

UNIVERSITY PARTNER



UNIVERSITY OF  
WOLVERHAMPTON



HERALD  
COLLEGE  
KATHMANDU

## Project and Professionalism (6CS020)

# A2: Project Report Diabetes Prediction using Data Mining

Student Id : 1928731  
Student Name : Sandesh Lamsal  
Group : C3G1  
Supervisor : Raj Shrestha  
Cohort : 3

Submitted on : 14-june 2020

## Declaration Sheet:

(Presented in partial fulfillment of the assessment requirements for the above award.)

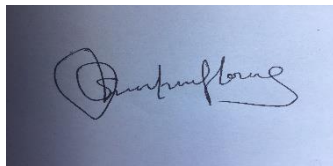
This work or any part thereof has not previously been presented in any form to the University or to any other institutional body whether for assessment or for other purposes. Save for any express acknowledgements, references and/or bibliographies cited in the work. I confirm that the intellectual content of the work is the result of my own efforts and of no other person.

It is acknowledged that the author of any project work shall own the copyright. However, by submitting such copyright work for assessment, the author grants to the University a perpetual royalty-free license to do all or any of those things referred to in section 16(l) of the Copyright Designs and Patents Act 1988. (viz. to copy work; to issue copies to the public; to perform or show or play the work in public; to broadcast the work or to make an adaptation of the work).

Student Name: **Sandesh Lamsal**

Student Number: **1928731**

Signature:

A handwritten signature in black ink on a light blue background. The signature is cursive and appears to read 'Sandesh Lamsal'.

Date: 11 may, 2020

**Abstract:**

Diabetes prediction system is made to simply predict the possibility of diabetes in a person. It will predict the possibility of diabetes in human beings by some general factors such as pregnancies, glucose, blood pressure, insulin, BMI, Diabetes Pedigree Function, age, and skin thickness. These all factors are sufficient to detect diabetes Meletus. All the processes and steps carried out to complete this project were explained in a form of this report. Different tools and techniques from various platforms likewise medical understandings and reviews, web development tools and frameworks, ML libraries, etc. were used to complete a project. The whole motivation behind the project is to serve people who are deprived of getting proper medical checkup and services, to take regular health status of the people and so on.

## **Acknowledgements:**

First, I would like to thank project supervisor Mr. Raj Shrestha who helped me to complete my project. He had given me well support and directions, sharing of tools and techniques, and reviewed my project time to time. As well, I would like to thank my reader Mr. Subiran Shrestha who also well supported me to complete my entire project and report.

I want to confess that, I could not able to complete this project without support and backbone provided by my supervisor. He is the only reason to motivate me when I am in trouble with different issues during the time of project development. This project is impossible without proper guidelines and milestones which I was able to set through his guidance. Respecting to my field of interest, he accepted my proposal and taught the project management techniques along with proper follow up in the development phage of project.

I would like to thank my personal academic advisor who given me support to continue the proposed project and deal with college and university team which provided many supports through the projects. All the materials provided from the college team were the graceful contents which helped us in every steps of project development.

Finally, I would like to thank all my colleagues who helped me in this project sharing important sources and information so that I able to use them in my project.

# Contents

1. Introduction to report structure: .....	1
1. Introduction:.....	2
1.1. History of diabetes: .....	2
1.2. Current Scenario:.....	3
1.2.1. General Introduction: .....	3
1.2.2. Introduction to an application:.....	5
1.2.3. Problem Statement: .....	5
1.2.4. Aims: .....	6
1.2.5. Objectives: .....	6
1.3. K-Nearest Neighbor algorithm: .....	6
1.4. Scope and Limitation of a project:.....	7
1.5. Limitations: .....	8
2. Literature Review:.....	8
2.1. Emergence of Artificial Intelligence: .....	9
2.2. Emergence of Machine Learning:.....	9
2.3. K-Nearest Neighbor Algorithm: .....	10
2.3.1. Features of KNN: .....	12
The algorithm of KNN is: .....	14
2.3.2. Drawbacks of KNN:.....	15
2.4. Different algorithms used before for diabetes Prediction:.....	15
2.4.1. J48 Decision tree: .....	15
2.4.2. Artificial Neural Network: .....	17
2.4.3. Logistic Regression: .....	19
2.5. Similar applications:.....	20
2.5.1. Diabetes: Management and blood sugar tracker app: .....	21
2.5.2. Blood Sugar Log: Diabetes Tracker: .....	21
2.5.3. BG Monitor Diabetes: .....	21
3. Main Body: .....	22
3.1. Development: .....	22
3.1.1. Iterative model: .....	22
3.2. System Architectures:.....	47
3.3. Tools and techniques:.....	49
3.4. System Requirement: .....	52

3.4.1.	Hardware Requirement:	52
3.4.2.	Software Requirement:	52
3.5.	Implementation:	53
3.5.1.	Data Collection:	53
3.5.2.	Data Processing:	53
3.5.3.	Data splitting:	58
3.5.4.	Building model:	59
3.6.	Implementation (diabetes prediction system):	60
3.7.	Wireframes:	62
3.7.1.	Home and About Us:	62
3.7.2.	Services and detect:	63
3.7.3.	Contact:	63
3.7.4.	Register:	64
3.7.5.	Logout:	65
3.8.	Page Views:	66
3.8.1.	Homepage:	67
3.8.2.	Register:	68
3.8.3.	Login:	69
3.8.4.	Database:	70
3.9.	Testing:	70
3.9.1.	White-Box Testing:	71
3.9.2.	Unit testing:	71
3.9.3.	Integration testing:	71
3.9.4.	Black-box testing:	71
4.	Answering academic question:	86
5.	Conclusion:	87
5.1.	Future Escalation:	87
6.	Critical Evaluation and Findings:	88
7.	References:	93
8.	Abbreviations:	95
9.	Appendix:	95
9.1.	Different analysis and visualizations done to develop models:	95

## Table of Figures

Figure 1 History of diabetes .....	3
Figure 2 Diabetes info stats.....	3
Figure 3 KNN algorithm introduction.....	7
Figure 4 KNN detailed.....	11
Figure 5 Distance formula .....	12
Figure 6 Canberra distance .....	13
Figure 7 Sorensen Distance .....	13
Figure 8 MCD Distance .....	13
Figure 9 Average Distance .....	14
Figure 10 Cosine distance.....	14
Figure 11 Decision tree Architecture .....	16
Figure 12 Entropy.....	17
Figure 13 calculate gain .....	17
Figure 14 ANN structure.....	18
Figure 15 ANN structure compared with brain analogy.....	18
Figure 16 Logistic regression sigmoid curve.....	20
Figure 17 Iterative Model.....	23
Figure 18 Diabetes related information I.....	25
Figure 19 Diabetes related information II.....	26
Figure 20 Users using web application and max used application.....	27
Figure 21 Health care application users and excited users to use healthcare applications .....	28
Figure 22 Family information about diabetes and survey on use of contact form.....	29
Figure 23 Excitements to use healthcare application in future .....	30
Figure 24 FDD diagram.....	32
Figure 25 Use Case Diagram .....	36
Figure 26 Activity diagram .....	37
Figure 27 Sequence diagram for an admin.....	39
Figure 28 Sequence diagram for a user .....	40
Figure 29 Class Diagram.....	41
Figure 30 Fill Contact Form .....	42
Figure 31 View Uploaded Photos .....	43
Figure 32 Check Diabetes .....	44
Figure 33 Entity Relationship Diagram .....	46
Figure 34 Client server (3-tier) architecture .....	48
Figure 35 Early data existence .....	53
Figure 36 Described data .....	54
Figure 37 Visualization "YES" vs "NO" .....	54
Figure 38 Histogram Plot.....	55
Figure 39 Handling Missing Values .....	56
Figure 40 Pair plot.....	57
Figure 41 Corelation.....	58
Figure 42 Training Test Spilt .....	59
Figure 43 Model building .....	59

Figure 44 Implementation of Model .....	60
Figure 45 Home and About Us .....	63
Figure 46 Services and detect.....	63
Figure 47 Contact.....	64
Figure 48 Register.....	65
Figure 49 Logout.....	66
Figure 50 Home Page I .....	67
Figure 51 Home Page II .....	68
Figure 52 Register.....	69
Figure 53 Login .....	70
Figure 54 KNN model.....	89
Figure 55 Decision tree model.....	89
Figure 56 Logistic regression model.....	90
Figure 57 Random Forest Classifier Model .....	90
Figure 58 MLP model.....	91

## Tables:

Table 1 Test case 1.....	73
Table 2 Test Case 2.....	76
Table 3 Test case 3.....	76
Table 4 Test Case 4.....	77
Table 5 Test Case 5.....	78
Table 6 Test case 6.....	80
Table 7 Test Case 7 .....	81
Table 8 Test Case 8.....	83
Table 9 Test Case 9.....	84
Table 10 Test case 10.....	84
Table 11 Test Case 11 .....	86

## Academic Questions:

*How this application detects diabetes in a Patient and what are its features?*



## **1. Introduction to report structure:**

The structure of this report is listed out as given below:

### **a. Introduction:**

This section consists of general introduction of project, scopes and limitations, artifacts, problem domains, algorithm, aims and objectives of preparing this project is introduced.

### **b. Literature review:**

This section consists of some details of researched done on different facts to develop and complete this project. It consists of an explanation of the model used in a project with introductions to some other models been practiced by other researchers on this domain. This part also talks about similar types of applications existing which functions to detect diabetes and diabetic based factors.

### **c. Main body:**

It consists of different parts including system development, architectural and functional designs, system implementation and techniques, fact finding techniques, system design representation, ERDs, and as well development of AI are explained here.

### **d. Academic questions:**

Here, academic question is answered. Working structure of an application is explained in brief.

### **e. Conclusion:**

This section will conclude the report with discovery made to this application with some probable future plans.

- f. References**
- g. Appendix**
- h. Abbreviations**

## **1. Introduction:**

### **1.1. History of diabetes:**

This disease was first characterized by the “too great emptying of the urine” to find its place in antiquity through Egyptian manuscripts dating back to 1500 BC.

It was named as ‘madhumeha’ by some Indian physicians because of ant attract to the urine of those people who are the patient of this disease. Two Indian physicians i.e., Sushruta and surgeon Chakra in around 400-500 AD were able to identify two different types of diabetes and named them Type I and Type II.

In 1869, Paul Langerhans distinguished the cell that came to be known as the 'islets of Langerhans'. The name for example insulin which could cut down glucose levels was begotten in 1909-1910 separately by Mayer and Schaefer. (Lakhtakia, 2013)

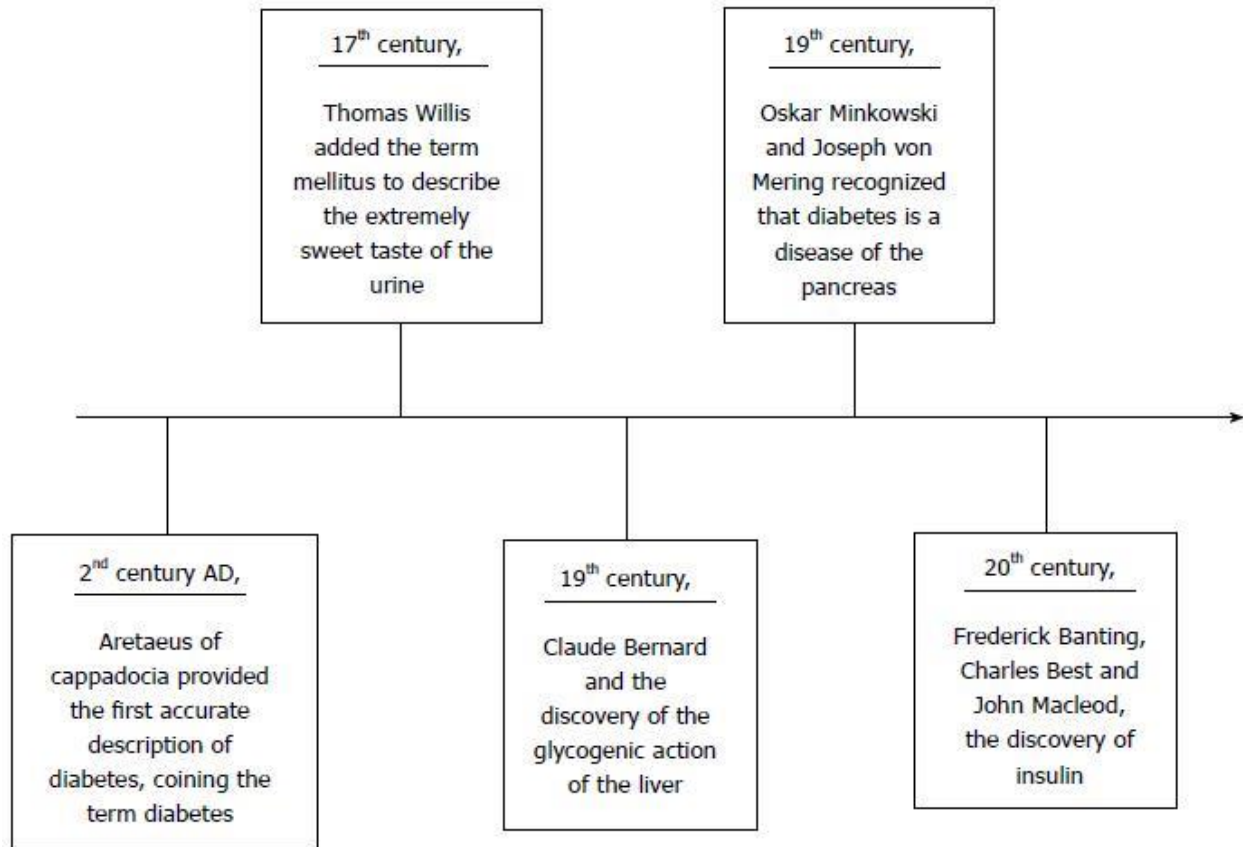


Figure 1 History of diabetes

## 1.2. Current Scenario:

### 1.2.1.General Introduction:

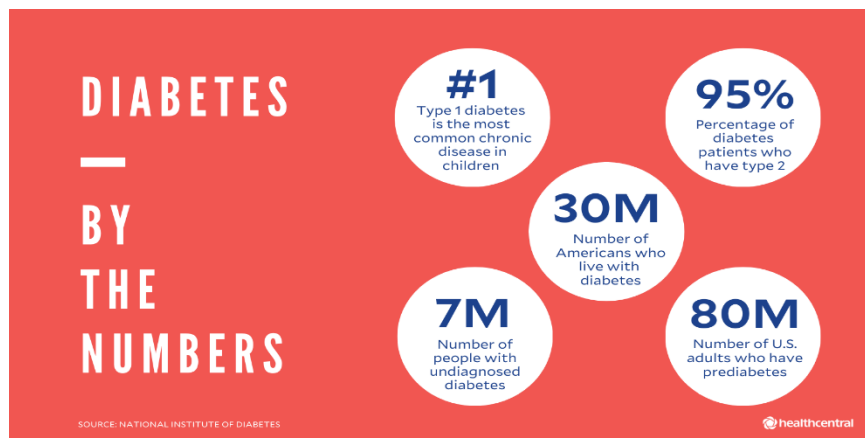


Figure 2 Diabetes info stats

It is the fastest-growing global problem with huge social, health, and economic consequences. As per the estimation data in 2010, it was claimed that there were globally 385 million people who are suffered from this disease. The two main reasons for this disease are said to be obesity and ageing. (K., et al., 2013)

It is a condition characterized by a metabolic disorder in which patients may have a high level of sugar in their blood due to less secretion of insulin which is an essential bio factor in the human body to balance the level of glucose. It also occurs if the body does not react properly with insulin.

