NITTE MEENAKSHI INSTITUTE OF TECHNOLOGY

AN AUTONOMOUS INSTITUTION, AFFILIATED TO VISVESVARAYA TECHNOLOGICAL UNIVERSITY,
BELGAUM, APPROVED BY AICTE & GOVT.OF KARNATAKA

KNOWLEDGE * CHARACTER * UNITY

INTERNSHIP REPORT

on

Predictive Model for Diabetes Prediction

Submitted in partial fulfilment of the requirement for the award of Degree of

Bachelor of Engineering
in

Information Science and Engineering

Submitted by:

Palak Dattatraya Kota 11NT22IS127-T

Internship Programme Organized and Conducted by,

Department of Information Science and Engineering, NMIT

NITTE
EDUCATION TRUST

Department of Information Science and Engineering
(Accredited by NBA Tier-1)

2022-2023

AN AUTONOMOUS INSTITUTION, AFFILIATED TO VISVESVARAYA TECHNOLOGICAL UNIVERSITY, BELGAUM

Department of Information Science and Engineering

(Accredited by NBA Tier-1)

KNOWLEDGE * CHARACTER * UNITY

CERTIFICATE

This is to certify that the Internship Report on **“Predictive Model for Diabetes Prediction”** is an authentic work carried out by **Palak Dattatraya Kota (11NT22IS127-T)** Bonafide student of Nitte Meenakshi Institute of Technology, Bangalore, in partial fulfilment for the award of the degree of Bachelor of Engineering in Information Science and Engineering of Visvesvaraya Technological University, Belagavi during the academic year 2022-2023**.** It is certified that all corrections and suggestions indicated during the internal assessment has been incorporated in the report.

Signature of Internal Reviewer

Name:

Designation:

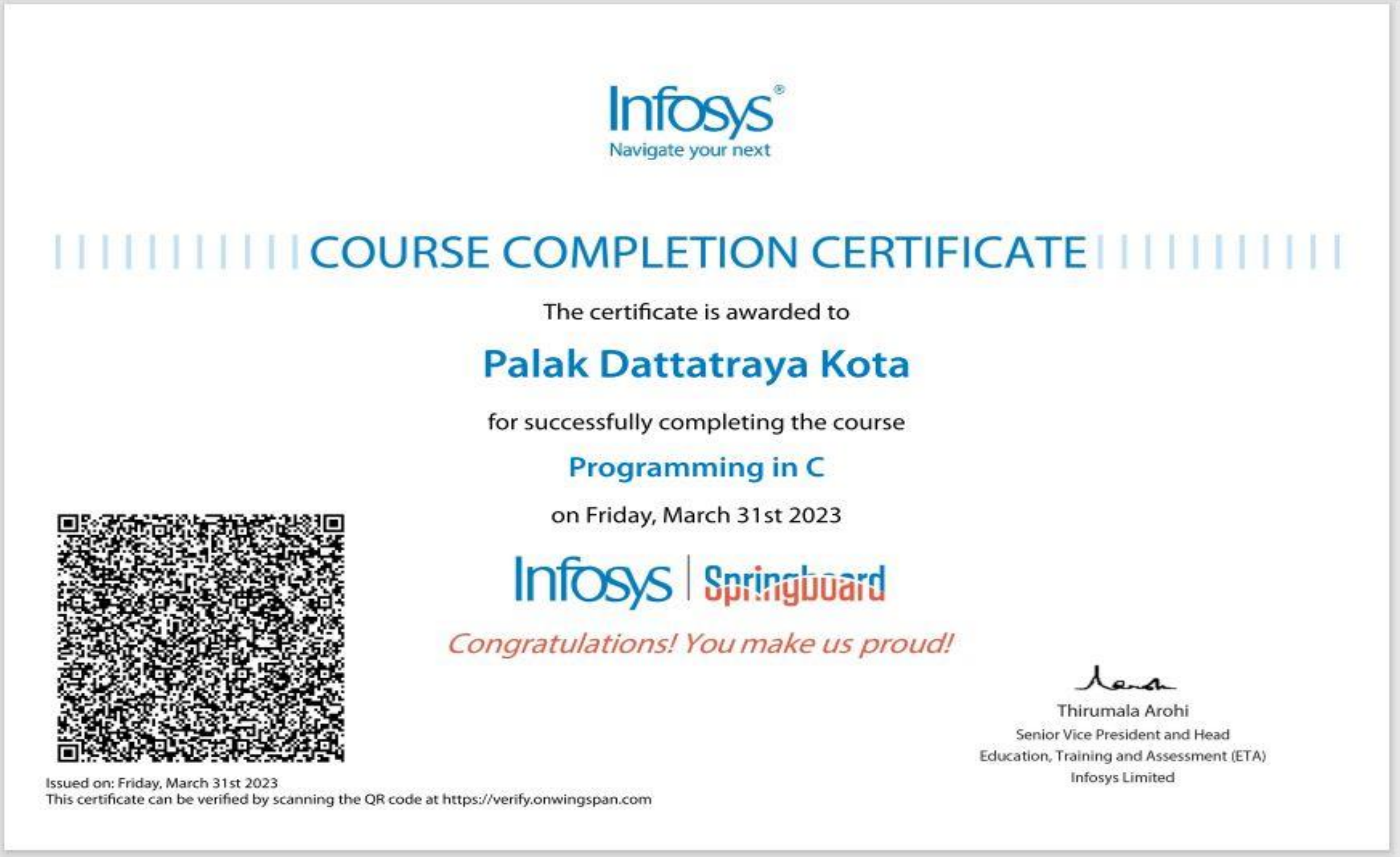
Signature of HoD

Dr. Mohan S G

Professor and HoD, ISE

Signature of External Reviewer

Name:



Infosys®
Navigate your next

COURSE COMPLETION CERTIFICATE

The certificate is awarded to

Palak Dattatraya Kota

for successfully completing the course

Programming in C

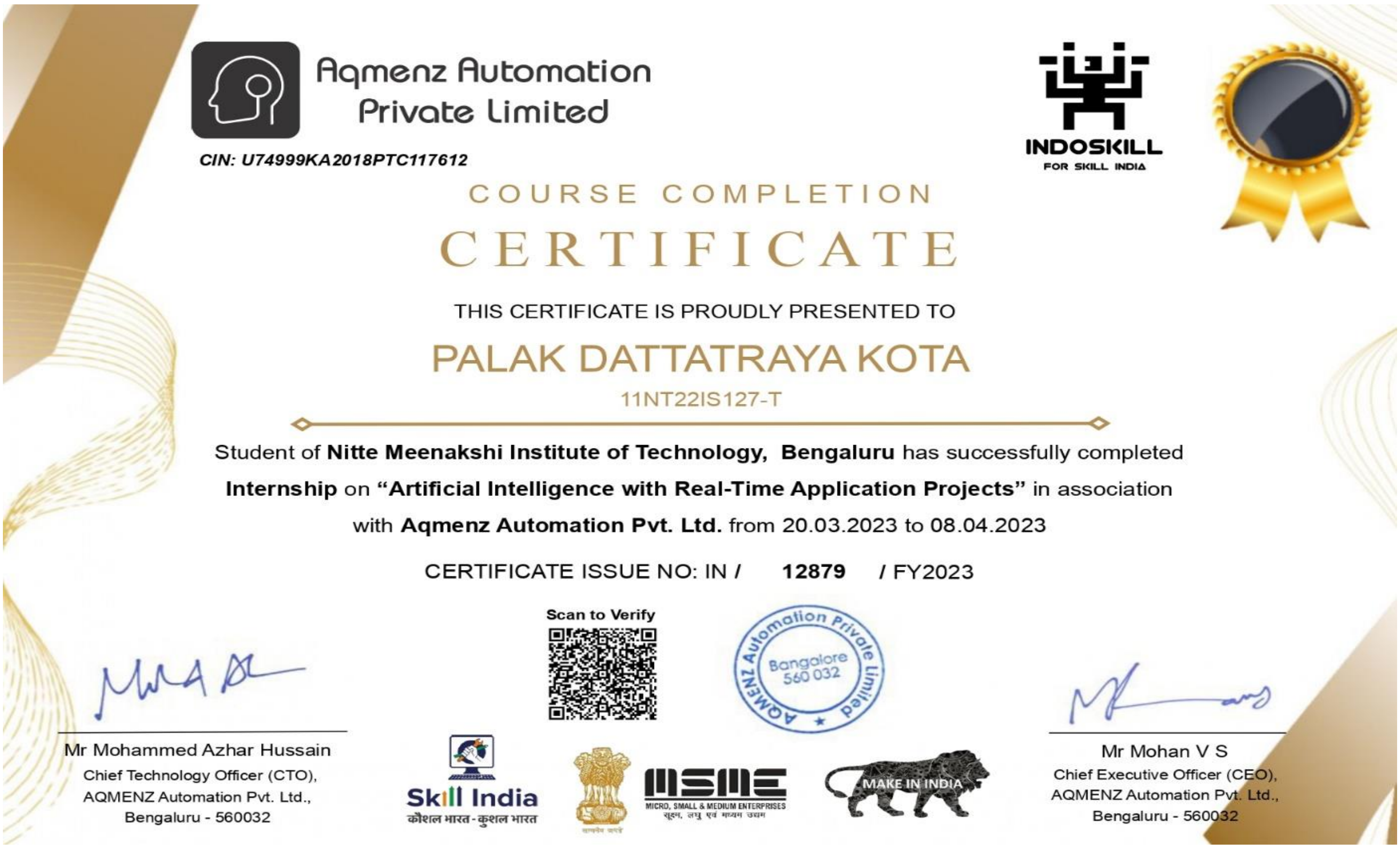
on Friday, March 31st 2023

Infosys | Springboard

Congratulations! You make us proud!

