45) and (0<=dpf<=2.5) and
(0<=age<=150)):
            result=dia_pred.predict([[gl,bp,sk,ins,bmi,dpf,age]])

result_perc=dia_pred.predict_proba([[gl,bp,sk,ins,bmi,dpf,age]]
)
```

```
            if(result==1):
                ans=str(round(max(result_perc[0])*100))+"%      You
are having Diabetics"
            else:
                ans=str(round(max(result_perc[0])*100))+"%      You
are not having Diabetics"
        else:
            ans="Invalid Details"
        lb8.configure(text="Prediction: ")
        lb81.configure(text=ans)


    lb1=Label(dia,text="Enter the Glucose level\t\t:
",font=('algerian',15),fg="black",bg="#d446cd")
    lb1.grid(row=0,column=0,padx=(0,10),pady=10)
    ent1=Entry(dia,textvariable=gl_input,font=('copperblack',1
```

```
5),fg="black",bg="white")

    ent1.grid(row=0,column=1)


    lb2=Label(dia,text="Enter the Blood pressure\t\t:
",font=('algerian',15),fg="black",bg="#d446cd")

    lb2.grid(row=1,column=0,padx=(0,10),pady=10)

    ent2=Entry(dia,textvariable=bp_input,font=('copperblack',
15),fg="black",bg="white")

    ent2.grid(row=1,column=1)


    lb3=Label(dia,text="Enter the Skin thickness\t\t:
",font=('algerian',15),fg="black",bg="#d446cd")

    lb3.grid(row=2,column=0,padx=(0,10),pady=10)

    ent3=Entry(dia,textvariable=sk_input,font=('copperblack',
15),fg="black",bg="white")
```

```
ent3.grid(row=2,column=1)


lb4=Label(dia,text="Enter the Insulin\t\t\t :
",font=('algerian',15),fg="black",bg="#d446cd")

lb4.grid(row=3,column=0,padx=(30,25),pady=10)

ent4=Entry(dia,textvariable=ins_input,font=('copperblack',
15),fg="black",bg="white")

ent4.grid(row=3,column=1)


lb5=Label(dia,text="Enter the BMI value \t\t:
",font=('algerian',15),fg="black",bg="#d446cd")

lb5.grid(row=4,column=0,padx=(0,10),pady=10)

ent5=Entry(dia,textvariable=bmi_input,font=('copperblack'
,15),fg="black",bg="white")

ent5.grid(row=4,column=1)
```

```
lb6=Label(dia,text="Enter the Diabetes pedigree function :
",font=('algerian',15),fg="black",bg="#d446cd")

lb6.grid(row=5,column=0,padx=(20,10),pady=10)

ent6=Entry(dia,textvariable=dpf_input,font=('copperblack'
,15),fg="black",bg="white")

ent6.grid(row=5,column=1)


lb7=Label(dia,text="Enter the Age\t\t \t:
",font=('algerian',15),fg="black",bg="#d446cd")

lb7.grid(row=6,column=0,padx=(0,10),pady=10)

ent7=Entry(dia,textvariable=age_input,font=('copperblack'
,15),fg="black",bg="white")

ent7.grid(row=6,column=1)
```

```
    btn=Button(dia,comman=prediction,text="PERDICT",font=
('aerial',12),fg="black",bg="silver",activebackground="black",ac
tiveforeground="silver")

    btn.grid(row=7,column=1,pady=(10,10))



    lb8=Label(dia,font=('algerian',15),fg="black",bg="#d446cd"
)

    lb8.grid(row=8,column=0,padx=(50,10),pady=10)

    lb81=Label(dia,font=('algerian',15),fg="black",bg="#d446c
d")

    lb81.grid(row=8,column=1,padx=(50,10),pady=10)