The standard definition of ICMR defines diabetes as “metabolic cum vascular syndrome of multiples etiologies characterized by hyperglycemia with disturbances of carbohydrate-protein and fat metabolism resulting from insulin secretion, insulin action, or both”.

Two types of diabetes are defined in which must of the diabetes patients are suffering from type II as per the data taken.

Some factor generally matters in having diabetes i.e., heredity character of the family, age. Enlarged waist or upper body adiposity, Body Mass Index, presence of hypertension, weight gain, Sedentary lifestyle, gestational diabetes, and so on.

People have to check this out after they have seen some general symptoms such as weight loss, frequent fatigue, irritability, dry mouth, burning, pain, etc. and so on. (A., 2014)

Thus, this application will help the people to test their health condition easily with the help of local health posts saving their time to visit hospitals and medical colleges to test whether they are the carriers of diabetes or not.

### **1.2.2.Introduction to an application:**

It is a web application that is equipped with an intelligent machine learning technique i.e., KNN algorithm which features to predict whether the patient is having the problem of diabetes or not. As being an academic project and works regarding the health issues, hospitals are not allowed to provide data for the academic project in Nepal, I used a dataset which is being taken from the hospital mainly situated in Germany. The main feature of this application is to predict the diabetes of patients feeding certain input from them. Input are some common possible factors that play a major role to indicate the possibility of diabetes.

Coming out from the main goal of this project, it also consists of different sub-features such as information about diabetes, preventive measures, research in diabetes, information of some possible hospitals which are related to diabetes, storage of upcoming data of diabetes for future analysis, contact forms and some other important features.

The main reason behind making this project targeting the people in a remote area who are deprived of getting advanced hospital services. It will help them to detect their diabetes status with the help of medical workers who work on their local health posts. They can take advantage of this service and plan for the earlier step as per the result obtained.

### **1.2.3.Problem Statement:**

- The difficulty of hospital services in remote and rural areas
- Lack of proper check-up equipment to the countryside
- People have to pay a lot of money to understand whether they have diabetes or not.
- The problem of traveling to long-distance hospitals for minor-checkups.
- Lack of Specialized check-up services in rural health and sub-health posts regards diabetes.

#### **1.2.4.Aims:**

- To save people's money from minor checkups of diabetes.
- To provide diabetes checkup services to rural and remote areas
- To make people available to check up if they face any symptoms related to diabetes
- To make people health conscious and ready for further treatment if they possibly have diabetes.
- To develop user friendly interfaces

#### **1.2.5.Objectives:**

- Collection of data and dataset from reliable sources.
- Implementation KNN algorithm to solve the problem.
- Comparison of other different algorithms and as well as Neural Network for implementation of this domain
- Train data using KNN
- Test whether the model predicts accurately or not
- Use the Output to understand whether people suffering from diabetes?
- Provide information such as sign, symptoms, preventive measures about the diabetes
- Provide information about some Nepalese hospitals related to healing to diabetes.
- To implement sign up and login pages

### **1.3. K-Nearest Neighbor algorithm:**

It is one of the famous machine learning algorithms which is mostly used for pattern recognition and data mining for classification. It is famous because of its simplicity, low complexity, and error rate. It is simple in the calculation and can be applied to high dimensional data sets. It is measured by the distance in the feature space hence known as the K Nearest Neighbor algorithm.

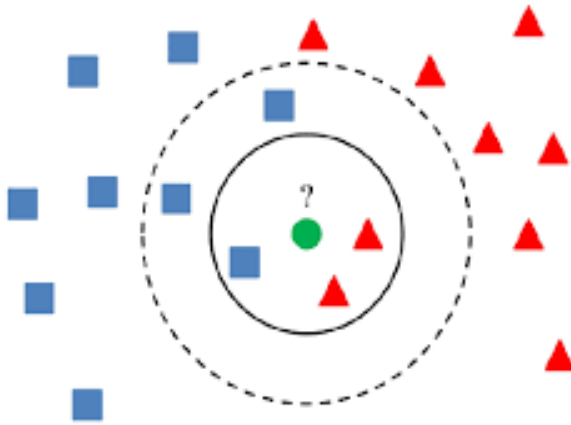


Figure 3 KNN algorithm introduction

(Alaliyat, 2020)

To determine the feature distances, some tools are used such as measurement of Euclidean distance and Manhattan distance which have their way of measuring the distance.

Analyzing the simplicity of my project featured with data mining and availability of limited records in a dataset, I concluded to use this algorithm which is giving its performance better comparatively to some famous machine learning algorithms and as well as Neural Nets. As deep learning and Neural Networks are advanced and best performers than machine learning, but in the case of my data analysis, I found KNN working better.

More details about this algorithm will be explained further.

#### 1.4. Scope and Limitation of a project:

Analyzing the history and presence of patients of diabetes, we can view result in emerging scope. As per the estimated result of causalities, 422 million of peoples were victims of this disease in 2014 which shows huge growth then comparing to 108 million of people in 1980. The percentage growth also seemed to be 4.7% to 8.5% of the total population. (World Helath Organization, 2016)

As per the report presented by WHO, it is seemed to be the major problem of the whole world and situation is becoming out of control. Condition is becoming worse because of huge victims of diabetes which is difficult to be handled by medical workers and as well as poverty of people that suppress them to do periodic medical check-ups. This project will be the solution to such peoples who want to do their medical tests but under the poverty. They will absolutely get check-up services from small health-care service providers, health posts and sub-health posts who will provide them details about these common factors of diabetes and patients can perform further action according to upcoming result. As this application becoming an academic and needed to be completed on certain milestones, its features are limited but it will be managed as to be equipped with other different features in a future such as diabetic retinopathy, use of more medical termed factors, genetical information of parents about appearance of diabetes, etc.

### **1.5. Limitations:**

This application is capable of predicting diabetes based on few common factors that seems in a diabetic patient. This app will not be suitable to each and every place of the world due to nature of sign and symptoms of diabetes to the people vary from one place to another due to their food habits and locations. Many other factors are responsible to suffer from diabetes which is not included in a project. These all are projects limitations.

## **2. Literature Review:**

Diabetes is becoming one of the vital diseases in current scenario. Medical research organizations had developed different medicines and techniques to cure diabetes and still, many researches are going on to develop enhanced equipment's and medicines so that this disease will be cured with more better approaches. Along with different techniques to detect diabetes, artificial intelligence will be the modern approach.



AI technology will uplift medical approaches and instruments in a field of healthcare and promotes to reach health industry to an enhanced level. Extracting a piece of field from the ocean of medical predictions, I have tried to develop a system that will help to detect possibility of diabetes mellites to a person taking some factorial information's related to diabetes.

## **2.1. Emergence of Artificial Intelligence:**

Artificial Intelligence is emerging on day-to-day basis. Many domains are shifting their technologies along with an Artificial Intelligence. This is the technology which was emerged in 1956 for the first time in the Dartmouth conference. (McCarthy, 1955) The growth of artificial intelligence in multi-disciplinary field will be beneficial to business, services and although the government works. The emergence of internet services, storages and powerful hardware lead to the strong emergence of artificial intelligence. It is not underpinned by any of the theories and emerging as strong tool in any of the domain. Few decades ago, it assumed that AI would become standalone system in a future likewise robots or expert systems, but today, Intelligence systems can be found combined and equipped with different technologies. For examples driverless cars, google translators, amazon assistants, etc. are the examples of advanced systems developed at this decade. All those advanced algorithms are the developments along with small algorithms and techniques called Machine Learning.

## **2.2. Emergence of Machine Learning:**

Among various subsets of AI, ML is the one. ML is a technique that helps to learn data and information provided and performs action automatically as per the pattern gained from the provided data. Abilities of computers and devices to communicate with humans, perform different predictions, business logics and intelligence, fraud detections, etc. many more are the grace of Machine learning. All advanced algorithms of artificial intelligence were developed in a base of an artificial intelligence. (Marr, 2016)

Some algorithms mostly used in Machine Learning for data and business intelligences are as follows:

- Decision tree
- KNN
- Random Forest
- Linear Regression, etc.

These all algorithms are used to find patterns in data and use dataset as the guidance point to perform further operations in Artificial Intelligence.

### **2.3. K-Nearest Neighbor Algorithm:**

It is a model based on supervised learning used as an algorithm that classifies whether the patient is diabetic or non-diabetic through an application. It classifies an algorithm into two classes i.e., either diabetic or non-diabetic understanding the pattern given by the past data.

This algorithm is introduced by Fix and Hodges in 1951/1952. It is a method used for classifying objects based on closest training examples in the feature space. It is also known as lazy learning because it does not learn differentiative function from the training data instead it memorizes the dataset. The prediction in KNN seems to be quite expansive because of its property to search the nearest neighbor in an entire training set during each time of prediction. But we can apply some tricks such as Ball Trees and KDtrees to speed up its performance in a little bit. (sebastianraschka, 2020)

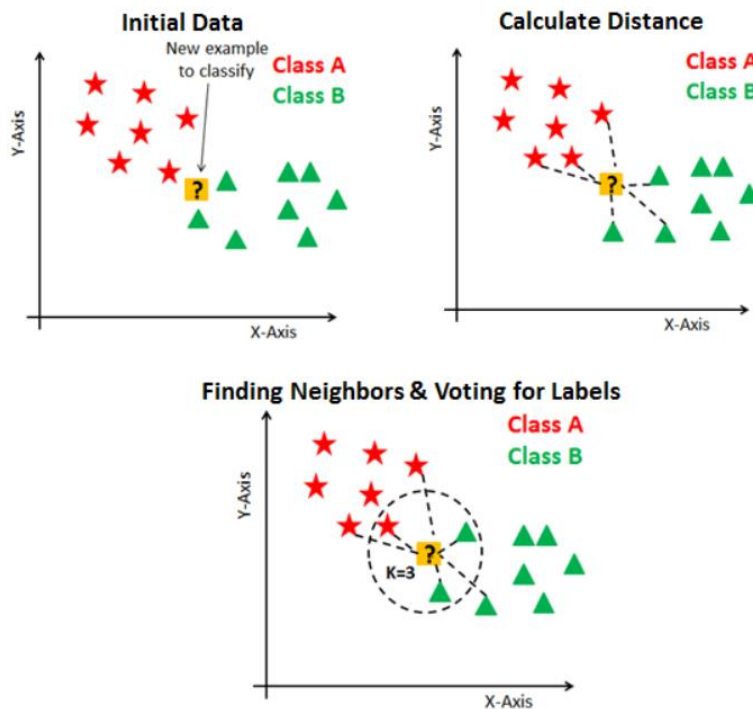


Figure 4 KNN detailed

K-nearest neighbor expect that observations, which are near one another, are likely to have a similar classification. The probability that a point  $x$  has a place with a class can be evaluated by the extent of preparing focuses in a predefined neighborhood of  $x$  that have a place with that class. The point either be ordered by greater part vote or by a similitude degree whole of indicated number ( $K$ ) of closest focuses as appeared in a figure above. In casting a ballot, greater part of focuses in the local having a place with each class is tallied, and the class which has higher extent of focuses has a place is the most probable order of  $x$  (Peterson, 2009) (Weinberger, et al., n.d.)

### 2.3.1.Features of KNN:

- All examples of information related to the focuses in an n-dimensional Euclidean distance.
- Classification is deferred till another distance shows up
- In KNN, the characterization is done by contrasting component vectors of the various focuses in a space area.
- The target region might be discrete or genuine esteemed.

(Saxena, et al., 2014)

In KNN, different distances are used to calculate nearby distance such as Euclidian, Manhattan, Mahalanobis, Cosine, Hamming, Jaccard, Spearman, Minkowski distances, etc. Among these, Euclidian distances are preferred in a major. Because of its simplicity and common way of measuring the distance. (Prasath, et al., 2019) In KNN, preparing tests are characterized by n-dimensional properties. The preparation tests are put away in an n-dimensional space. At the point when the test is given, k-nearest neighbor begins looking "k" preparing test which is nearest to the obscure sample or test sample. Euclidean distance defines the closeness of sample data points.

The Euclidean distance between two points are defined by equation:

$$d(P, Q) = \sum_{i=1}^n (P_i - Q_i)^2$$

Figure 5 Distance formula

Where, P and Q be the two different points given by

P (p<sub>1</sub>, p<sub>2</sub>,.....p<sub>n</sub>)

And

Q(q<sub>1</sub>,q<sub>2</sub>,.....q<sub>n</sub>)

(Saxena, et al., 2014)

Similarly, other many distance formulas that can be used in KNN algorithms to find the shortest distances are given below:

**Canberra distance (canD):**

It is the weighted variant of Manhattan distance, where the total distinction between the characteristic estimations of vectors x and y isolates by the total of supreme qualities preceding adding.

$$CanD(x, y) = \sum_{i=1}^n \frac{|x_i - y_i|}{|x_i| + |y_i|}$$

Figure 6 Canberra distance

**Sorensen distance (SD):**

It is an adjusted Manhattan metric, where the added contrasts between the quality's estimations of vectors x and y are normalized by their added attributes esteems.

$$SD(x, y) = \frac{\sum_{i=1}^n |x_i - y_i|}{\sum_{i=1}^n (x_i + y_i)}$$

Figure 7 Sorensen Distance

**Mean character Distance:**

It is also known as average Manhattan, or Gower distance.

$$MCD(x, y) = \frac{\sum_{i=1}^n |x_i - y_i|}{n}$$

Figure 8 MCD Distance

**Average distance:**

It is the modified version of Euclidian distance. This distance formula tries to cover some limitations of Euclidian formula i.e. "if two data vectors have no attribute values in

common, they may have a smaller distance than the other pairs of data vectors containing the same attribute values”.

$$AD(x, y) = \sqrt{\frac{1}{n} \sum_{i=1}^n (x_i - y_i)^2}$$

Figure 9 Average Distance

### **Cosine distance:**

It is additionally called angular distance gotten from the cosine similitude that quantifies the edge between two vectors, where cosine separation is acquired by taking away the cosine comparability from one.

$$CosD(x, y) = 1 - \frac{\sum_{i=1}^n x_i y_i}{\sqrt{\sum_{i=1}^n x_i^2} \sqrt{\sum_{i=1}^n y_i^2}}$$

Figure 10 Cosine distance

All these are the distance formulas that can be used in KNN to calculate distance between two attributes to find out nearest distance between them. (Prasath, et al., 2019)

### **The algorithm of KNN is:**

#### **Step I:**

Every one of the new occurrences is checked with the effectively accessible cases, in view of various distance measurement formulas explained above and classified with K value.