Thirumala Arohi
Senior Vice President and Head
Education, Training and Assessment (ETA)
Infosys Limited

Issued on: Friday, March 31st 2023
This certificate can be verified by scanning the QR code at https://verify.onwingspan.com



Aqmenz Automation
Private Limited

CIN: U74999KA2018PTC117612

INDOSKILL
FOR SKILL INDIA

COURSE COMPLETION

CERTIFICATE

THIS CERTIFICATE IS PROUDLY PRESENTED TO

PALAK DATTATRAYA KOTA

11NT22IS127-T

Student of **Nitte Meenakshi Institute of Technology, Bengaluru** has successfully completed **Internship** on **"Artificial Intelligence with Real-Time Application Projects"** in association with **Aqmenz Automation Pvt. Ltd.** from 20.03.2023 to 08.04.2023

CERTIFICATE ISSUE NO: IN / **12879** / FY2023

Scan to Verify

Aqmenz Automation Private Limited
Bangalore
560 032

Mr Mohammed Azhar Hussain
Chief Technology Officer (CTO),
AQMENZ Automation Pvt. Ltd.,
Bengaluru - 560032

Skill India
कौशल भारत - कुशल भारत

MSME
MICRO, SMALL & MEDIUM ENTERPRISES
सूक्ष्म, लघु एवं मध्यम उद्यम

MAKE IN INDIA

Mr Mohan V S
Chief Executive Officer (CEO),
AQMENZ Automation Pvt. Ltd.,
Bengaluru - 560032

Abstract

Machine learning is an emerging scientific field in data science dealing with the ways in which machines learn from experience. 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. This project aims to predict diabetes using Logistic Regression. Practical implementation of logistic regression using various functions of python programming language.

Acknowledgement

The satisfaction and euphoria that accompany the successful completion of any task would be incomplete without the mention of the people who made it possible, whose constant guidance and encouragement crowned our effort with success. I express my sincere gratitude to our Principal, **Dr. H. C. Nagaraj**, Nitte Meenakshi Institute of Technology, for providing facilities.

We wish to thank our HoD**, Dr. Mohan S. G.,** for the excellent environment created to further educational growth in our college. We also thank him for the invaluable guidance provided, which has helped in the creation of a quality internship project.

I hereby like to thank our Department of Information Science & Engineering for their periodic inspection, organizing, and time-to-time evaluation of the internship and for help to in bringing a project to the final form.

Thanks to our Department Internship coordinator. We also thank all our friends, teaching/trainers, and non-teaching staff at NMIT, Bangalore, for all the direct and indirect help provided in the completion of the internship project.

Table of Contents

Chapter 1

INTRODUCTION

Diabetes is a chronic disease with the potential to cause a worldwide health care crisis. According to International Diabetes Federation 382 million people are living with diabetes across the whole world. By 2035, this will be doubled as 592 million. Diabetes is a disease caused due to the increase level of blood glucose. This high blood glucose produces the symptoms of frequent urination, increased thirst, and increased hunger. Diabetes is a one of the leading causes of blindness, kidney failure, amputations, heart failure and stroke. Type 1 and type 2 diabetes are the most common forms of the disease, but there are also other kinds, such as gestational diabetes, which occurs during pregnancy, as well as other forms. Machine learning is an emerging scientific field in data science dealing with the ways in which machines learn from experience. 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. This project aims to predict diabetes using Logistic Regression. • The software used in designing is Google Colab.

Chapter 2

TOOLS EXPOSED

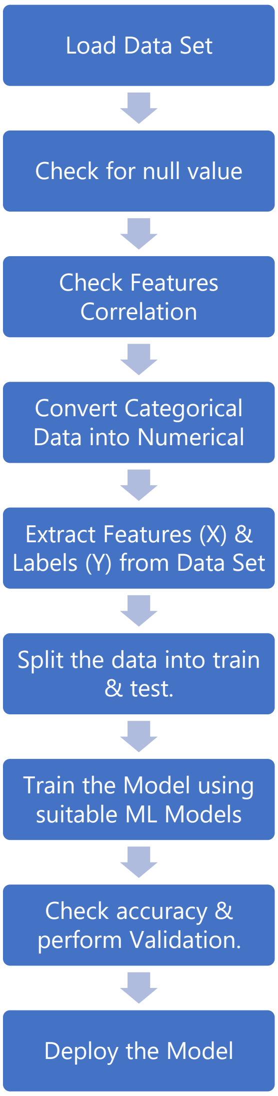
3.1 Google colab

Colaboratory or colab for short, is a product from Google research. 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. More technically colab is a hosted jupyter notebook service that requires no setup to use, while providing access free of charge to computing resources including GPUs. Colab resources are not guaranteed and not unlimited, and the usage limits sometimes fluctuate. This is necessary for colab to be able to provide resources free charge. Resources in colab are prioritized for interactive use cases. We prohibit actions associated with bulk compute, actions that negatively impact others as well as actions associated with bypassing the policies. Jupyter is an open-source project on which the colab is based. Colab allows you to use and share jupyter notebooks with others without having to download, install or run anything. You can search colab notes using google drive. Clicking on the colab logo at the top left of the notebook view will show all notebooks in drive. You can also search for notebooks that you have opened recently by clicking on file and then open notebook. Google drive operations can time out when the number of folders or subfolders in a folder grows too large. If thousands of items are directly contained in the top level “My drive” folder then mounting the drive will likely time out. Repeated attempts may eventually succeed as failed attempts cache partial state locally before timing out. Colab is able to provide resources free of cost in part by having dynamic usage limits that sometimes fluctuate this means that overall usage limits as well as idle timeout periods, maximum VM lifetime, GPU types available and other factors vary over time. Colab does not publish these limits in parts because they can vary quickly. This is necessary for colab to be able to provide access these resources free of charge. Colab works with most of the major browsers and is most thoroughly tested with the latest versions of Chrome, Firefox and Safari.

Chapter 3

TASKS PERFORMED: PREDICTIVE MODEL FOR DIABETES

4.1 General steps in Logistic regression:

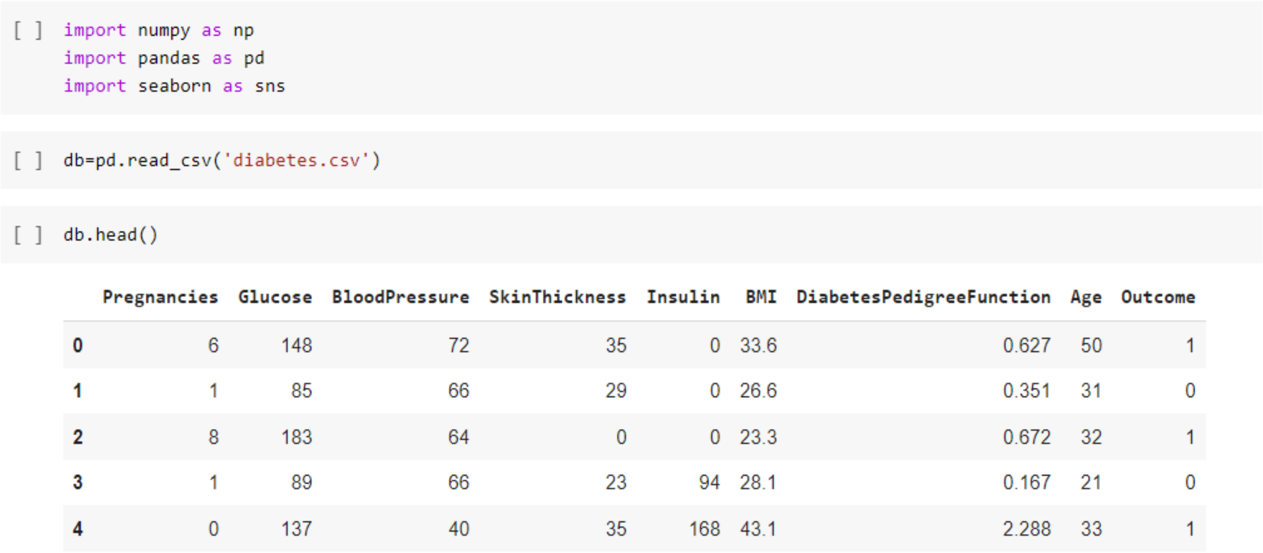

Load Data Set
Check for null value
Check Features Correlation
Convert Categorical Data into Numerical
Extract Features (X) & Labels (Y) from Data Set
Split the data into train & test.
Train the Model using suitable ML Models
Check accuracy & perform Validation.
Deploy the Model

The general steps are about loading the data set in the application where I am trying to build the model. Then I need to check for the empty values in the data set. If there are no null values, then I can proceed to the next step or else I need to clean the data for further building of the project. Once done with the above steps, any data which is categorical is changed to numerical. But in diabetes prediction, the data is mostly numerical. If not, we have to change it otherwise.