    dia.mainloop()
```

CHAPTER 6

SOFTWARE TEST DESCRIPTION

The best practices for testing traditional software systems and developing high-quality software.

A typical software testing suite will include:

- **Unit tests** which operate on atomic pieces of the codebase and can be run quickly during development,
- **Regression tests** replicate bugs that we've previously encountered and fixed,
- **Integration tests** which are typically longer-running tests that observe higher-level behaviors that leverage multiple components in the codebase,

Follow conventions such as:

- don't merge code unless all tests are passing,
- always write tests for newly introduced logic when contributing code,

- when contributing a bug fix, be sure to write a test to capture the bug and prevent future regressions.

6.1 UNIT TESTING

It is a type of software testing where individual units or components of a software are tested. The purpose is to validate that each unit of the software code performs as expected. Unit Testing is done during the development (coding phase) of an application by the developers. Unit Tests isolate a section of code and verify its correctness. A unit may be an individual function, method, procedure, module, or object. In our project we are testing each and every algorithms and functions

6.2 INTEGRATION TESTING

It is a level of software testing where individual units/components are combined and tested as a group. The purpose of this level of testing is to expose faults in the interaction between integrated units. Test drivers and test stubs are used to assist in Integration Testing. Here we tested every

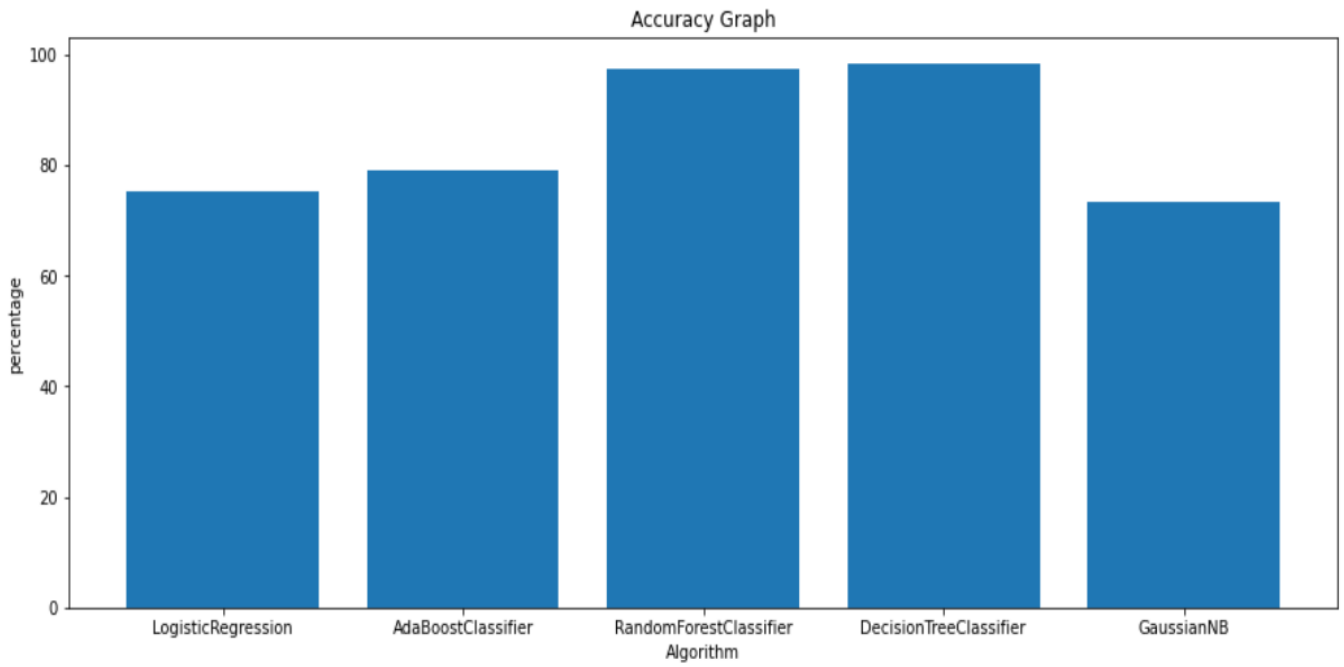
modules that are used in program , we can rectify it

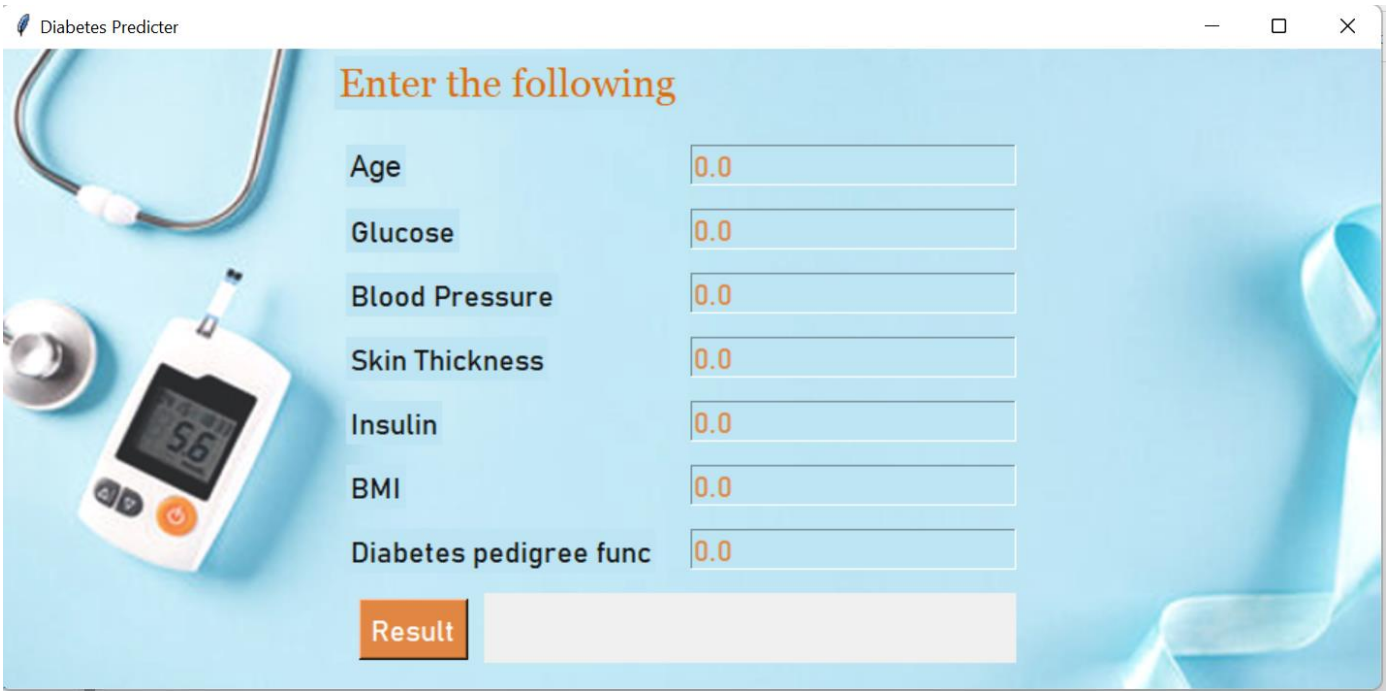
6.3 SYSTEM TESTING

It is a level of testing that validates the complete and fully integrated software product. The purpose of a system test is to evaluate the end-to-end system specifications. Usually, the software is only one element of a larger computer-based system. we tested our entire project thoroughly.

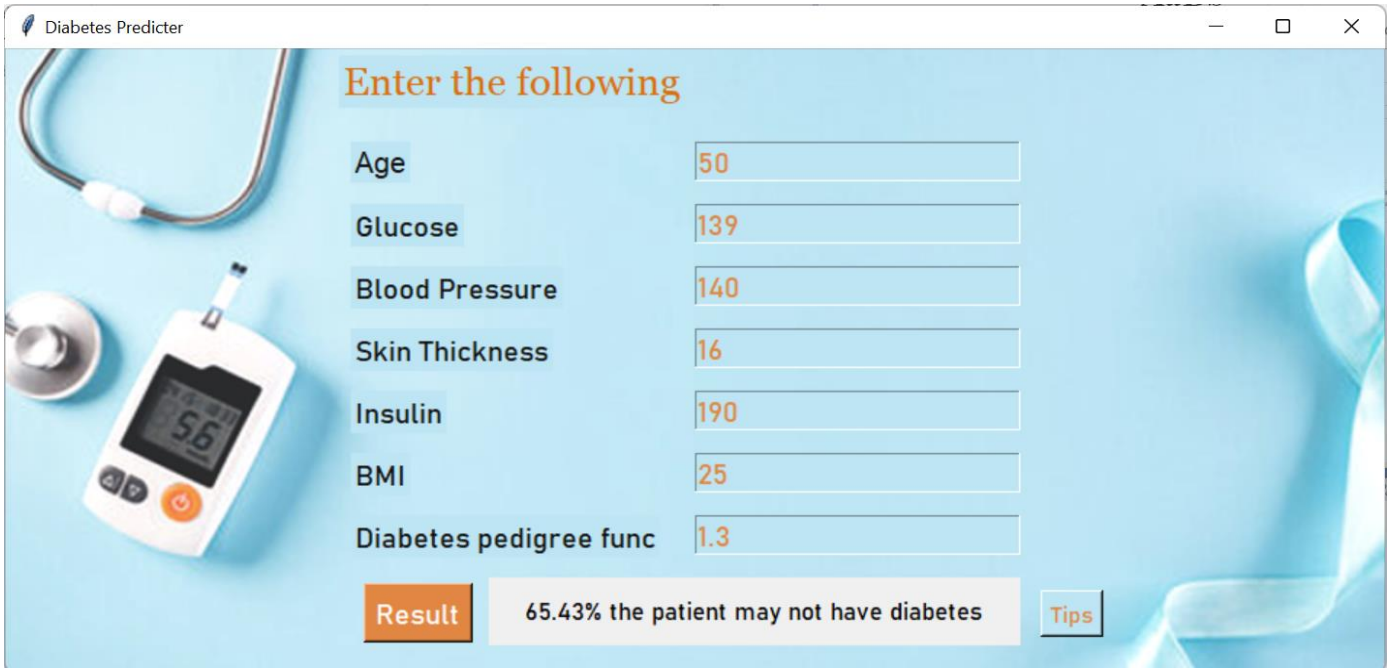
CHAPTER 7

RESULT

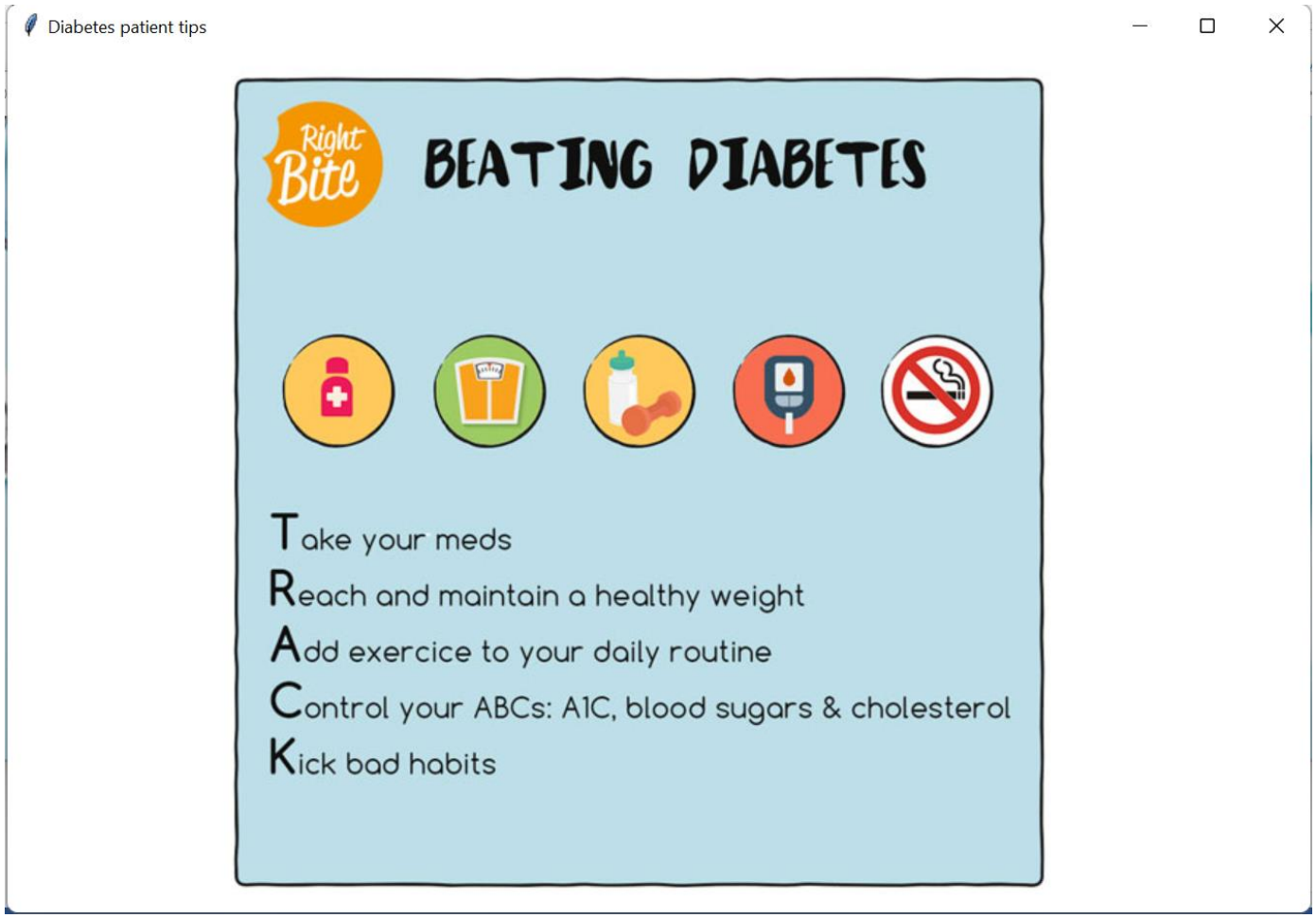





Diabetes Predicter
Enter the following
Age
12
Glucose
250
Blood Pressure
138
Skin Thickness
14
Insulin
150
BMI
24
Diabetes pedigree func
1.5
Result
90.62% the patient may have diabetes
Tips


Diabetes Predicter
Enter the following
Age
50
Glucose
139
Blood Pressure
140
Skin Thickness
16
Insulin
190
BMI
25
Diabetes pedigree func
1.3
Result
65.43% the patient may not have diabetes
Tips


Diabetes Predicter
Enter the following
Age
34
Glucose
250
Blood Pressure
175
Skin Thickness
12
Insulin
281
BMI
27
Diabetes pedigree func
1.9
Result
94.33% the patient may have diabetes
Tips


Diabetes patient tips
Right Bite
BEATING DIABETES
Take your meds
Reach and maintain a healthy weight
Add exercice to your daily routine
Control your ABCs: A1C, blood sugars & cholesterol
Kick bad habits

CHAPTER 8

Conclusion