#### **Step II:**

The distance will be less when the instances are increasingly comparative or the other way around.

**Step III:**

Watch the distance K-value and example. In light of these perceptions' occasions are assigned to a particular class.

**Step IV:**

The prediction depends on the K value. So KNN classifier will completely rely upon doled out estimation of K. Here K represents the quantity of closest neighbors and for the distinctive value of K, the result may fluctuate.

**Step V:**

Decide the Value of K for Dataset picked for classification accuracy.

**2.3.2.Drawbacks of KNN:**

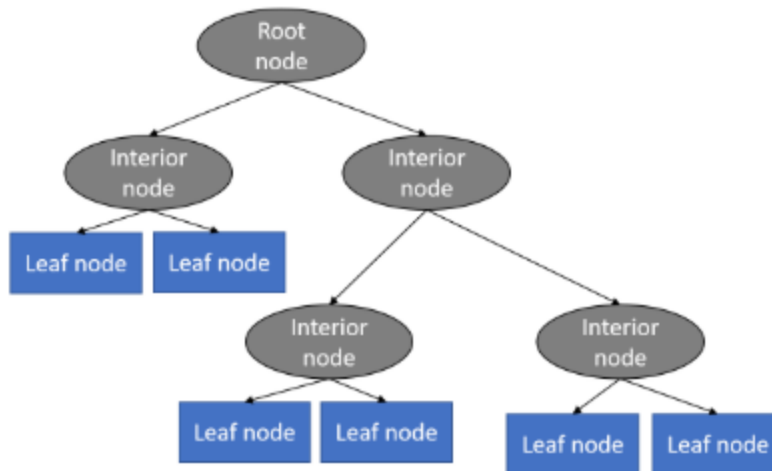
- We must work out on distance of each occasion (instances) to all other training samples which increases the computation cost.
- Large amount of data set is required along with high memories space.
- It is not suitable on large multidimensional dataset. Increase in dimension will definitely reduce accuracy.

No standard is accessible to decide the estimation of the parameter of K. (K speaks to the number of nearest neighbors) (Cheng, et al., 2014)

**2.4. Different algorithms used before for diabetes Prediction:**

Different algorithms were used to develop diabetes prediction system. All these different algorithms have their different results and outputs. Some of the famous algorithms had been practiced and going to be discussed are:

**2.4.1.J48 Decision tree:**



*Figure 11 Decision tree Architecture*

This algorithm is an expansion of the ID3 decision tree. Some extra highlights are found in this algorithm that represents missing qualities, decision tree pruning, consistent characteristic worth reaches, derivation rules, and so forth. In the WAKA mining tool, J48 is an open-source java execution of the c4.5 algorithm. This algorithm produces the principles for the prediction of the target variable. With the assistance of this tree characterization algorithm, the dissemination of the information will be justifiable and easier to classify.

Some basic steps of this algorithm are given below:

### **Step I:**

In the examples has a place with a similar class, the tree speaks to a leaf so the leaf is returned by naming with a similar class.

### **Step II:**

At that point the best trait is found based on the present determination rule and that property chose for expanding.



### Step III:

At that, the best attribute is discovered dependent on the current assurance standard and that property decided for branching.

Loss function or the degree of loss found on the dataset is calculated using Entropy which is also called degree of randomness.

The entropy is calculated by formula given below i.e.

$$\text{Entropy}(\vec{y}) = - \sum_{j=1}^n \frac{|y_j|}{|\vec{y}|} \log \left( \frac{|y_j|}{|\vec{y}|} \right)$$
$$\text{Entropy}(j|\vec{y}) = \frac{|y_j|}{|\vec{y}|} \log \left( \frac{|y_j|}{|\vec{y}|} \right)$$

Figure 12 Entropy

And the gain is calculated by formula

$$\text{Gain}(\vec{y}, j) = \text{Entropy}(\vec{y}) - \text{Entropy}(j|\vec{y})$$

Figure 13 calculate gain

(Kaur & chhabra, 2014)

### 2.4.2. Artificial Neural Network:

It is the advanced model of Artificial Intelligence which was introduced few decades ago. It refers to the modelling of human brain in terms of two aspects i.e. structure and function. This emerging algorithm and techniques in artificial intelligence works kind similar to human brain and represented as a brain anomaly. It performs as an ability of our human brain such as logical predictions, controlling, optimization, association, thinking,

reasoning and so on. . (EL\_Jerjawi & Abu-Naser, 2018) Its structure is represented similar to human brain as given below:

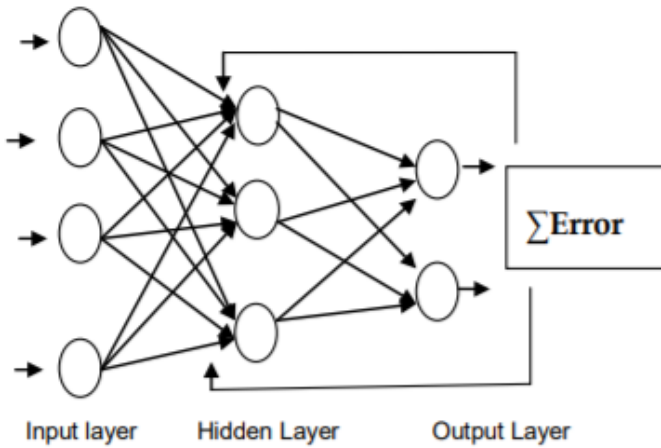


Figure 14 ANN structure

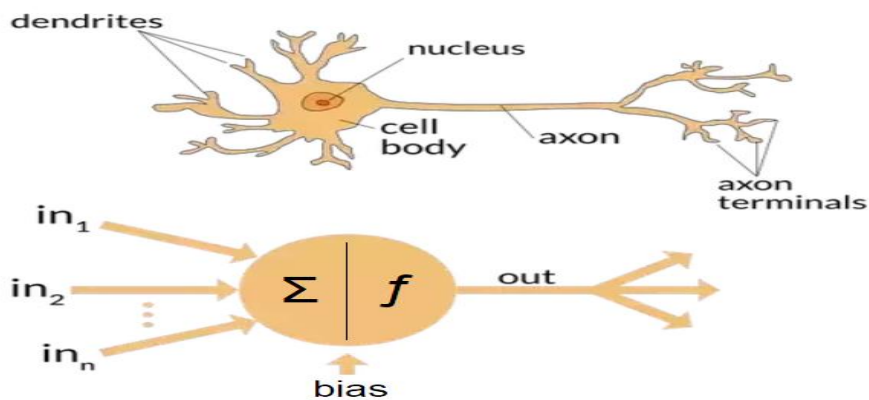


Figure 15 ANN structure compared with brain analogy

(Nagyfi, 2018)

The weighted sum in the ANN is given by:

$$T.P_y = \sum W_{xy} X_y + \theta_y$$

Images represented above is a structure of ANN, which seems as similar to the neurons in human brain. Input layer is the first layer that percepts data and information from the different sources as done by dendrites as anomaly in brain. Similarly, all those inputs were functioned in various mechanisms in inner layers and at last the output is given by output layers after various works on it. In the hidden layers, different activation functions were used to change the input types to non-linear because all the problems existing in a real world seems to be deviating from condition of linearity. (Azam & , 2000)

Different types of activation functions are chosen in the model as per type of model that we going to develop. Some activation functions such as step function, Linear function, Ramp function, Relu function, tanh function, Leaky Relu function, etc. were the mostly used activation functions.

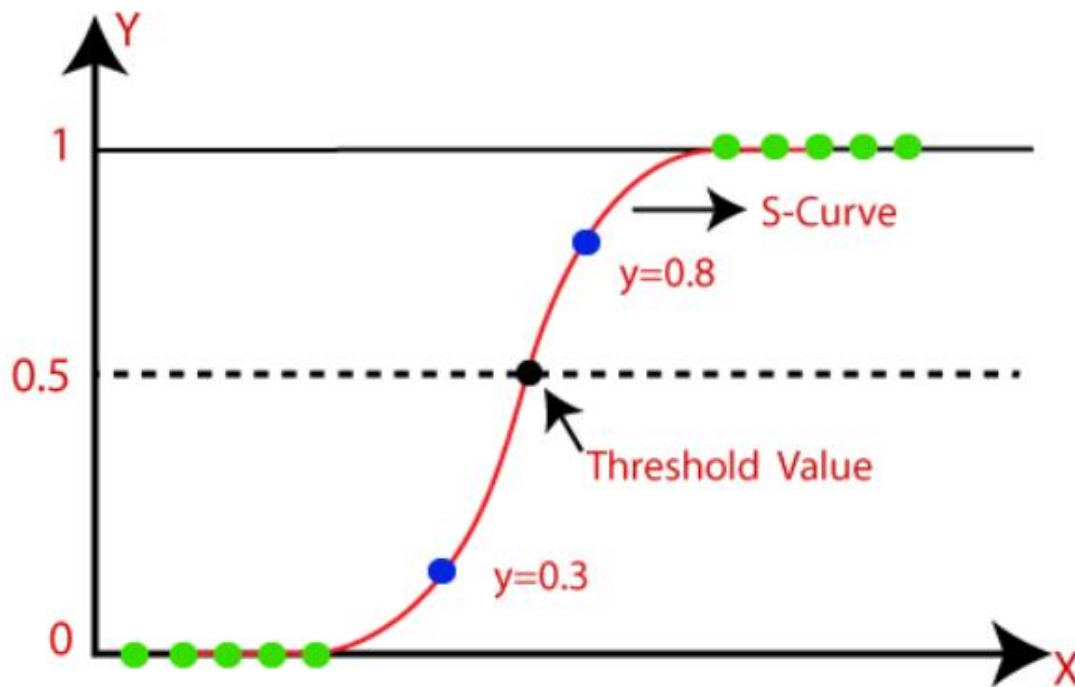
Cost functions in this model is mainly calculated using back propagation and weightage are tried to be balanced by giving both input and output result to the model.

Hidden layers also can be from single to many as per the logical requirement and complexity of works that we going to be done. We can set single hidden layer for this diabetes prediction system because it only involves binary classifications i.e. “Yes” or “No” which is can be almost classified using one or two hidden layers.

### **2.4.3.Logistic Regression:**

It is a regression model where the dependent variable is categorical i.e. can takes only two values e.g. 0 or 1 that represents an outcome of either yes/no, qualified/disqualified, pass/fail, etc. This algorithm is widely used in different fields such as engineering to predict probability of particular process, medicines to predict the mortality of an injured patient, marketing, economics, business and so on. It is derived from linear regression which generally suits for classification problem while showing particular properties by different types of data in a dataset. (Joshi & Chawan, 2018)

**Working mechanism:**



*Figure 16 Logistic regression sigmoid curve*

In this type of classification, threshold is set between the S-curve and partition between two curves and the output is classified into two different classes. As shown above in a figure 0.5 is the threshold value set to the sigmoid curve which predicts values above as class 1 and below to class 0.

## **2.5. Similar applications:**

I have done some researches on similar kinds of systems based on diabetes detection. Among many applications that works on diabetes, I have chosen two different system which are in high use by patients through the world.

These are:

### **2.5.1.Diabetes: Management and blood sugar tracker app:**

It is one of the well-known diabetes blood sugar tracker applications which can be found on play store with more than 500k+ downloads. It seems to be kind similar to an application which I am going to build as FYP. It is a full featured application which almost developed to manage all different types of diabetes such as prediabetes, Type I, Type II, Gestational and LADA. It takes different factors such as rapid/short acting insulin, immediate/Long-acting insulin, Medications, Gender, Height, weight, glucose levels at various target range, Insulin sensitivity, carbohydrate ratio, etc. to track the level of blood sugar in a body.

In contrast, my application is web-based application and have some common factors which is slightly different than this kind of application. As both are the diabetic based applications but have slightly different purposes i.e., prediction of diabetes and tracking of blood sugar level.

### **2.5.2.Blood Sugar Log: Diabetes Tracker:**

This application is also based on android application which is responsible for tracing diabetes in human. This application also takes some parameters such as blood sugar, medication, blood pressure Weight, etc. as some factors that provides an information about diabetes.

In contrast, application is using different factors than this application to detect diabetes.

### **2.5.3.BG Monitor Diabetes:**

This app provides insulin bolus calculation and blood sugar targets. It also handles two different types of diabetes i.e., Type 1 and Type 2. It is also an android based application.

Similarly, other different applications can be found such as Glucose Buddy, MySugr Diabetes Logbook, DiabetesConnect, SugarSense, One Drop, etc. and so on which are responsible to detect diabetes and different factors of diabetes. Each of them has their own functions and uses different factors as an input to contribute diabetic-checkups and based on different platforms like IOS, android, web and so on.