Chapter 4

CODE SCREENSHOTS

4.1 Load the Data set



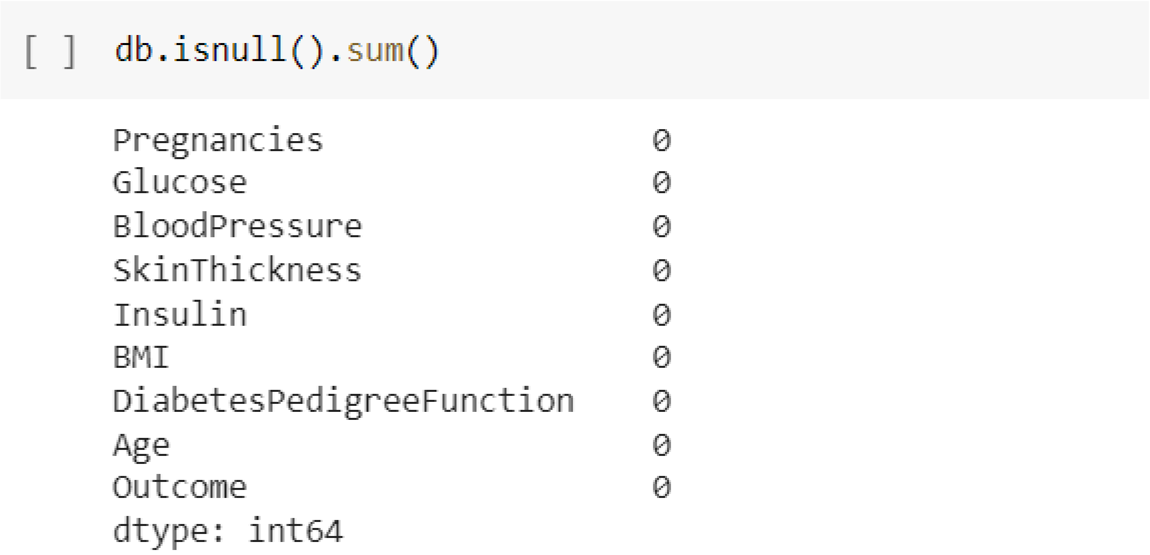
```
[ ] import numpy as np
    import pandas as pd
    import seaborn as sns
```

```
[ ] db=pd.read_csv('diabetes.csv')
```

```
[ ] db.head()
```

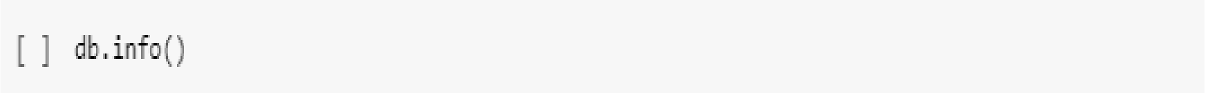
	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	6	148	72	35	0	33.6	0.627	50	1
1	1	85	66	29	0	26.6	0.351	31	0
2	8	183	64	0	0	23.3	0.672	32	1
3	1	89	66	23	94	28.1	0.167	21	0
4	0	137	40	35	168	43.1	2.288	33	1

4.2 Check for Null Values

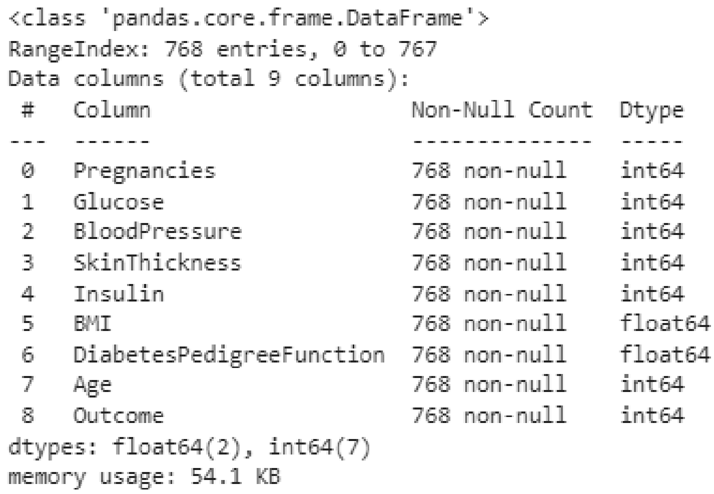


```
[ ] db.isnull().sum()

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

4.3 Check Features Corelation

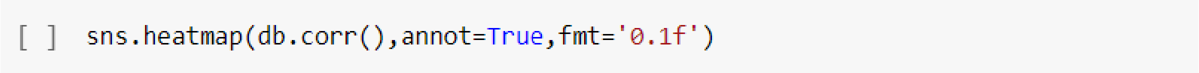


```
[ ] db.info()
```

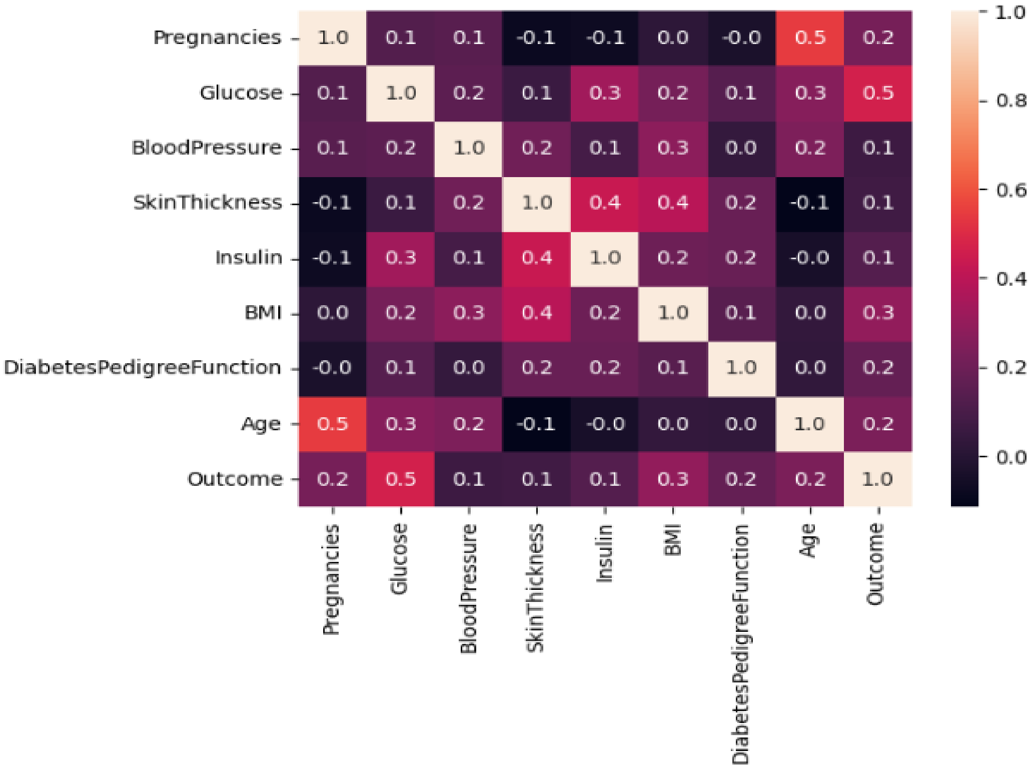


```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 768 entries, 0 to 767
Data columns (total 9 columns):
 #   Column                    Non-Null Count  Dtype  
---  ------                    --------------  -----  
 0   Pregnancies               768 non-null    int64  
 1   Glucose                   768 non-null    int64  
 2   BloodPressure             768 non-null    int64  
 3   SkinThickness             768 non-null    int64  
 4   Insulin                   768 non-null    int64  
 5   BMI                       768 non-null    float64
 6   DiabetesPedigreeFunction  768 non-null    float64
 7   Age                       768 non-null    int64  
 8   Outcome                   768 non-null    int64  
dtypes: float64(2), int64(7)
memory usage: 54.1 KB
```

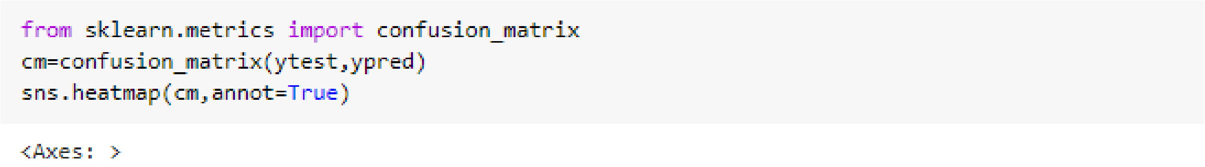
HEATMAP