The prediction of diabetes is one the of great importance in today scenario, and concerning with its severe complications. Due to the biggest reason for the death in worldwide is diabetes. The System model is mainly focus to identification of diabetes using some of the parameters. System is useful to physicians to predict the diabetes in initial dais. So, that conventional treatments and solutions may be given to the patients. System used some of the techniques like ML for the prediction, so that to get the more precise results. There have been fortune of investigation on the diabetes imprint. Building diabetes disease prediction system is useful for hospitals and doctors. System predicts disease at early stages, so doctors can treat patients in a better way. Proposed model is the real time application in which is meant for multiple hospitals and predicts disease in less time. As we use machine learning algorithms for disease prediction, we will get more accurate and efficient results.

Future Scope

Proposed system uses “DECISION TREE algorithm” to find the diabetes disease, in data science we have many algorithms for classification such as Naive Bayes, SVM, KNN , ID3 etc… in future we can add more algorithms to find outputs and algorithms can be compared to find the efficient algorithm. We can add visitor query module, where visitors can post queries to administrator and admin can send reply to those queries. We can add treatment module, where doctors upload treatment details for patients and patient can view those treatment details.

CHAPTER 9

REFERENCES

1. Perveen, S., Shahbaz, M., Saba, T., Keshavjee, K., Rehman, A., & Guergachi, A. (2020). Handling Irregularly Sampled Longitudinal Data and Prognostic Modeling of Diabetes Using Machine Learning Technique. IEEE Access, 8, 21875-21885.

2. Hasan, Md Kamrul, et al. "Diabetes Prediction Using Ensembling of Different Machine Learning Classifiers." IEEE Access 8 (2020): 76516-76531.

3.JACOB, SHON MATHEW, KUMUDHA RAIMOND, and DEEPA KANMANI. "Associated Machine Learning Techniques based On Diabetes Based Predictions." 2019 International Conference on Intelligent Computing and Control Systems (ICCS). IEEE, 2019.

4. VijiyaKumar, K., et al. "Random Forest Algorithm for the Prediction of Diabetes." 2019 IEEE International Conference on System, Computation, Automation and Networking (ICSCAN).

IEEE, 2019.

5. Syed, Rukhsar, Rajeev Kumar Gupta, and Nikhlesh Pathik. "An Advance Tree Adaptive Data Classification for the Diabetes Disease Prediction." 2018 International Conference on Recent Innovations in Electrical, Electronics & Communication Engineering (ICRIEECE). IEEE, 2018.

6.Warsi, Gulam Gaus, Sonia Saini, and Kumar Khatri. "Ensemble Learning on Diabetes Data Set and Early Diabetes Prediction." 2019 International Conference on Computing, Power and Communication Technologies (GUCON). IEEE, 2019.

7.Dutta, Debadri, Debpriyo Paul, and Parthajeet Ghosh. "Analysing feature importances for diabetes prediction using machine learning." 2018 IEEE 9th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON). IEEE, 2018.