### **3. Main Body:**

#### **3.1. Development:**

The methodology used to make this web-based diabetes prediction system is the iterative Model. The reason behind the use of this methodology is because of its simplicity and a single-handed project which needed to be developed with a single contribution. It will be equipped with more features later in the future releasing newer versions with more effective features and designs. Overview of this model is given below:

##### **3.1.1.Iterative model:**

It is one of the famous methods of software development whose process starts with the simple implementation of a subset of system requirements. At each iteration, it gives emergence to a new design and functionality in a system. (Ivivity, 2019)

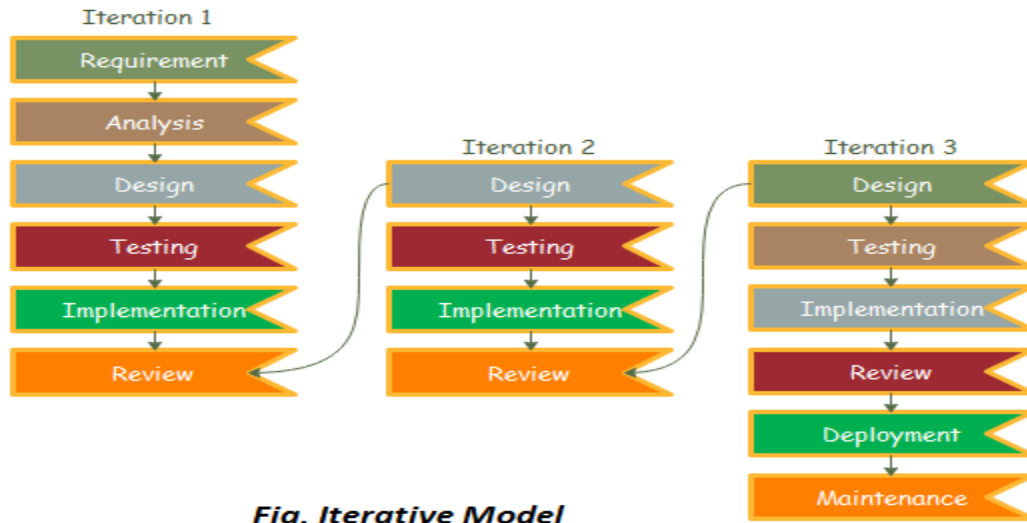


Figure 17 Iterative Model

(Javatpoint, 2018)

To develop the software with this methodology, different steps are followed which are given below:

#### 3.1.1.1. Feasibility study:

It is the first process in software development methodology which follows different steps to be performed after identification of the project. This phase understands the feasibility of the project i.e. problem analysis, legal permissions, cost estimation, time estimation, technical considerations, etc. It helps to minimize the risk of being project failure. (Kenton, 2019)

#### 3.1.1.2. Requirement analysis:

This step of the iterative model that deals to identify the requirements to deploy the project successfully. It involves the tasks to identify all the systems, technical, and tools that are required to solve the problem. It helps to address all the system requirements from clients and its documentation for its proper use. It is an essential part to address the problem.

#### **3.1.1.2.1. Fact finding techniques:**

There are several fact findings techniques were used to complete this project successfully. Fact find techniques means different types of researches and observations done that are responsible to complete the whole project.

Some techniques used to make this project completed are:

##### **3.1.1.2.1.1. Research and site visits:**

This technique has played the major role to complete this project. Researches on different contents of the project, use of their researches, information, data sources, technologies, etc. are done which are playing the major role in completion of this project.

##### **3.1.1.2.1.2. Sampling of documentations:**

Many documentations about the model and systems development were sampled and different functions are used which are pre-existing in a library. For example, ML libraries documentations, Django documentations, etc. were helped a lot in development of this project.

##### **3.1.1.2.1.3. Interview:**

This technique also has played an important role in a project. Asking of some technical and medical related questions to medical related personnel is done. Interview of common users are also taken which helped me to develop UI designs as per users prospective.

##### **3.1.1.2.1.4. Observation:**

Observations on different applications such as One Drop, Sugar Sense, DiabetesConnect, etc. are done which direct and indirectly have provided numerous ideas to develop medical web applications and main areas to focus on.



### 3.1.1.2.1.5. Google form surveys:

I also have done two different surveys with the help of google form services. I have taken some ideas about the uses of applications by the people and as well as taken some information about UX and UI so that people will find it easier to use application. Moreover, next survey is with peoples related to medical fields where it collects some essential information about diabetes.

#### 3.1.1.2.1.5.1. Survey about diabetes with health workers:

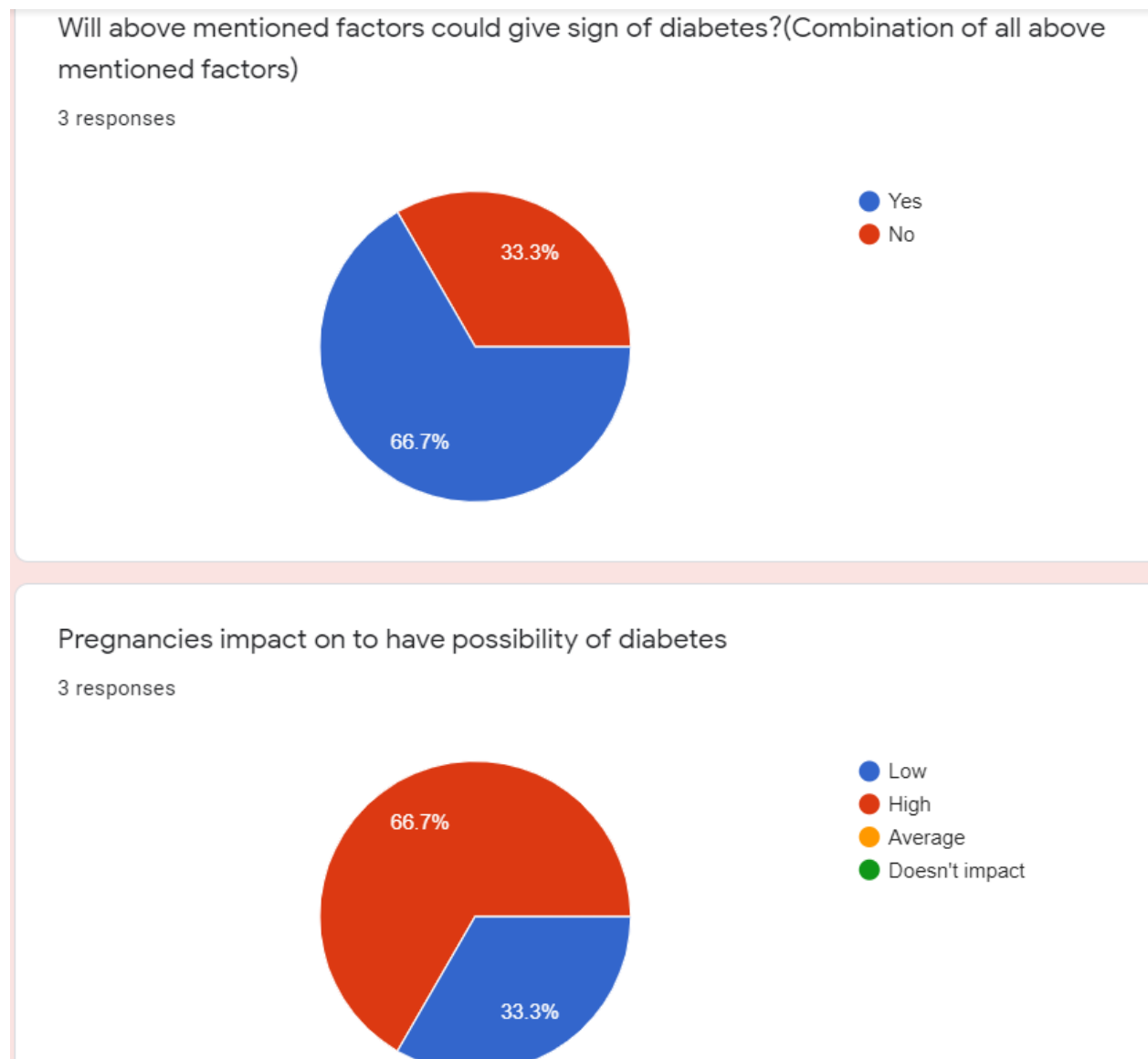
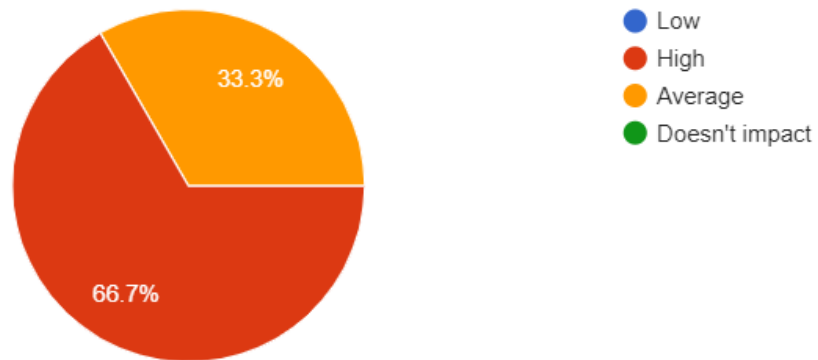


Figure 18 Diabetes related information I

### Age impact on to have possibility of diabetes

3 responses



### Gender impact on to have possibility of diabetes

3 responses



Figure 19 Diabetes related information II

### 3.1.1.2.1.5.2. Surveys on targeted users and their feedback on this application

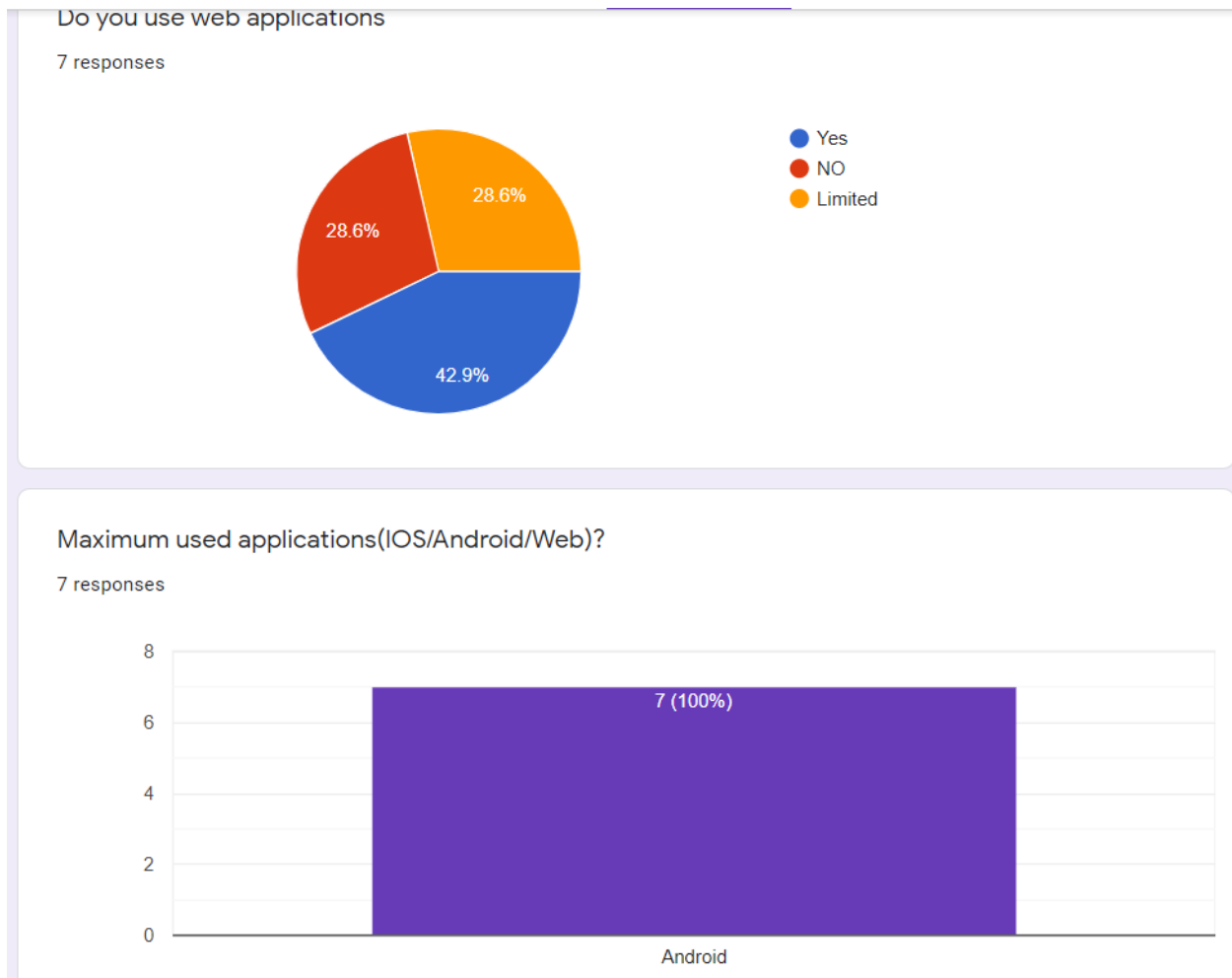
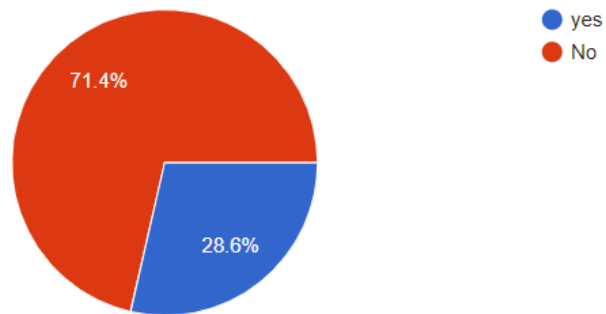


Figure 20 Users using web application and max used application

Have you ever used Health Care application?

7 responses



Are you excited to use Diabetes Predictor developed using Artificial Intelligence?

7 responses

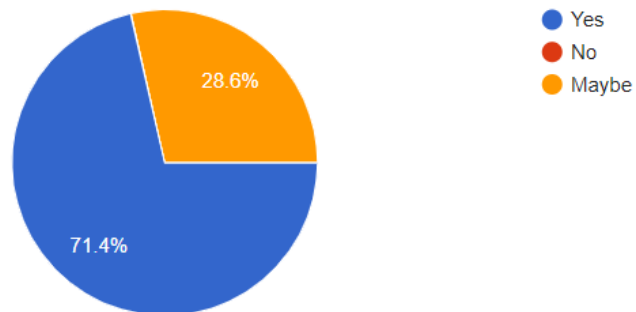
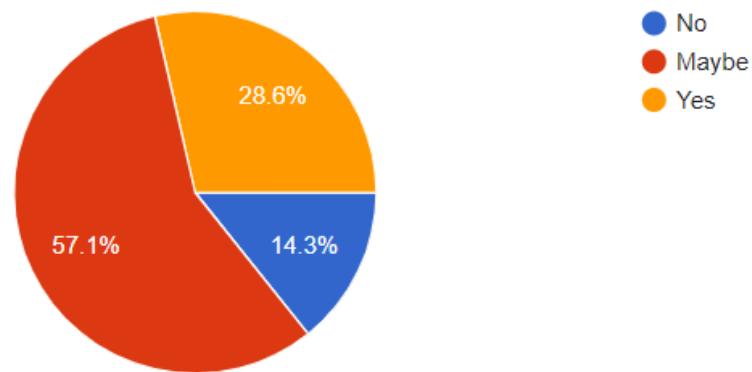


Figure 21 Health care application users and excited users to use healthcare applications

---

Do you want contact page so that you can contact with admin?