```
[ ] sns.heatmap(db.corr(),annot=True,fmt='0.1f')
```

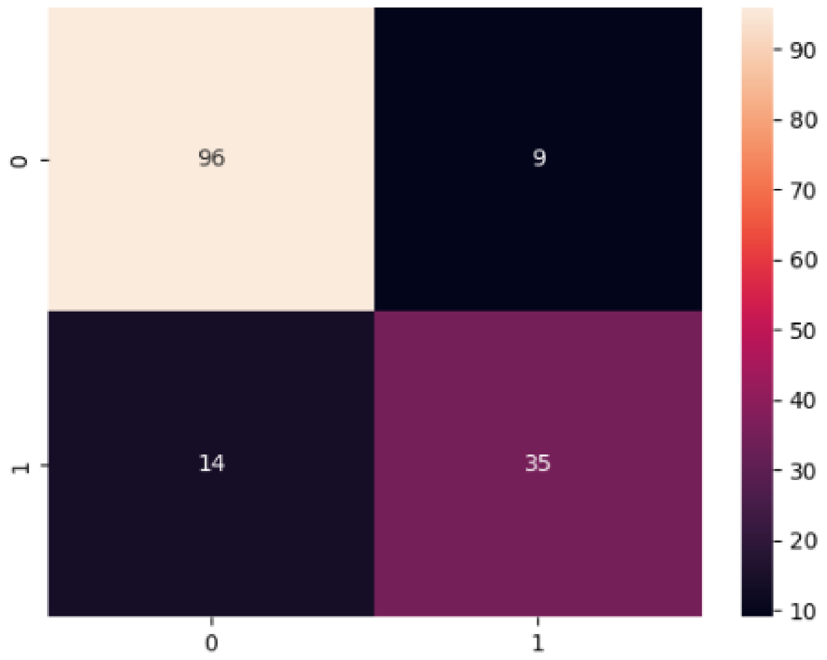


HEATMAP – CONFUSION MATRIX

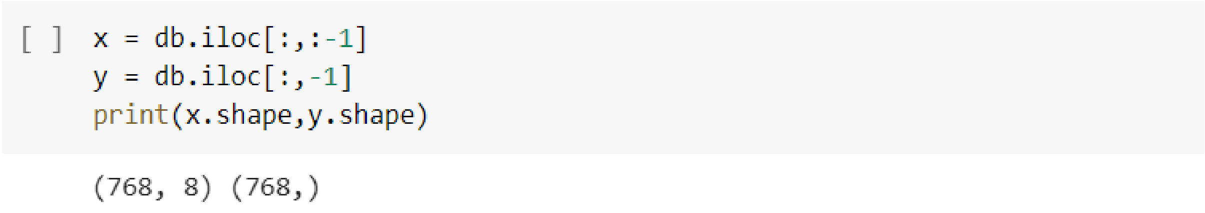


```
from sklearn.metrics import confusion_matrix
cm=confusion_matrix(ytest,ypred)
sns.heatmap(cm,annot=True)
```

```
<Axes: >
```

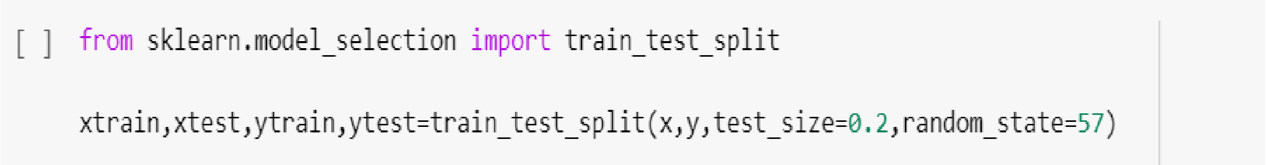


4.4 Extract Features (X) & Labels (Y) from Data Set



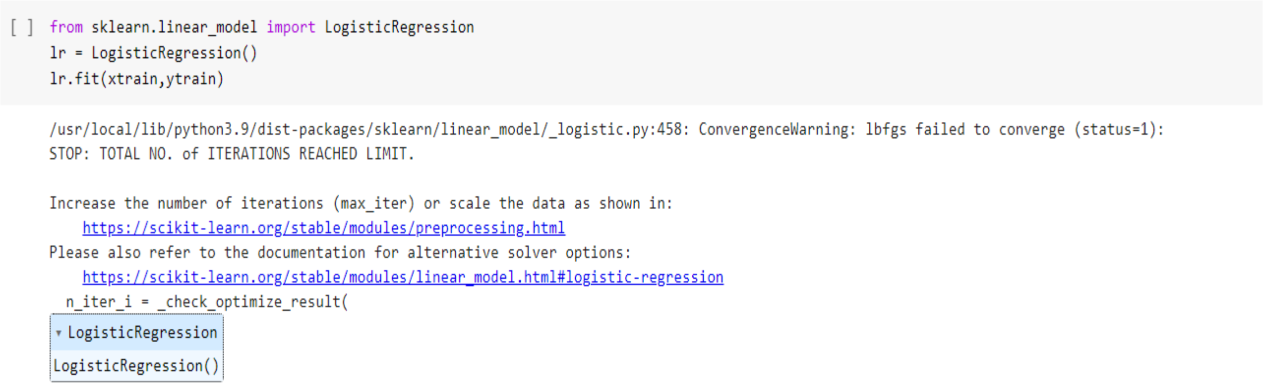
```
[ ]  x = db.iloc[:,:-1]
     y = db.iloc[:,-1]
     print(x.shape,y.shape)

     (768, 8) (768,)
```

4.5 Split the data into train & test



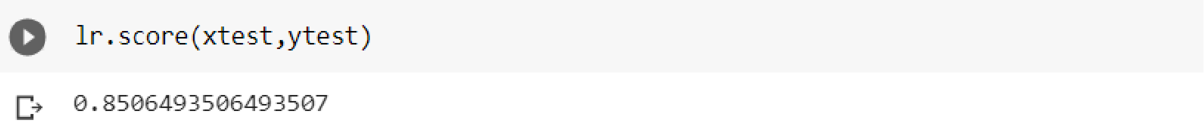
```
[ ]  from sklearn.model_selection import train_test_split

     xtrain,xtest,ytrain,ytest=train_test_split(x,y,test_size=0.2,random_state=57)
```

4.6 Train the model using ML models



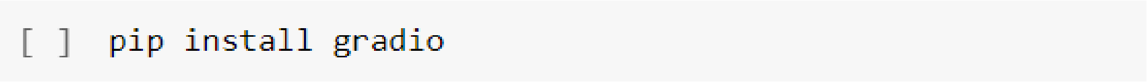
```
[ ] from sklearn.linear_model import LogisticRegression
    lr = LogisticRegression()
    lr.fit(xtrain,ytrain)

    /usr/local/lib/python3.9/dist-packages/sklearn/linear_model/_logistic.py:458: ConvergenceWarning: lbfgs failed to converge (status=1):
    STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

    Increase the number of iterations (max_iter) or scale the data as shown in:
        https://scikit-learn.org/stable/modules/preprocessing.html
    Please also refer to the documentation for alternative solver options:
        https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
      n_iter_i = _check_optimize_result(
    ▾ LogisticRegression
    LogisticRegression()
```

4.7 Check Accuracy Score of ML Model

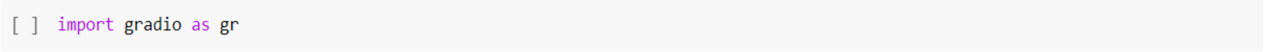


```
lr.score(xtest,ytest)

0.8506493506493507
```

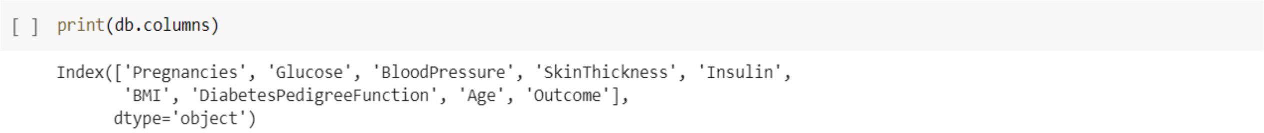
4.8 Deploy the Model



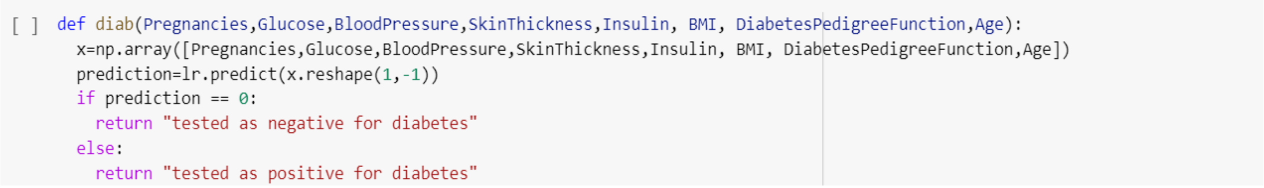
```
[ ]  pip install gradio
```



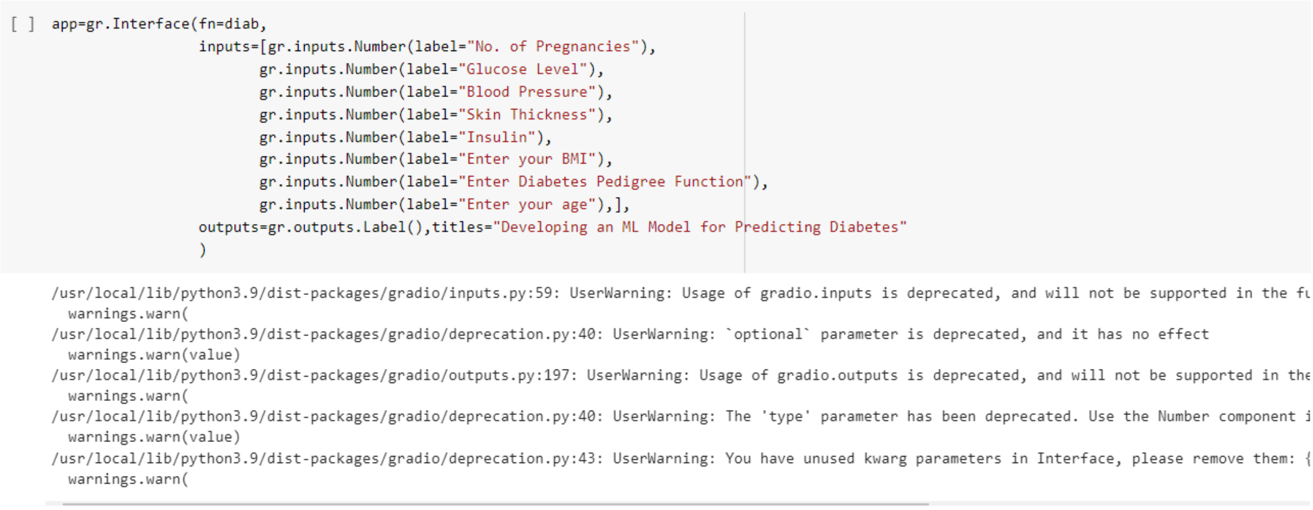
```
[ ] import gradio as gr
```



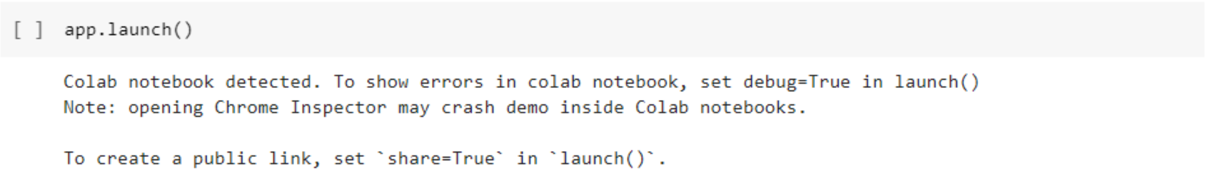
```
[ ] print(db.columns)

    Index(['Pregnancies', 'Glucose', 'BloodPressure', 'SkinThickness', 'Insulin',
           'BMI', 'DiabetesPedigreeFunction', 'Age', 'Outcome'],
          dtype='object')
```



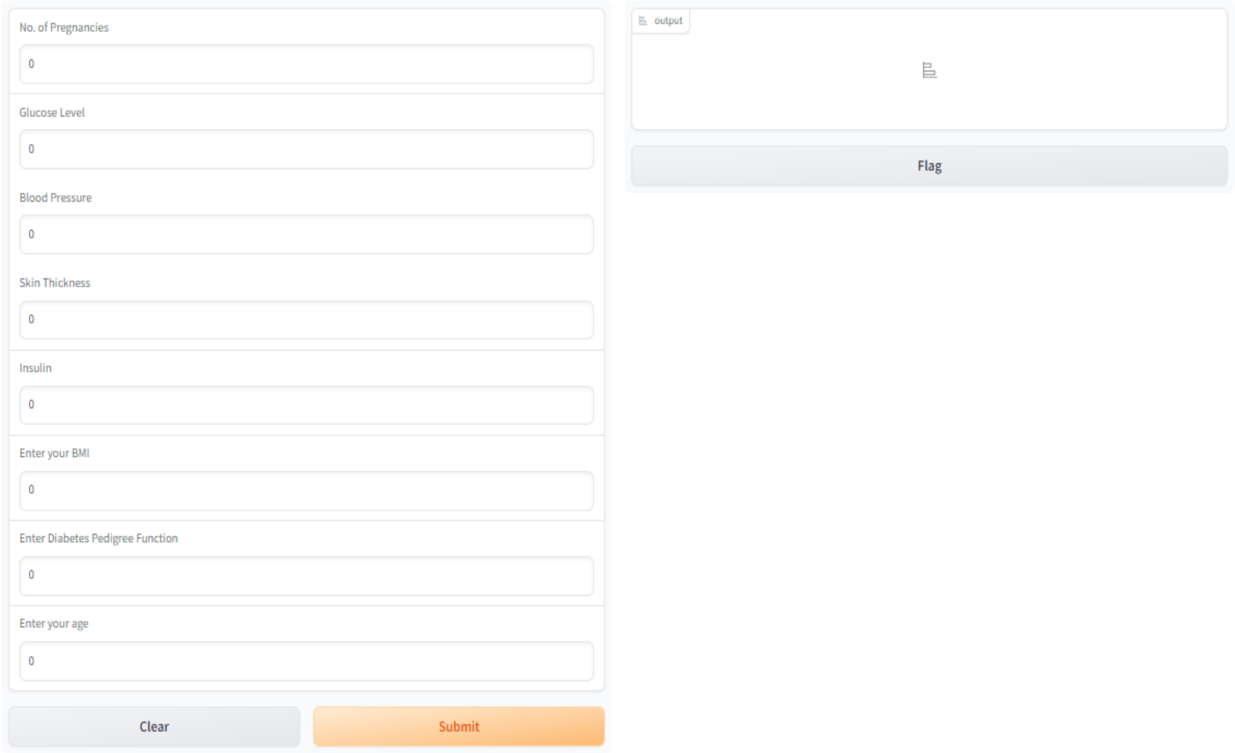
```
[ ] def diab(Pregnancies,Glucose,BloodPressure,SkinThickness,Insulin, BMI, DiabetesPedigreeFunction,Age):
      x=np.array([Pregnancies,Glucose,BloodPressure,SkinThickness,Insulin, BMI, DiabetesPedigreeFunction,Age])
      prediction=lr.predict(x.reshape(1,-1))
      if prediction == 0:
        return "tested as negative for diabetes"
      else:
        return "tested as positive for diabetes"
```



```
[ ] app=gr.Interface(fn=diab,
                     inputs=[gr.inputs.Number(label="No. of Pregnancies"),
                             gr.inputs.Number(label="Glucose Level"),
                             gr.inputs.Number(label="Blood Pressure"),
                             gr.inputs.Number(label="Skin Thickness"),
                             gr.inputs.Number(label="Insulin"),
                             gr.inputs.Number(label="Enter your BMI"),
                             gr.inputs.Number(label="Enter Diabetes Pedigree Function"),
                             gr.inputs.Number(label="Enter your age"),],
                     outputs=gr.outputs.Label(),titles="Developing an ML Model for Predicting Diabetes"
                     )

    /usr/local/lib/python3.9/dist-packages/gradio/inputs.py:59: UserWarning: Usage of gradio.inputs is deprecated, and will not be supported in the fu
      warnings.warn(
    /usr/local/lib/python3.9/dist-packages/gradio/deprecation.py:40: UserWarning: `optional` parameter is deprecated, and it has no effect
      warnings.warn(value)
    /usr/local/lib/python3.9/dist-packages/gradio/outputs.py:197: UserWarning: Usage of gradio.outputs is deprecated, and will not be supported in the
      warnings.warn(
    /usr/local/lib/python3.9/dist-packages/gradio/deprecation.py:40: UserWarning: The 'type' parameter has been deprecated. Use the Number component i
      warnings.warn(value)
    /usr/local/lib/python3.9/dist-packages/gradio/deprecation.py:43: UserWarning: You have unused kwarg parameters in Interface, please remove them: {
      warnings.warn(
```



```
[ ]  app.launch()

     Colab notebook detected. To show errors in colab notebook, set debug=True in launch()
     Note: opening Chrome Inspector may crash demo inside Colab notebooks.