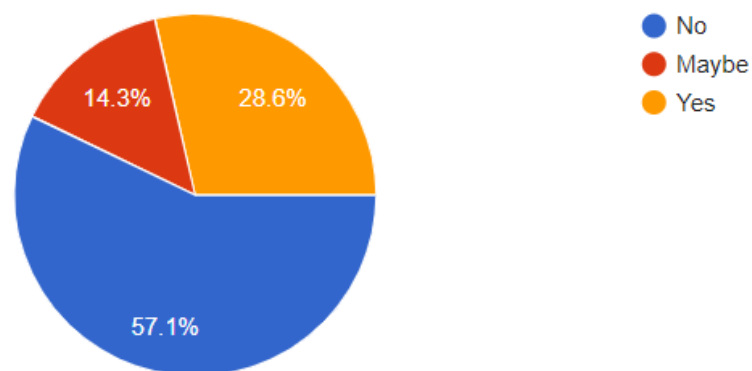
7 responses



---

Do any member in your family are victims of Diabetes Type I?

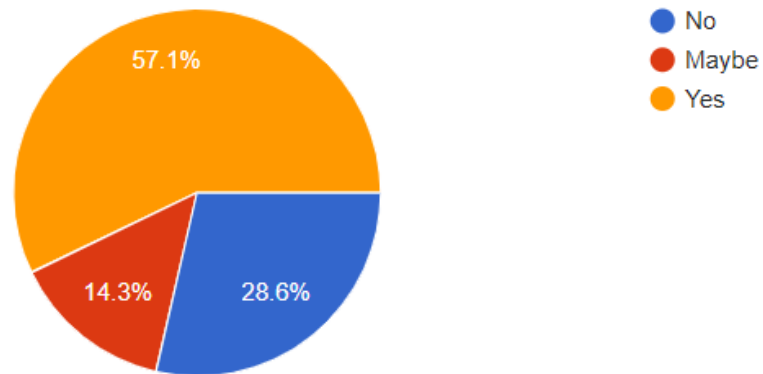
7 responses



*Figure 22 Family information about diabetes and survey on use of contact form*

Do you want to Have Type I diabetes check-up for free using web application?

7 responses



Want to use health Care application in a future?

7 responses

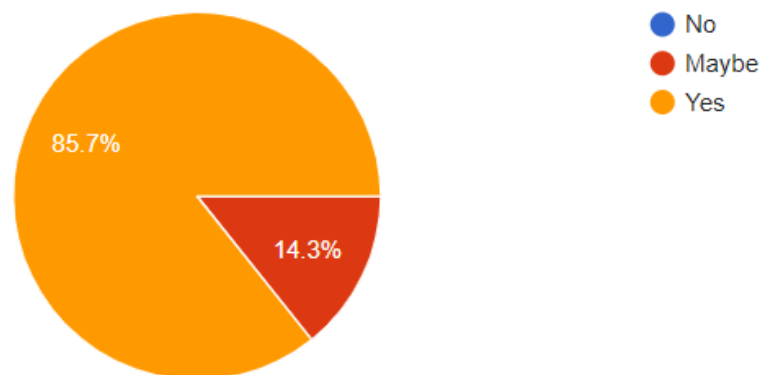
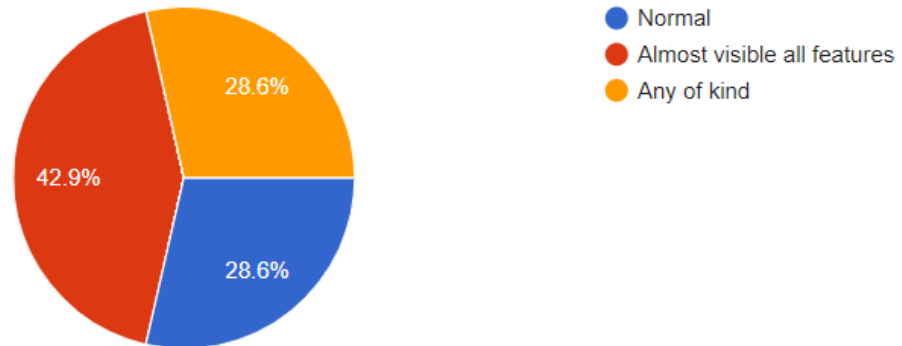


Figure 23 Excitements to use healthcare application in future

What types of User Interface you like to have ?

7 responses



#### **3.1.1.2.2. BPM (Business Process Management):**

It is a method how company creates, edits and analyzes the predictable processes that makes the core of its business.

In terms of this application, it will be completely free of cost to its clients so that they can take full advantage of it. The financial income of this applications will be based on advertisements due to its online availability.

#### **3.1.1.2.3. Functional Decomposition Diagram:**

This diagram is responsible for including overall functions and sub functions of a project needed to achieve the overall objective.

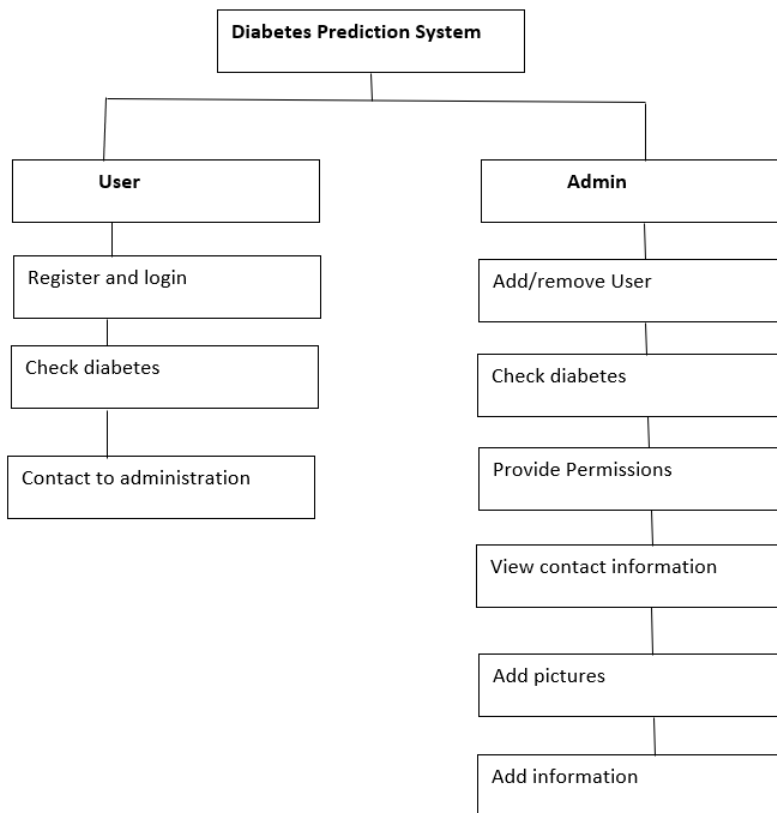


Figure 24 FDD diagram

### 3.1.1.3. Requirement Document:

#### 3.1.1.3.1. Functional requirements:

A Functional Requirement is a portrayal of the organization that an application must offer. It portrays a product framework or its fragment. The limit is just commitments to the framework, its lead, and yields. It will in general be a figuring, data control, business process, customer correspondence, or whatever other explicit usefulness that portrays what work a product is presumably going to perform. Functional Requirements are furthermore called Functional Specification. (Guru99, 2020)

Functional Requirements are categorized into two types i.e. user requirements and system requirements.



**User requirements are:**

- User must able to view some contents without registration to a system.
- User must be able to register to a system
- User Must be able to do login to the system
- User Must be able to view pictures and contents uploaded by an admin
- User must be able to fill out contact form
- Admin must be able to view contact information in an admin panel
- Admin must able to check diabetes
- Admin must able to add and delete user
- Admin must able to delete contact information

**System Requirements are:**

- System must give user interface to register.
- System must give user interface to login
- System must provide user interface with and without login to fill contact form and contact to the system
- System must allow to view some additional contents after login
- System must provide admin panels to perform various tasks by an admin
- System must store all data about user registration, and other several data given by admin and user.

**3.1.1.3.2. Non-Functional Requirements:**

It is a specification that describes the systems operation abilities and constraints that enhances its functionality. These are responsible to make system efficient, secure, reliable, etc.

Some important non-functional requirements of the system are as follows:

- Performance:  
System should have to give its better performance for several tasks and works

- Authentication:

Authentication system should be implemented so that only authorized people will have accessibility to the core part of a system.

- Reliability

System should give better result i.e. possibly having no bugs during runtime. Otherwise, people will not rely on it and limits its usage.

- Error Handling

System should handle possible upcoming errors in a system likewise patterns of phone numbers, email ID, etc.

- Security

System should be secured so that non-authorized and third parties will not be able to harm the system.

- Storage:

Storage capacity of a system should be made high so that it will not create any problem regard storage when there is possibility of data being high.

- Availability:

System should be available anytime whenever and wherever it needs.

### **3.1.1.3.3. Usability requirements:**

These requirements are reported desires and particulars intended to guarantee that an item, administration, administration or condition is easy to use.

It includes different factors such as:

- Accessibility

The system could be made accessible to various parts and from different devices as possible.

- User friendly

User interfaces could be made favorable so that they may felt easy to use the system. Pages could be made responsive thus can be used by devices having varied media screens.

- Performance

System could be increased performance wise.

All these usability requirements play effective role for product efficiency.

### **3.1.1.4. Architecture and Design:**

This step of the iterative model deals with designs. In this progression, framework engineering also plans to occur. Here, the client got the real plan of the framework and real engineering of the framework. Undertaking achievement, buyer fulfillment, and framework quality are likewise evolved when planning the framework. (Alghamdi, et al., 2016)

#### **3.1.1.4.1. Process Design (modeling document):**

It represents some earlier approaches done for the designing of any kinds of software. Process design helps in explaining different processes involved in development of web application. It defines how one function interacts with another and

how should the system made so that it achieves flow of information from one model to another model and how they communicate each other. Some process designs that describes logical representation of a software are:

#### 3.1.1.4.1.1. Use case diagram:

This diagram helps to summarize some of the relationship between use cases, actors and systems. Use case diagram for this system is given below:

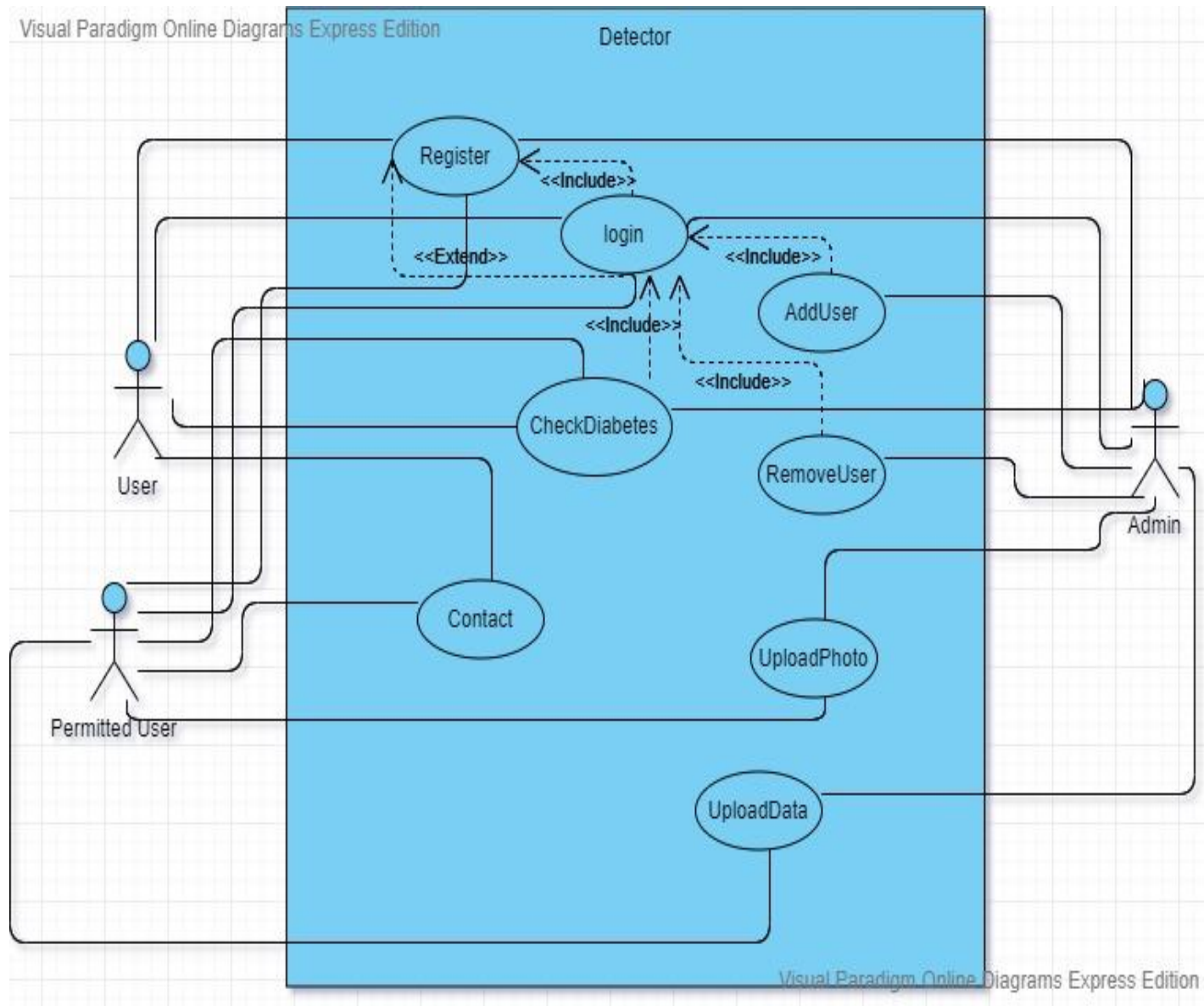


Figure 25 Use Case Diagram

### 3.1.1.4.1.2. Activity diagram:

It is a propelled type of flow chart that portrays the dynamic parts of a framework. It helps in displaying a stream starting with one movement then onto the next. Activity diagram for this system is given below:

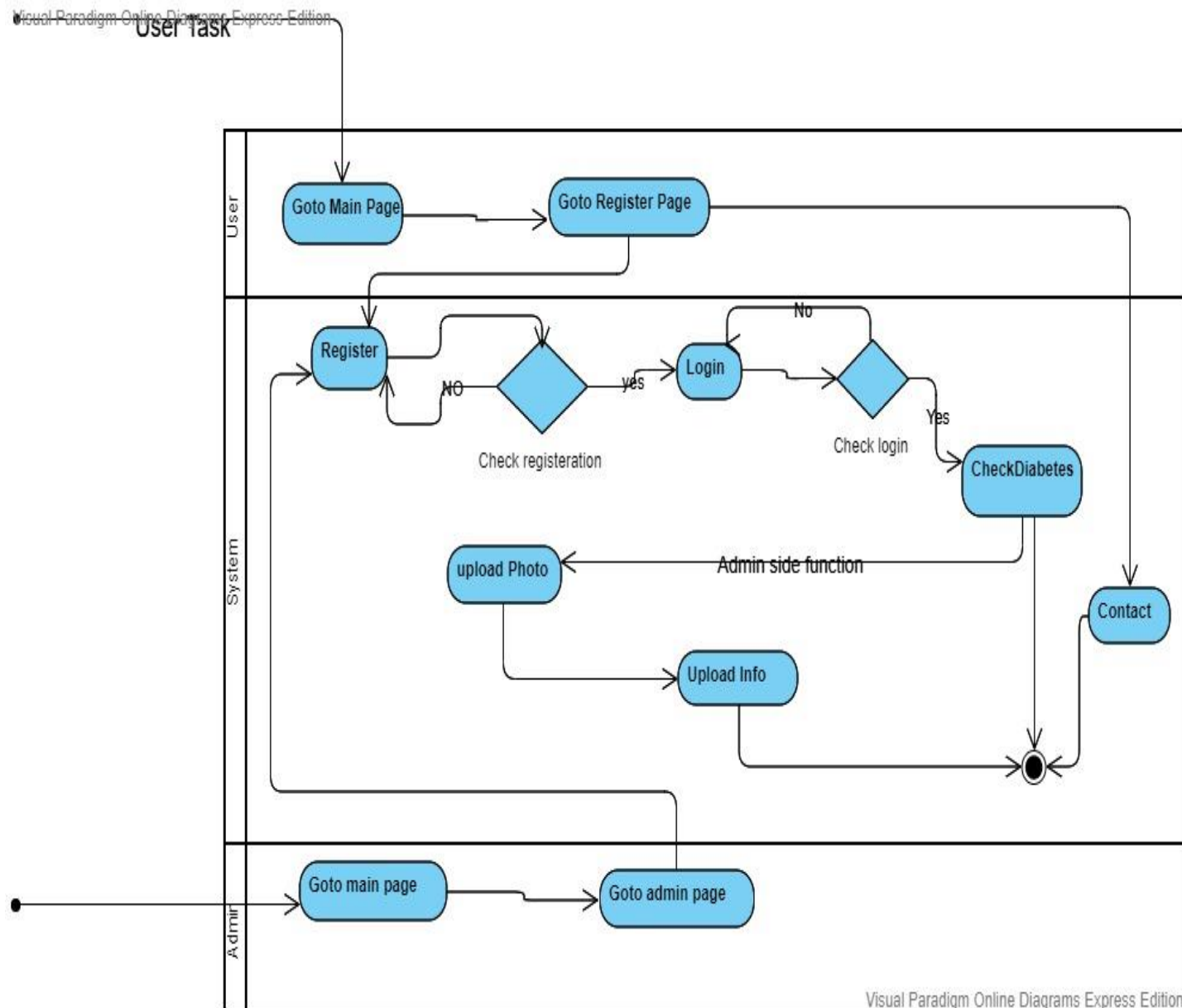


Figure 26 Activity diagram

### **3.1.1.4.1.3. Sequence diagram:**

This outline shows the delineates connection between objects in a consecutive request. It depicts how and in what request the items in a framework work. Sequence diagrams for this project are given below:

### 3.1.1.4.1.3.1. For admin:

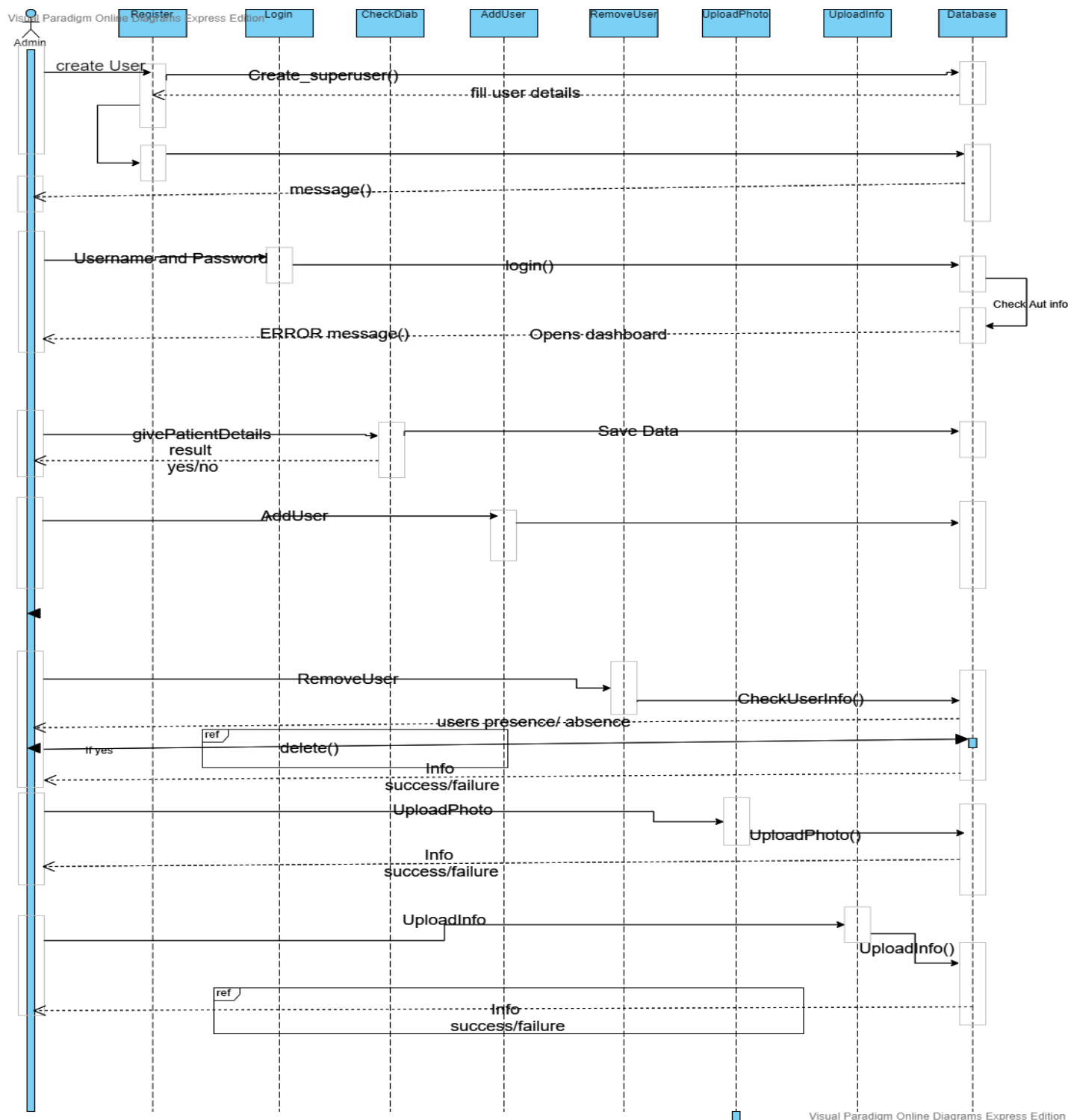


Figure 27 Sequence diagram for an admin

### 3.1.1.4.1.3.2. For user:

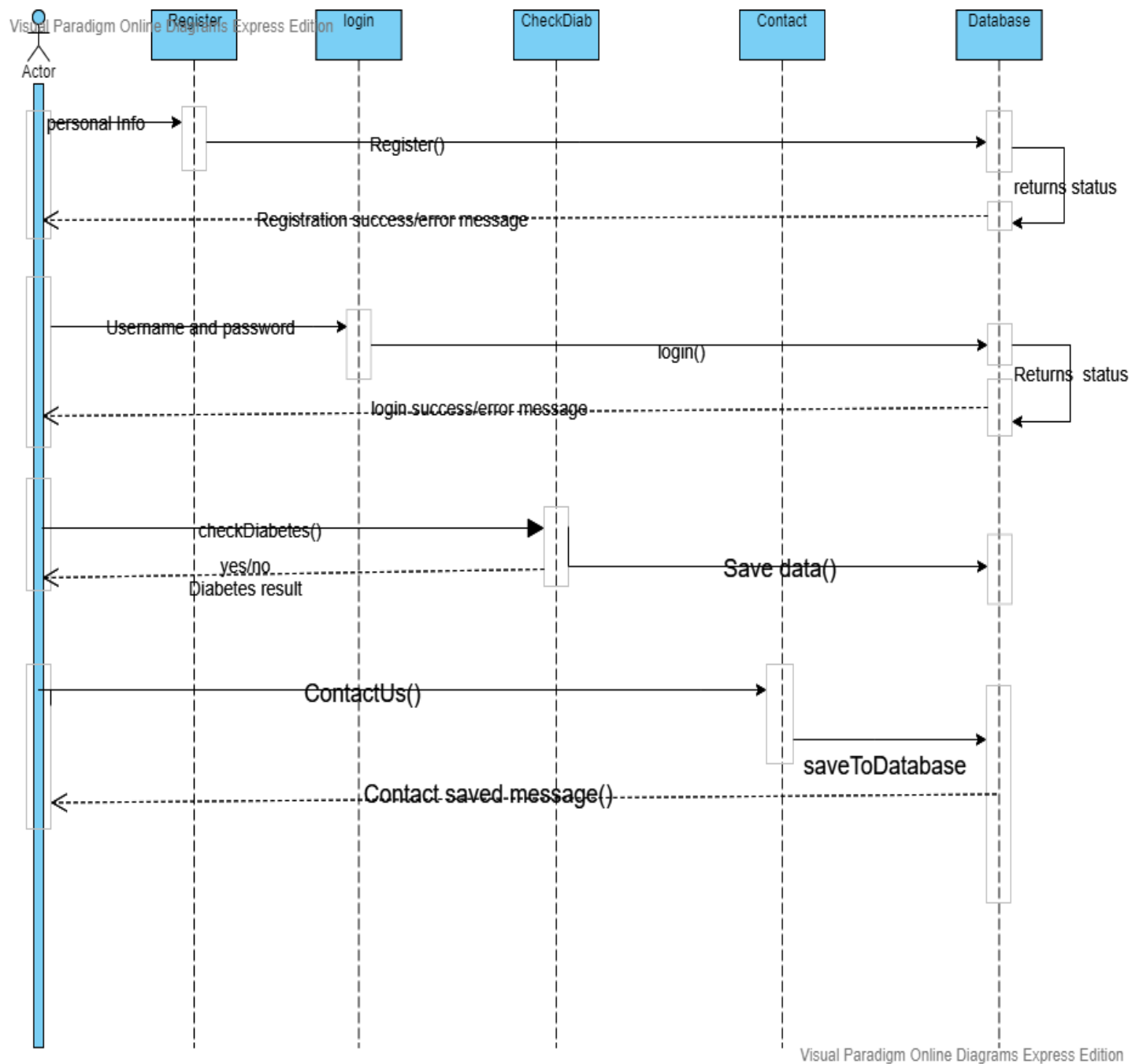


Figure 28 Sequence diagram for a user



### 3.1.1.4.1.4. Class diagram:

It is a kind of static structure program that depicts the structure of the framework by demonstrating the framework's classes, their properties, activities, and relationship among the items utilized in a framework.

The relationship among the system are explained on various ways such as aggregation, composition, associations, dependencies, etc. (Paradism, 2020)

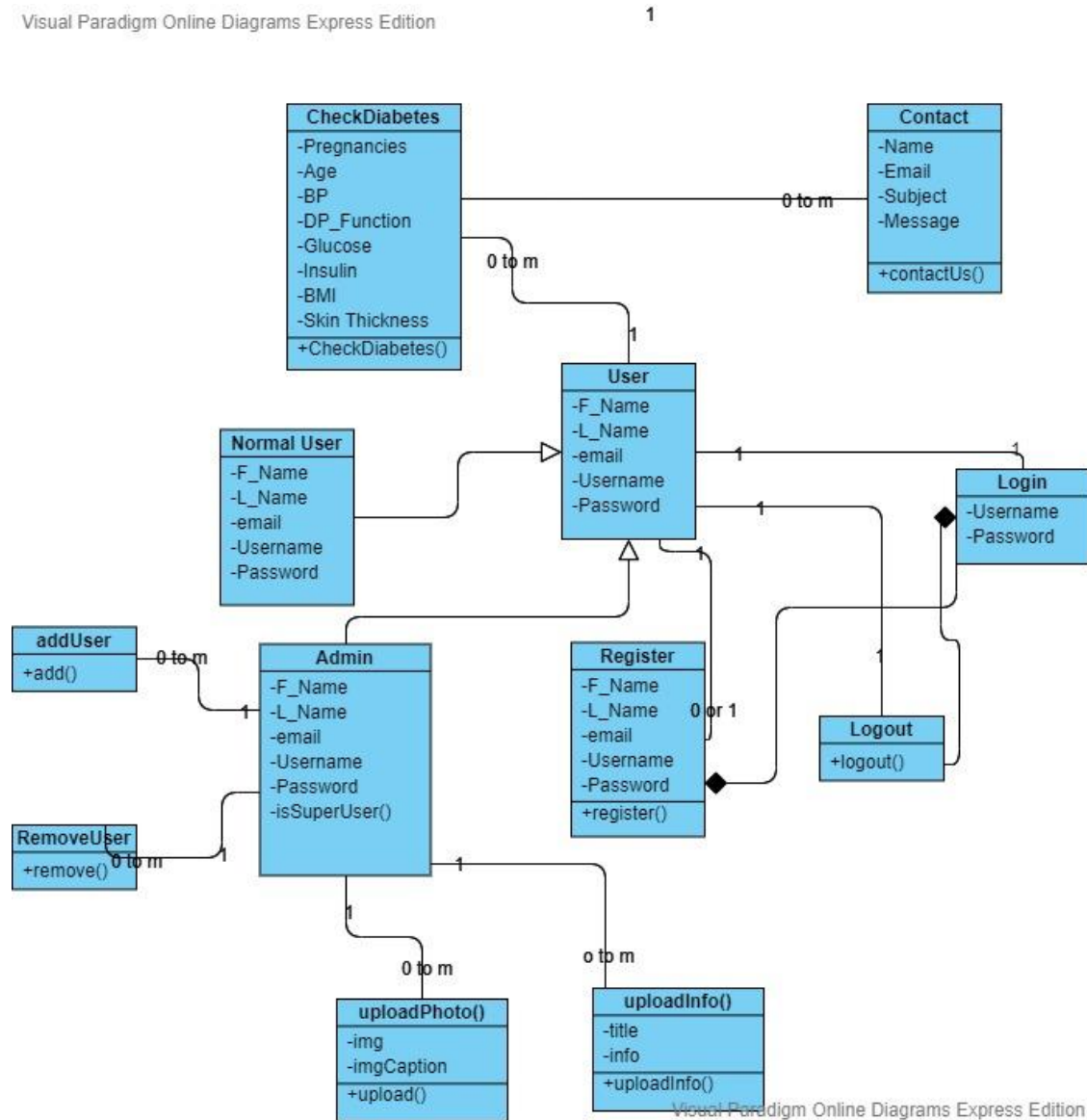


Figure 29 Class Diagram

### 3.1.1.4.1.5. State transition diagram:

It depicts all the states that an object can have, the occasions under which an object changes state. Some state transition diagrams for an application are as follows:

#### 3.1.1.4.1.5.1. Fill contact Form:

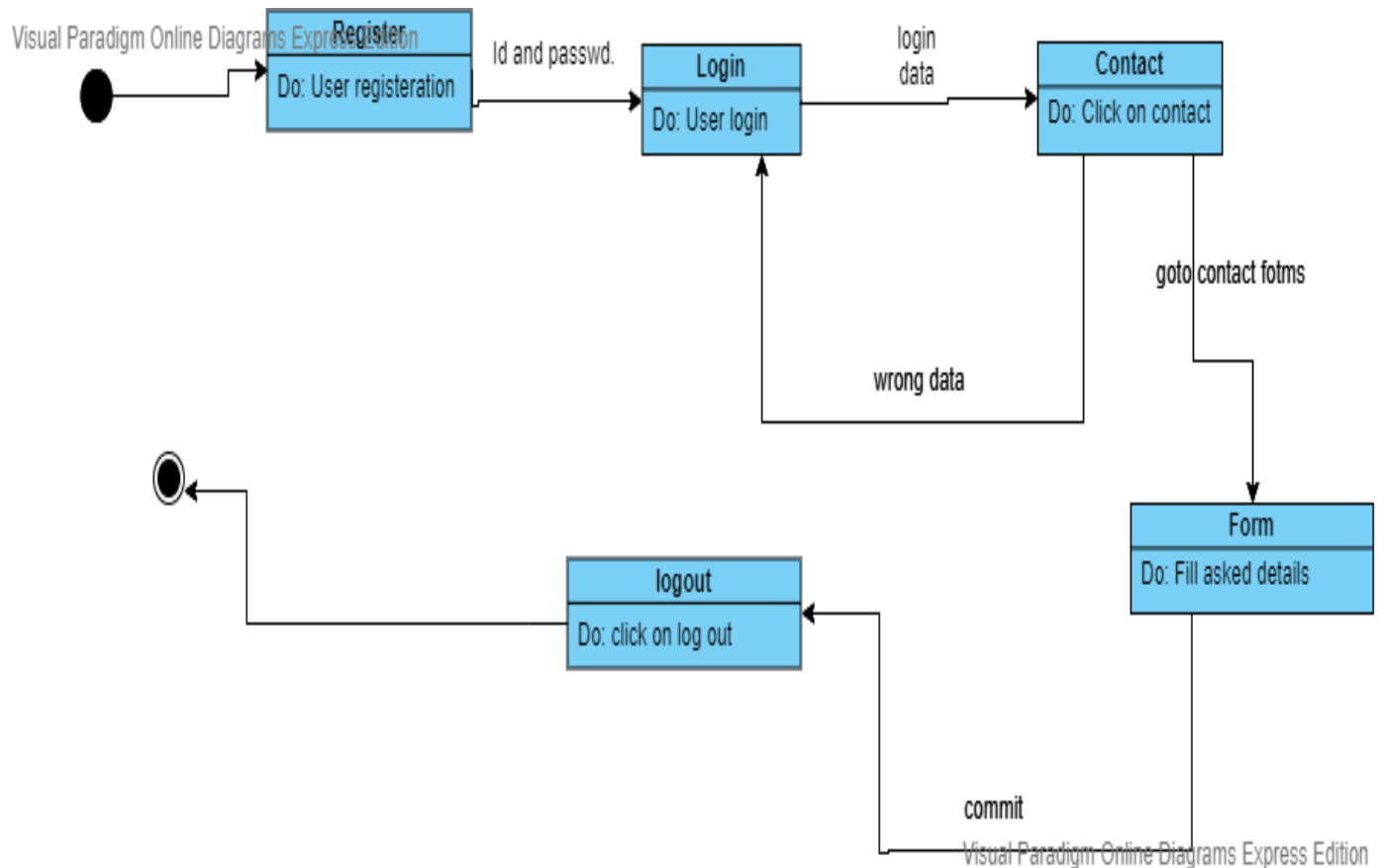


Figure 30 Fill Contact Form

### 3.1.1.4.1.5.2. View Uploaded Images on gallery:

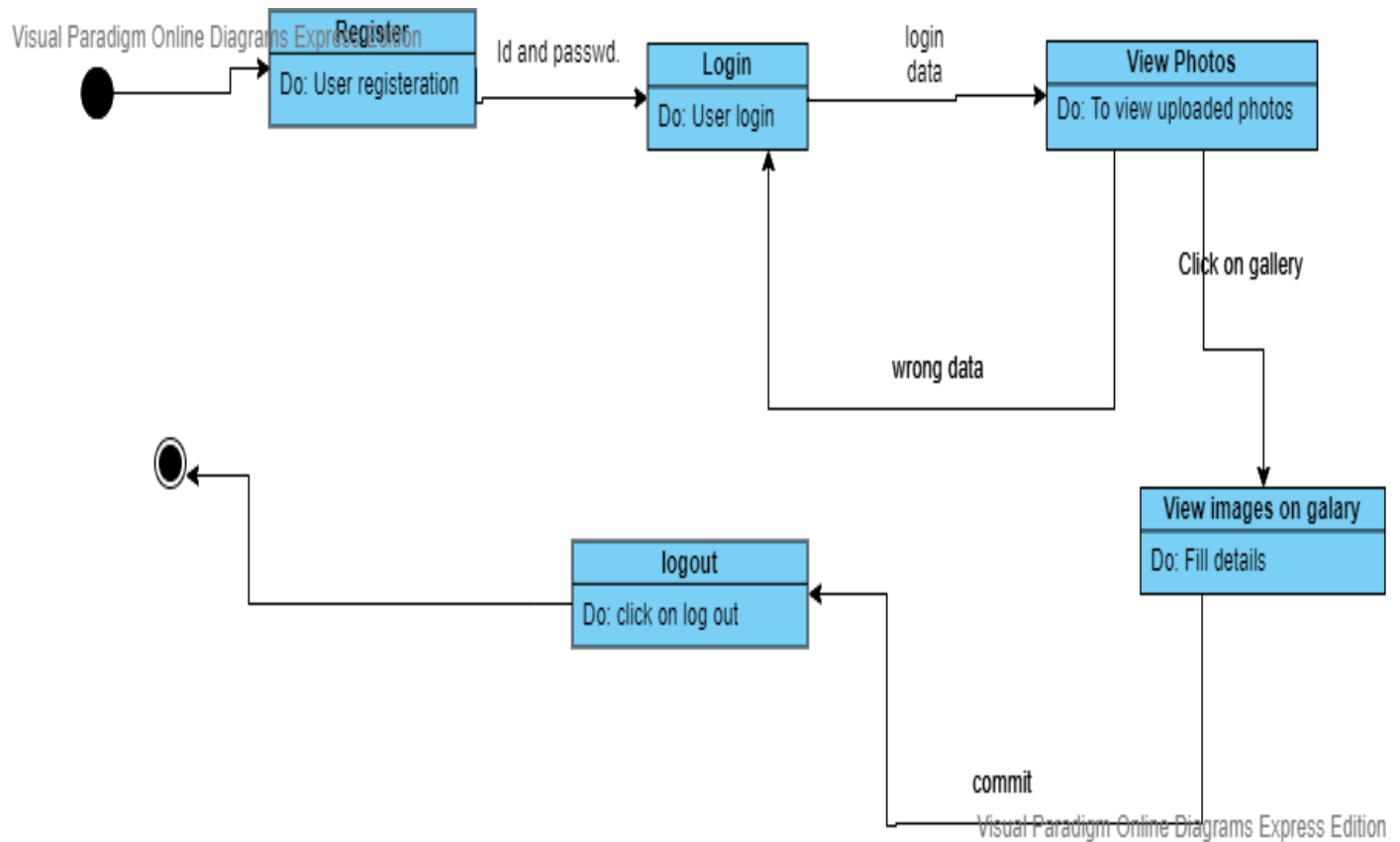


Figure 31 View Uploaded Photos

### 3.1.1.4.1.5.3. Check Diabetes:

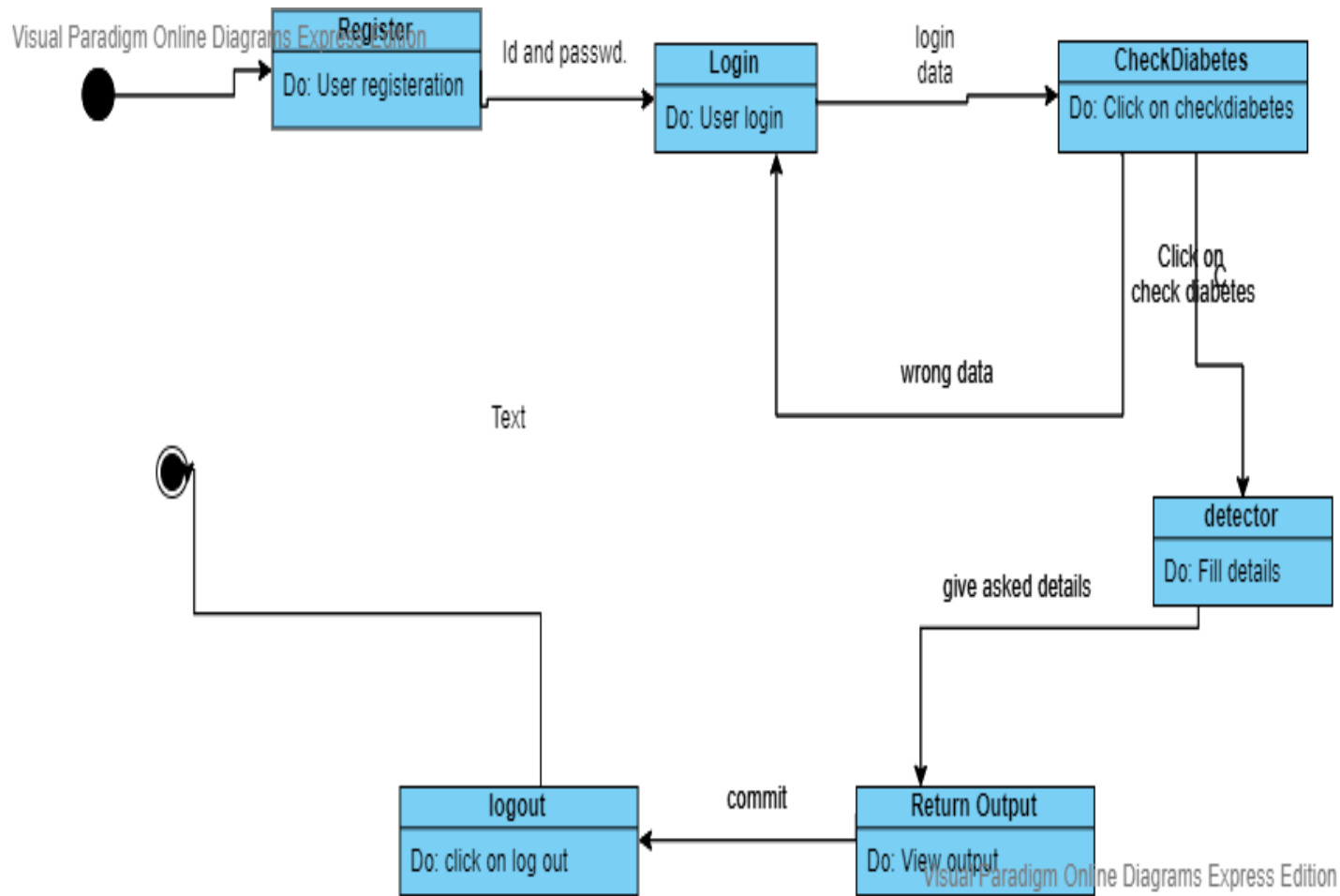


Figure 32 Check Diabetes

#### **3.1.1.4.1.6. Data Dictionary:**

- CheckDiabetes = Pregnancies + Age + BP + DP Function + Glucose + Insulin + BMI+

Skin thickness

- User =PK: ID + F\_Name + L\_Name + email+ username + password
- Contact= Name + Email + subject + Message
- Register = F\_Name + L\_Name + Email + Username + Password + confirm\_Password
- Login= username+ password
- UploadPhoto= img + imgCaption
- UploadInfo= title + info

#### **3.1.1.4.1.7. Entity-Relationship Diagram (ERD):**

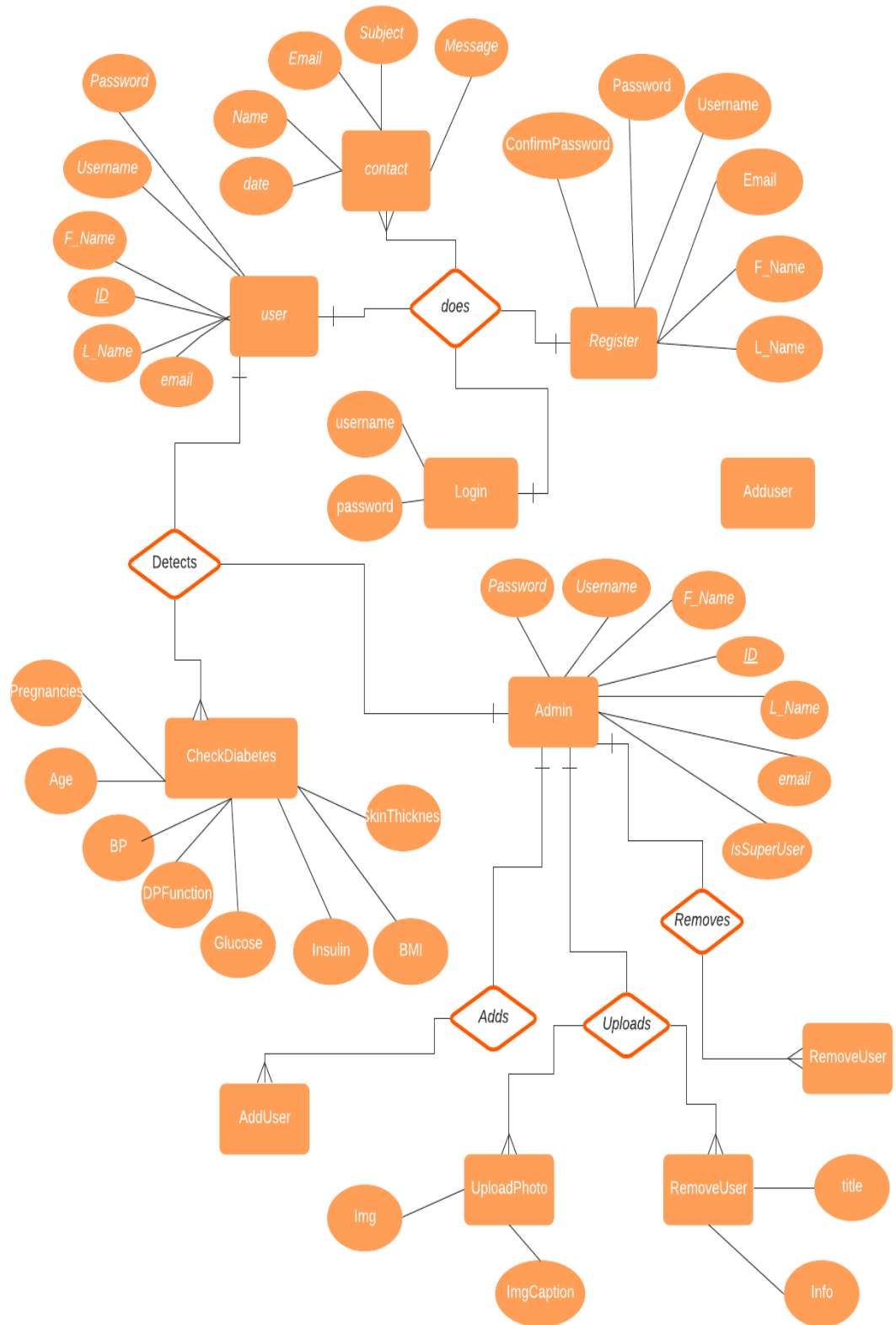


Figure 33 Entity Relationship Diagram

#### **3.1.1.5. Coding:**

It is the core step of the SDLC model. All the documented and designed system architectures are implemented in this phase. This step leads to the programming of the model and implementation of all the proposed system requirements as far as possible. Coding is done in this step. (Javatpoint, 2018)