     To create a public link, set `share=True` in `launch()`.
```



No. of Pregnancies
0

Glucose Level
0

Blood Pressure
0

Skin Thickness
0

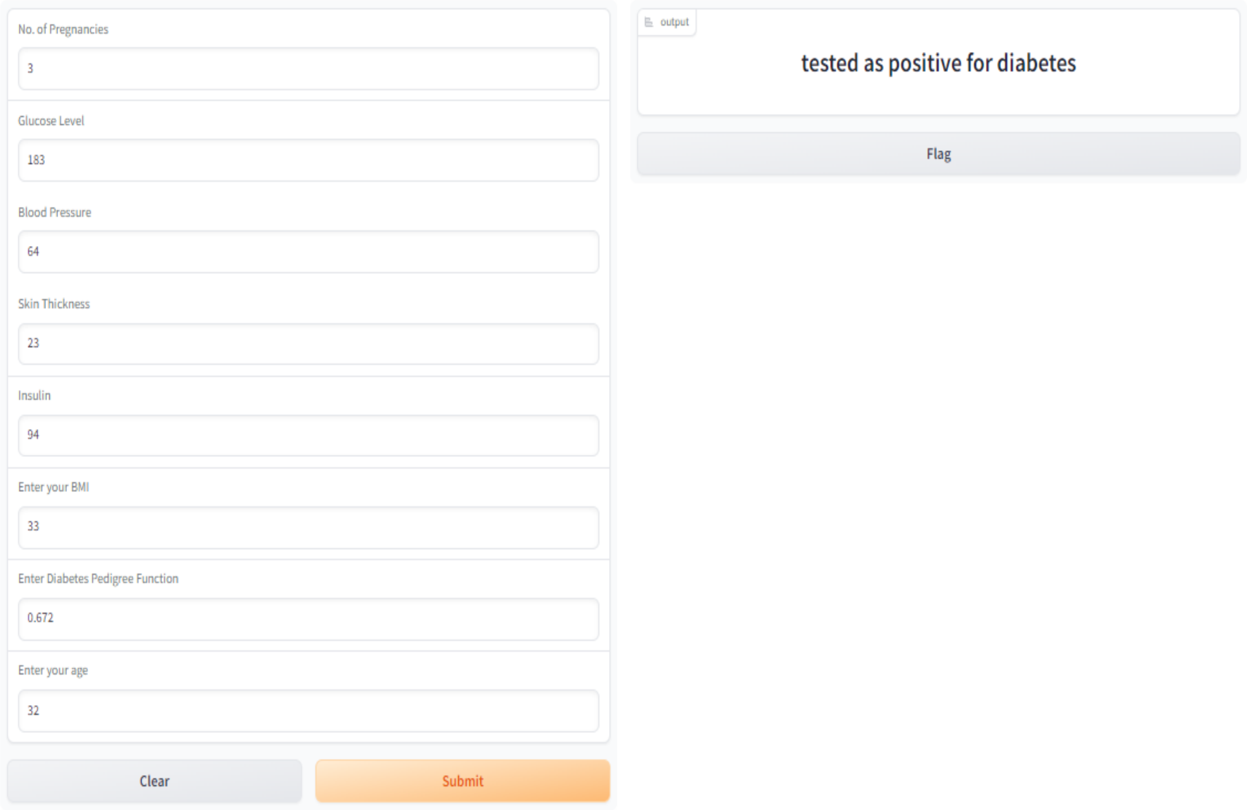
Insulin
0

Enter your BMI
0

Enter Diabetes Pedigree Function
0

Enter your age
0

Example:



No. of Pregnancies
3

Glucose Level
183

Blood Pressure
64

Skin Thickness
23

Insulin
94

Enter your BMI
33

Enter Diabetes Pedigree Function
0.672

Enter your age
32

output
tested as positive for diabetes

Chapter 5

SOFTWARE TECHNOLOGY

5.1 Introduction to Microcontrollers:

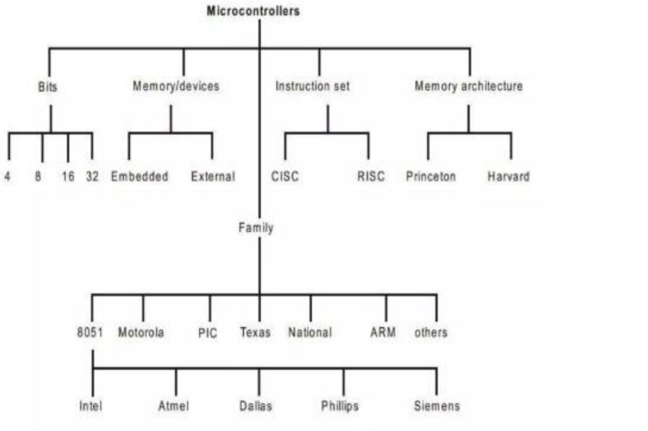
A microcontroller is a compact integrated circuit designed to govern a specific operation in an embedded system. It is used for specific purpose application.

A microcontroller consists of:

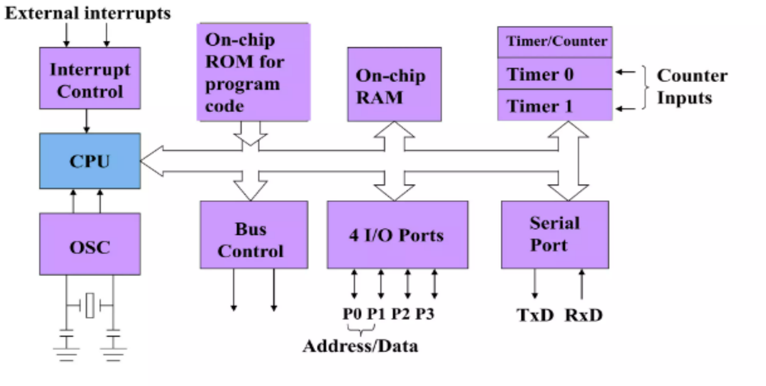
a) CPU – Central Processing Unit
b) I/O – Input/Output
c) Bus – Address bus & Data bus
d) Memory – RAM & ROM
e) Timer
f) Interrupt
g) Serial Port

Microprocessor - an integrated circuit that contains all the functions of a central processing unit of a computer. Ex: Intel’s 8085

Types of Microcontrollers

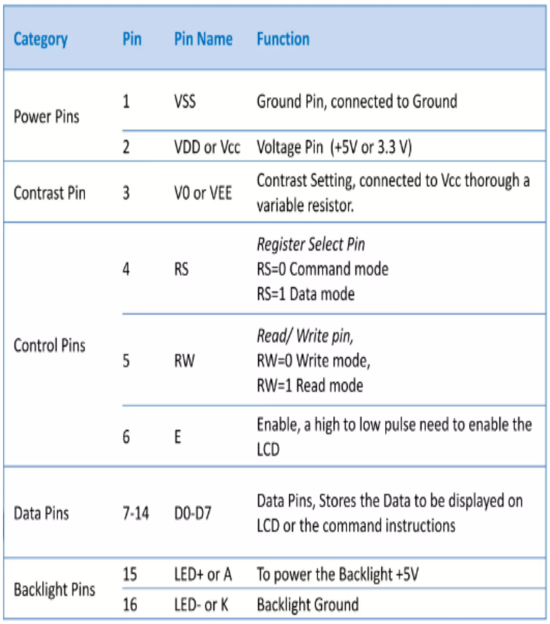


Block Diagram of Microcontrollers



Interfacing of 8051 Microcontroller with LCD

Pin Description of LCD

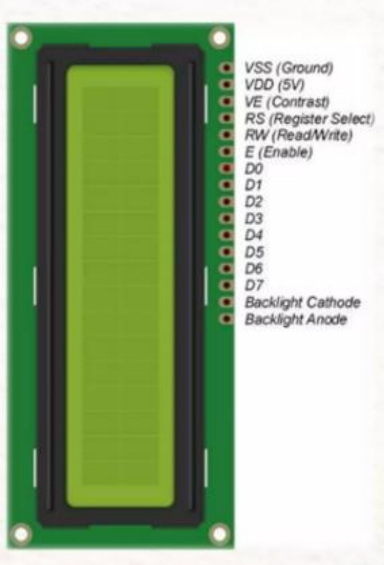


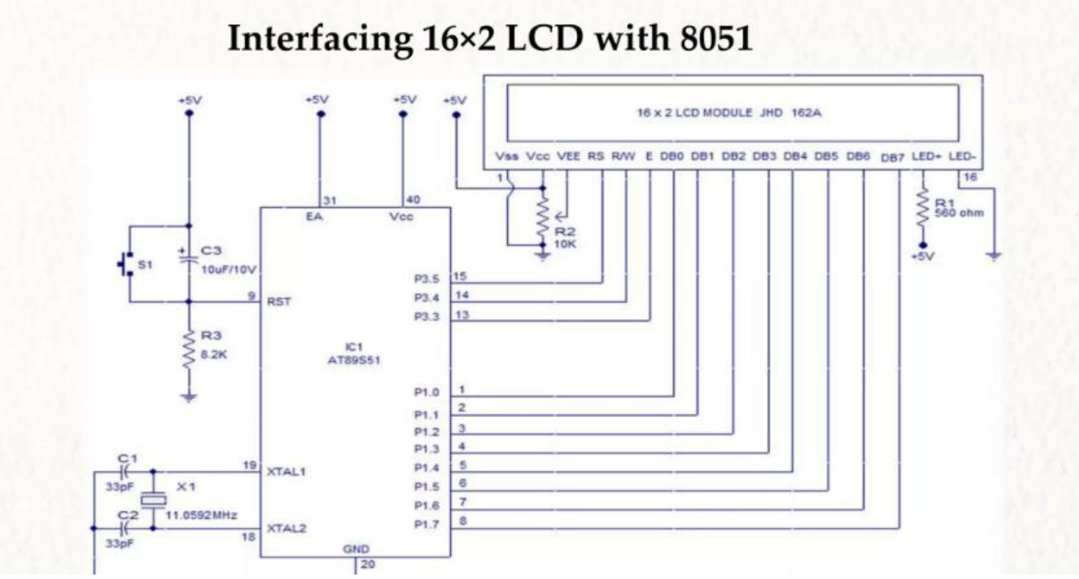
Category	Pin	Pin Name	Function
Power Pins	1	VSS	Ground Pin, connected to Ground
	2	VDD or Vcc	Voltage Pin (+5V or 3.3 V)
Contrast Pin	3	V0 or VEE	Contrast Setting, connected to Vcc thorough a variable resistor.

Control Pins	4	RS	*Register Select Pin*
			RS=0 Command mode
			RS=1 Data mode

	5	RW	*Read/ Write pin,*
			RW=0 Write mode,
			RW=1 Read mode

	6	E	Enable, a high to low pulse need to enable the LCD
Data Pins	7-14	D0-D7	Data Pins, Stores the Data to be displayed on LCD or the command instructions
Backlight Pins	15	LED+ or A	To power the Backlight +5V
	16	LED- or K	Backlight Ground




Interfacing 16×2 LCD with 8051

Interfacing of 8051 Microcontroller with DC Motor

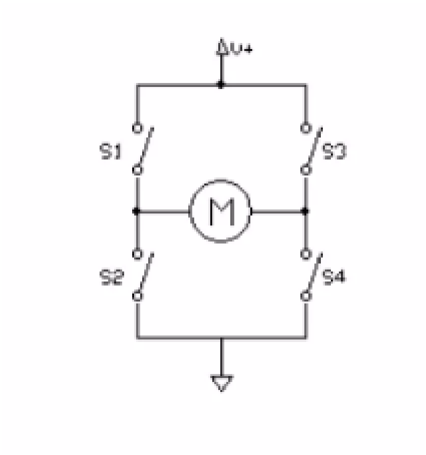
A direct current (DC) motor is another widely used device that translates electrical pulses into mechanical movement. In DC we have only (+) and (-) lead. Connecting them to DC voltage source moves motor in one direction and by reversing polarity motor moves in opposite direction.

Motor can be controlled in two ways:

- Unidirectional Control
- Bidirectional Control

With help of relays or some special chips we change the direction of DC motor rotation.

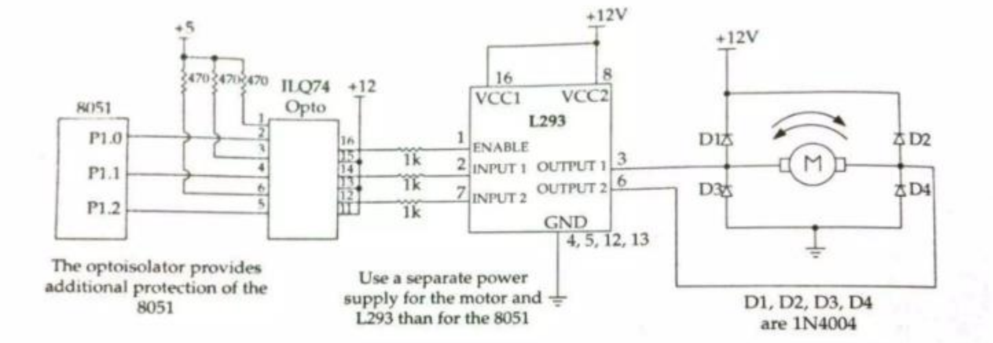
H-Bridge motor control circuits



An H-bridge is a bipolar driver circuit is that often used to control load such as a brush type DC motor. It consists of four switches – Two half-bridges with load connected between them.

The figure shows H- bridge motor control circuit in rest condition

Configuration for Interface 8051 with DC motor using L293



The optoisolator provides additional protection of the 8051

Use a separate power supply for the motor and L293 than for the 8051

D1, D2, D3, D4 are 1N4004

Bidirectional Motor Control Using an L293 Chip

5.2 Introduction to IoT (Internet of Things)

The Internet of Things (IoT) is a system of interrelated computing devices, mechanical and digital machines, objects, animals or people that are provided with unique identifiers and the ability to transfer data over a network without requiring human-to-human or human-to-computer interaction.

Applications of IoT

1. Wearables

Wearable technology is the hallmark of IoT applications and one of the earliest industries to deploy IoT. We have fit bits, heart rate monitors and smartwatches these days. Guardian glucose monitoring device has been developed to help people with diabetes.

2. Health care

It uses controlled environments, leftover data, and volunteers for clinical trials. The Internet of Things improves the device's power**,** precision and availability. IoT focuses on building systems rather than just tools. Here's how the IoT-enabled care device works.

3. Smart Farming

Farmers can minimize waste and increase productivity. The system allows the monitoring of fields with the help of sensors. Farmers can monitor the status of the area.

4. Smart Retail

IoT applications in retail give shoppers a new experience. Customers do not have to stand in long queues as the checkout system can read the tags of the products and deduct the total amount from the customer's payment app with IoT applications' help.

5. Smart Cities

Smart city uses technology to provide services. The smart city includes improving transportation and social services, promoting stability and giving voice to their citizens.

5.3 Programming Languages

a. Basics of Java

Java is a high-level, general-purpose, object-oriented, and secure programming language developed by James Gosling at Sun Microsystems, Inc. in 1991.

Features of Java:

- **Simple:** Java is a simple language because its syntax is simple, clean, and easy to understand.
- **Object-Oriented:** In Java, everything is in the form of the object. It means it has some data and behaviour.
- **Robust:** Java makes an effort to check error at run time and compile time. It uses a strong memory management system called garbage collector. Exception handling and garbage collection features make it strong.
- **Secure:** Java is a secure programming language because it has no explicit pointer and programs runs in the virtual machine.
- **Platform-Independent:** Java provides a guarantee that code writes once and run anywhere.
- **Portable:** Java Byte code can be carried to any platform.
- **High Performance:** Java is an interpreted language.
- **Distributed:** Java also has networking facilities. It is designed for the distributed environment of the internet because it supports TCP/IP protocol.
- **Multi-threaded:** Java also supports multi-threading. It means to handle more than one job a time.

OOPs (Object Oriented Programming System)

Object-oriented programming is a way of solving a complex problem by breaking them into a small sub-problem. An object is a real-world entity. It is easier to develop a program by using an object. In OOPs, we create programs using class and object in a structured manner.

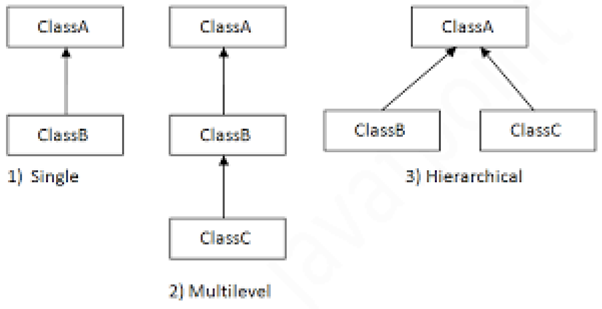
- **Class**: A class is a template or blueprint or prototype that defines data members and methods of an object. An object is the instance of the class. We can define a class by using the class keyword.

- **Object**: An object is a real-world entity that can be identified distinctly. For example, a desk, a circle can be considered as objects. An object has a unique behaviour, identity, and state. Data fields with their current values represent the state of an object (also known as its properties or attributes).

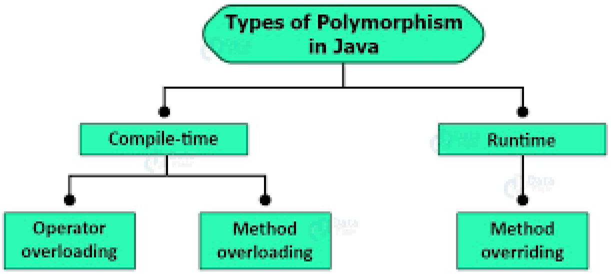
- **Abstraction**: An abstraction is a method of hiding irrelevant information from the user. For example, the driver only knows how to drive a car; there is no need to know how does the car run. We can make a class abstract by using the keyword abstract. In Java, we use abstract class and interface to achieve abstraction.

- **Encapsulation**: An encapsulation is the process of binding data and functions into a single unit. A class is an example of encapsulation. In Java, Java bean is a fully encapsulated class.

- **Inheritance**: Inheritance is the mechanism in which one class acquire all the features of another class. We can achieve inheritance by using the extends keyword. It facilitates the reusability of the code.



- **Polymorphism**: The polymorphism is the ability to appear in many forms. In other words, single action in different ways. For example, a boy in the classroom behaves like a student, in house behaves like a son. There are two types of polymorphism: run time polymorphism and compile-time polymorphism.



b. Basics of Python

Python is a general purpose, dynamic, high-level, and interpreted programming language. It supports Object Oriented programming approach to develop applications. It is simple and easy to learn and provides lots of high-level data structures. It was created in 1991 by Guido Van Rossum.

Python Basics:

Python Built-in Data Types

- int (Integer): They are positive or negative whole numbers with no decimal point.
- float: Decimal numbers comes under float data type.
- String (str)
- Boolean: It has two boolean values True or False.

Type Casting:

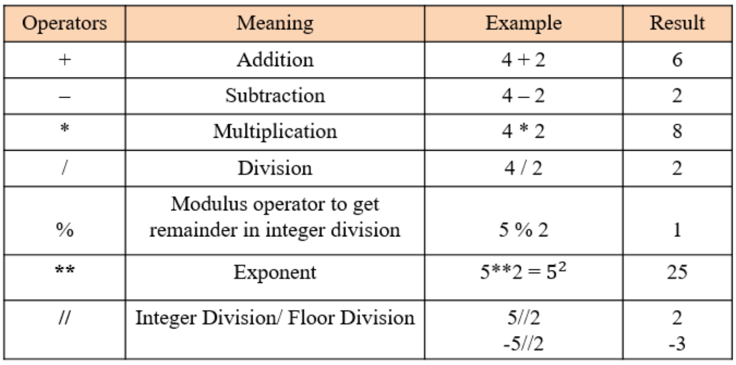
Casting in python is therefore done using constructor functions

● int() - Used to Convert any type of data into int data type

● float() - Used to Convert any type of data into float data type

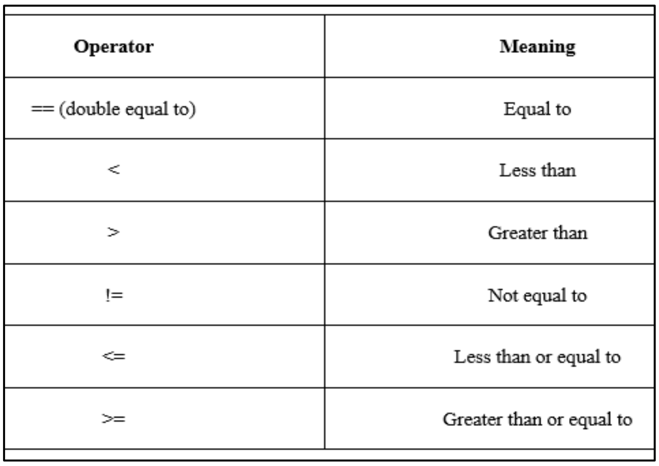
● str() - Used to Convert any type of data into string data type

Arithmetic Operators:



Operators	Meaning	Example	Result
+	Addition	4 + 2	6
–	Subtraction	4 – 2	2
*	Multiplication	4 * 2	8
/	Division	4 / 2	2
%	Modulus operator to get remainder in integer division	5 % 2	1
**	Exponent	5**2 = 5^2	25
//	Integer Division/ Floor Division	5//2 -5//2	2 -3

Conditional Operators in Python:



Operator	Meaning
== (double equal to)	Equal to
<	Less than
>	Greater than
!=	Not equal to
<=	Less than or equal to
>=	Greater than or equal to

Python If-else statements

- if statement
- if-else statement
- elif statement
- nested if statement

Logical Operators:

- and- Returns True if both statements are true
- or- Returns True if one of the statements is true
- not- Reverse the result, returns False if the result is true

Loop Control Statements

- ‘for’ loop: A for loop is used for iterating over a sequence. Using for loop we can execute a set of statements, once for each item in a list, tuple, set etc.
  range() function, used in for loop, returns a sequence of numbers, starting from 0 by default, and increments by 1 (by default), and ends at a specified number.
- while loop:The while is used to repeat statements until the condition is True. And the loop is broken when condition becomes False.
  Syntax: while (<condition>)::
  ```
                  statement-1
                  statement-2
                  ……
  ```

Lists:

Lists are used to store multiple items in a single variable. List items are ordered, changeable, and allow duplicate values.

Ex: Fruits_list = ["apple", "banana", "cherry"]

list1 = ["abc", 34, True, 40, "male"]

Tuple:

A tuple is a collection which is ordered and unchangeable. In Python tuples are written with round brackets.

Ex :thistuple = (“one”, “two”, “HelloWorld”)

print(thistuple)

Output:

('one', 'two', 'HelloWorld‘)

Dictionaries:

Dictionaries are used to store data values in key-value pairs.

syntax:

thisdictionary = {key1:value, key2:value, key3:value . . . }

Ex-:thisdict = {"brand": "Ford", "model": "Mustang", "year": 1964 }

print(thisdict)

<u>Python Libraries:</u>

Python packages-

- Numpy- NumPy or Numerical Python is a general-purpose array processing python package for scientific computing.

importing numpy library:

import numpy as np

Numpy Functions:

- .arange( )
- np.arange(start_no, end_no + 1, step)
- .linspace( )
- np.linspace(start_value,end_value,no_of_items)
- .random.random( )
- np.random.random((shape of array))
- .random.randint( )
- np.random.randint(start_value,end_value,(shape of array))
- .zeros( )
- np.zeros((shape of array),dtype)
- .ones( )
- np.ones((shape of array),dtype)
- .eye( )
- np.eye(No. of rows,dtype)

- Pandas - Pandas is an open-source Python Library. Used for high-performance data manipulation and analysis
- Matplotlib -

  Matplotlib Graphs:

  ➔ Line Plot (Data: Numerical vs Numerical)

  ➔ Scatter Plot (Data: Numerical vs Numerical)

  ➔ Bar Plot (Data: Numerical vs Categorical)

  ➔ Pie Plot (Data: Numerical vs Categorical, Distribution in %)
- Seaborn

  Types of Data Analysis

  1. Univariate Analysis: boxplot(), countplot() ➔ Analysis of one variable at a time.
  2. Bivariate Analysis: barplot(), scatterplot() ➔ Analysis of Two variable at a time.
  3. Multivariate Analysis: Heatmap ➔ Analyzing Multiple Features at a Time
- Scikit learn

5.4 Artificial Intelligence and Machine Learning using Python

Artificial Intelligence

Artificial Intelligence refers to intelligence displayed by machines that simulates human and animal intelligence.

Machine Learning

The capability of Artificial Intelligence systems to learn by extracting patterns from data is known as Machine Learning.

Applications of Machine Learning

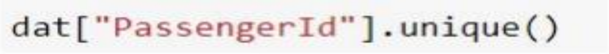
- Image Processing
- Robotics
- Data Mining
- Video Games
- Text Analysis
- Healthcare

Steps in Machine Learning

1. Data Gathering
2. Data Cleaning
3. Data Visualization
4. Data Processing
5. Training & Testing
6. Predictive Model

Data Exploration Techniques

1. Identifying Unique Elements →



```
dat["PassengerId"].unique()
```

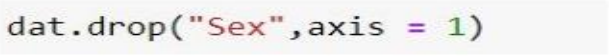
2. Value Count →

```
dat["Sex"].value_counts()
```

3. Null Value Check →

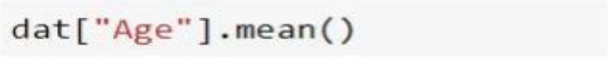
```
dat.isnull().sum()
```

4. Drop Feature →



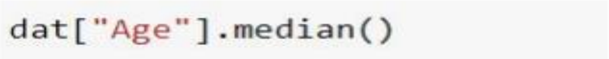
```
dat.drop("Sex",axis = 1)
```

5. Feature Mean →



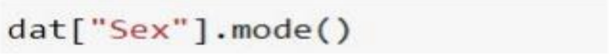
```
dat["Age"].mean()
```

6. Feature Median →



```
dat["Age"].median()
```

7. Feature Mode →



```
dat["Sex"].mode()
```

Data Cleaning Methods

1. Dropping ➔ Deleting the Rows or columns

2. Imputing ➔ Imputing the missing value by statistical Method (Mean, Median & Mode)

Data Visualization using matplotlib

.plot(): Used to Plot the Line graph

Syntax: plt.plot(x_data, y_data)

distplot( ):

Syntax: sns.distplot(<data>)

boxplot( ):

Syntax: sns.boxplot(x =<data>)

countplot( ):

Syntax: sns.countplot(data = <dataset>, x =<col name>)

barplot( ):

Syntax: sns.barplot(data=<dataset>, x=<column>, y=< column>, hue=< column>)

scatterplot( ):

Syntax: sns.scatterplot(data=, x=<column>, y=< column>, hue=< column>, style=< column>)

heatmap( ):

Syntax: sns.heatmap(data.corr(), annot=<True/False>, fmt=”0.1f”)

Feature Engineering

➔ It is the process of using domain knowledge to extract features (characteristics, properties, attributes) from raw data.

➔ Feature engineering is the process of selecting, manipulating, and transforming raw data into features that can be used in supervised learning.

Feature Transformation

- Data Cleaning
- Decomposing & Binning Features
- Outlier Detection
- Encoding
- Feature Scaling

Chapter-6

CONCLUSION

It was a wonderful learning experience for me while working on this project. This project took me through the various phases of project development and gave me real insight into the world of software engineering. The joy of working and the thrill involved while tackling the various problems and challenges gave me a feel of the developers' industry. It was due to this project I came to know how professional software is designed. The main aim of this project was to design and implement Diabetes Prediction Using Machine Learning Methods and Performance Analysis of that methods and it has been achieved successfully. The proposed approach uses various classification and ensemble learning method in which Logistic Regression is used. I hereby conclude my project design of diabetes prediction. Started with understanding what the project is about. Observed how the software works. Practical implementation of logistic regression came in handy. After using various functions of python programming language, the data was extracted successfully and the model was trained. In the end, the model was able to give appropriate results. Which is conformation that the product works properly and there are no errors in building it. Got 85% of accuracy with it. Now aiming to build similar projects. The Experimental results can be used by health care to take early prediction and make early decision to cure diabetes and save humans life.

List of Figures

REFERENCES

1. Paul Z, Eaton C. Understanding big data: analytics for enterprise class Hadoop and streaming data. McGraw-Hill Osborne Media;2011.
2. https://content.wisestep.com/conclusion-project-work/#:~:text=It%20was%20a%20wonderful%20learning,the%20world%20of%20software%20engineering.
3. https://www.researchgate.net/publication/347091823_Diabetes_Prediction_Using_Machine_Learning
4. https://www.studocu.com/in/document/indraprastha-institute-of-information-technology-delhi/computer-sciences-and-applied-mathematics/projectreport-diabetes-prediction/30748724