#### **3.1.1.6. Testing and Maintenance:**

It is the process of checking whether an implemented system is working properly with proper delivery or not. All the internal and external working mechanisms of the system are checked in this step of the SDLC methodology. Some testing techniques include black-box testing, white-box testing, etc.

After the development of a system, its maintenance is necessary to make a system free of possible bugs and its update processes.

### **3.2. System Architectures:**

The web is one of the most useful, easy, and reliable ways of browsing through the internet.

A working mechanism of a web framework is given below:

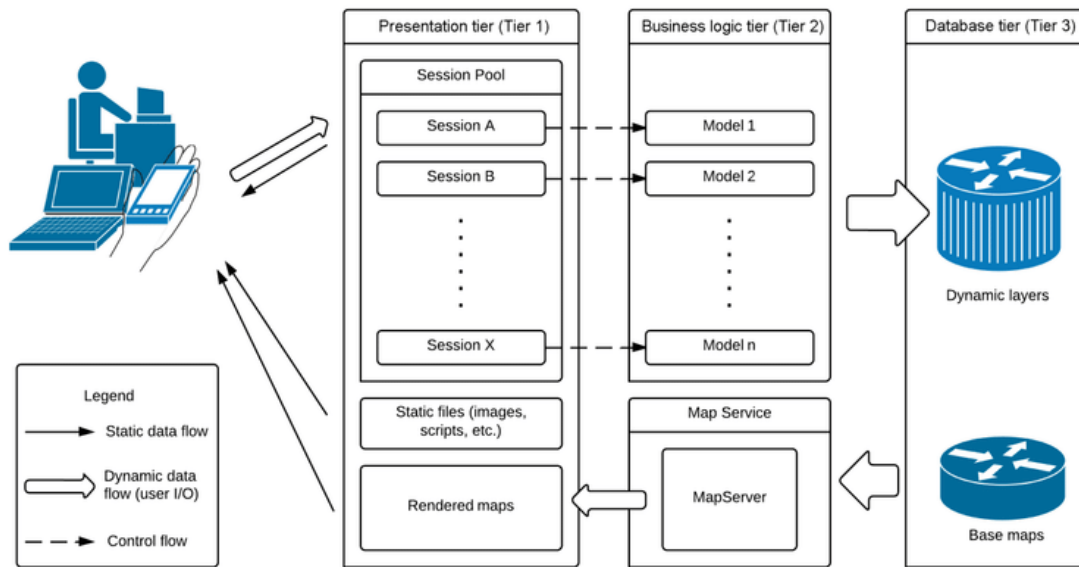


Figure 34 Client server (3-tier) architecture

This diagram is based on a client-server architecture. In the above figure, there will be a client who uses the web as a service, requests browsing using different web browsers. Internet and data are the media to use the web browser to communicate with the webserver. When a client requests a web server for a specific task with the help of browsing, it tries to match the task with the business logic, searches for the matched component in storage, and helps to retrieve the information. (altexsoft, 2019)

The different tiers present in the system architecture are given below:

### Client-tier:

It is also known as presentation tier which acts as the front layer that deals with clients. Clients are only authorized to access presentation layer where they get an availability of easily usable user interfaces and user experiences. Different functions such as interaction with their registration and login, form fill-ups, diabetes detection



through an API, Diabetes detection in general, availability of viewing and downloading contents, etc. which are included in my system are included in presentation tiers.

### **Application tier:**

It is connectivity layer lies between the model and the client layers which is responsible to build business logic for proper interaction and operations between client and database. Control of performances, restrictions and permissions to client, usability, authentication logics, content managements, etc. all were included in this tire of architecture.

### **Database tier:**

It is used to gather all data and information and store it. All the operations related to data storage, processing, querying, etc. are done in this tier. It always plays a prime role interacting with the business task to perform any kinds of dynamic performances. Business logic is responsible for data storage and its retrieval in every frameworks.

## **3.3. Tools and techniques:**

### **Python:**

It is widely used high-level object-oriented programming language which seems today in high trend due to its simplicity, readability, rich libraries, frameworks, and as well as easily writable. It is easy to use and implement because of large library resources available in an enormous amount. Nowadays, it is widely used to build web applications development, machine learning, data analysis, Desktop GUI, data science and visualization, CAD applications, Embedded applications, and so on.

Nowadays, we can see its use in a significant amount because of its simplicity and it is freely available. It is widely used in Artificial Intelligence and data science projects

because of its highly supported internal and external python libraries such as NumPy, pandas, matplotlib, seaborn, etc.

In my project, I have used python in most of the fields such as web development, data analysis, model building, data visualization, numerical computing, data manipulation, and many more.

### **NumPy:**

It stands for Numerical python. It is one of the center libraries for logical processing in python. It gives a superior multidimensional array object and tools for working with these arrays. It is an open-source venture which is offered allowed to everybody. It has capacities for working in the area of straight polynomial math, Fourier change, and matrices. (Johnson, n.d.)

NumPy partially returns in python and must of the parts that require fast computation are written in c or c++. The main purpose to use NumPy is to provide an array object which is up to 50 times faster than that of traditional python lists because of their storage at one continuous place in memory.

### **Sci-kit learn:**

The library is based upon the SciPy (logical python). It gives a scope of supervised and unsupervised learning algorithms through a good interface in python. The library is centered around modeling data.

Some famous models gave by sci-unit learn are:

Grouping, cross-validation, Datasets, dimensionality reduction, Ensemble strategies, Feature extraction, Feature selection, Parameter Tuning,

Supervised models, for example, naive Bayes, decision trees, SVM, and so on which of them have their usefulness. (Brownlee, 2014)

### **Matplotlib:**

It is the famous python library used to create multi-dimensional graphs and plots by using python scripts i.e. data visualization. It is most commonly used to translate complex data into digestible insights for a non-technical audience. I mostly have used the pyplot sub-library of matplotlib which provides accessibility to generate plots, histograms, power spectra, bar charts, error charts, scatter plots, etc.

### **Seaborn:**

It is a data visualization library dependent on matplotlib. It gives an elevated level component to draw appealing and informative factual diagrams.

### **Pandas:**

It is an open-source library in python which is made with the purpose of data manipulation and analysis. It was mainly used for data preparation, loading, manipulation, modeling, and analysis.

A few elements of pandas are:

- Tools for stacking data into in-memory data objects from various document positions.
- Data arrangement and coordinated treatment of missing data.
- Insertion and erasure of sections from the information structure.
- Data consolidating and join

- Different tasks in huge data collections, for example, Label-based cutting, ordering, and sub-setting, and so on. (Tutorialspoint, 2020)

### **SQLite:**

It is the most deployed SQL database engine in the world. It is a software library that actualizes an independent, serverless value-based SQL database engine. It is used here for database storage.

## **3.4. System Requirement:**

### **3.4.1. Hardware Requirement:**

PC with a feature of having RAM = 4 GB/8 GB or higher

Storage= 30 GB or higher

### **3.4.2. Software Requirement:**

OS: Windows, Linux

IDE: Jupyter notebook, PyCharm, sublime text

Programming language: python, JavaScript

Framework: Django

Library: Numpy, Sci-kit learn, jQuery, Pandas, Matplotlib, Seaborn,  
etc.

Design tool: Photoshop

Web design: HTML/CSS

### 3.5. Implementation:

#### 3.5.1. Data Collection:

The dataset I am using in this application is taken from the hospital of Frankfurt Germany. As it is a health issue, I tried well to collect data from Nepalese hospitals but they denied to provide because of their legal reason. Hence, as per the proposal to be fulfilled, I come up with the solution to use a real-time dataset which is taken from a hospital of Germany and a reliable source. It consists of 8 different factors i.e. pregnancies, glucose, BP, skin-thickness, insulin, BMI, Diabetes Pedigree Function, and age. All these are some general factors taken so that we can predict diabetes in upcoming patients taking all these data into a dataset as a guideline. All these data are available in CSV format.

	A	B	C	D	E	F	G	H	I
1	pregnancies	glucose	BP	SkinThickness	insulin	BMI	DPFunction	age	outcome
2	2	138	62	35	0	33.6	0.127	47	1
3	0	84	82	31	125	38.2	0.233	23	0
4	0	145	0	0	0	44.2	0.63	31	1
5	0	135	68	42	250	42.3	0.365	24	1
6	1	139	62	41	480	40.7	0.536	21	0
7	0	173	78	32	265	46.5	1.159	58	0
8	4	99	72	17	0	25.6	0.294	28	0
9	8	194	80	0	0	26.1	0.551	67	0
10	2	83	65	28	66	36.8	0.629	24	0
11	2	89	90	30	0	33.5	0.292	42	0
12	4	99	68	38	0	32.8	0.145	33	0
13	4	125	70	18	122	28.9	1.144	45	1
14	3	80	0	0	0	0	0.174	22	0

Figure 35 Early data existence

#### 3.5.2. Data Processing:

It includes many works to be done whether the data is useful or highly deviated. Different data cleaning and statistical techniques are used to get data in a proper format.

Some steps involved in the processing are as follows:

```
In [7]: data.describe()
```

```
Out[7]:
```

	pregnancies	glucose	BP	SkinThickness	insulin	BMI	DPFunction	age	outcome
count	2000.000000	2000.000000	2000.000000	2000.000000	2000.000000	2000.000000	2000.000000	2000.000000	2000.000000
mean	3.703500	121.182500	69.145500	20.935000	80.254000	32.193000	0.470930	33.090500	0.342000
std	3.306063	32.068636	19.188315	16.103243	111.180534	8.149901	0.323553	11.786423	0.474498
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.078000	21.000000	0.000000
25%	1.000000	99.000000	63.500000	0.000000	0.000000	27.375000	0.244000	24.000000	0.000000
50%	3.000000	117.000000	72.000000	23.000000	40.000000	32.300000	0.376000	29.000000	0.000000
75%	6.000000	141.000000	80.000000	32.000000	130.000000	36.800000	0.624000	40.000000	1.000000
max	17.000000	199.000000	122.000000	110.000000	744.000000	80.600000	2.420000	81.000000	1.000000

Figure 36 Described data

It describes many about the dataset. From this description, we can find their average, deviation, and quarterly ranged values. In some minimum values, the result seems to be 0 and as well as 0 to first quartile values. Insulin seems to be the value with a very high deviation. All these issues are needed to be solved.

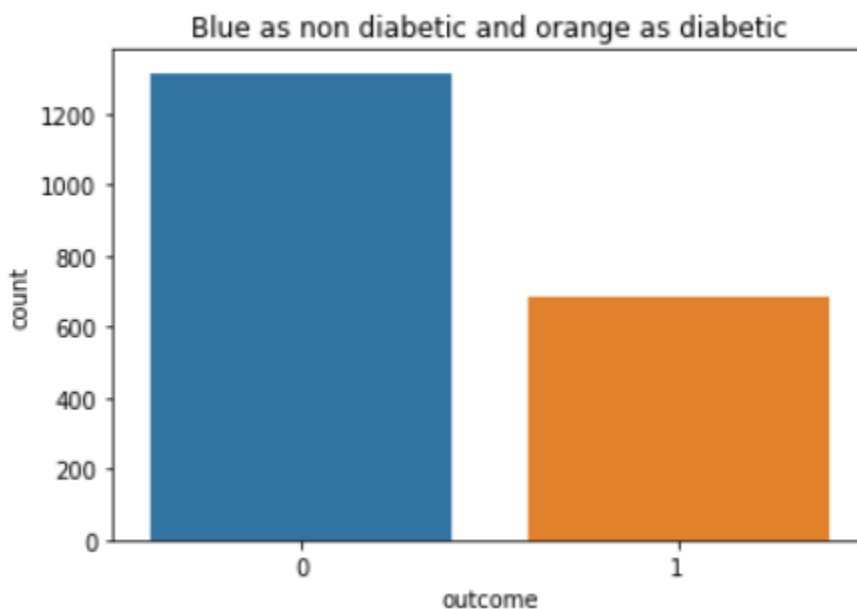


Figure 37 Visualization "YES" vs "NO"

This plot function is used to visualize the total number of patients whether diabetic or non-diabetic. Here 0 means non-diabetic and 1 means diabetic.

```
In [9]: data.hist(figsize = (10,10), bins=25)
```

```
Out[9]: array([[<matplotlib.axes._subplots.AxesSubplot object at 0x000001B78CD520C8>,
<matplotlib.axes._subplots.AxesSubplot object at 0x000001B78CE3AD08>,
<matplotlib.axes._subplots.AxesSubplot object at 0x000001B78CE76588>],
[<matplotlib.axes._subplots.AxesSubplot object at 0x000001B78CEAD6C8>,
<matplotlib.axes._subplots.AxesSubplot object at 0x000001B78CEE7808>,
<matplotlib.axes._subplots.AxesSubplot object at 0x000001B78CF1F888>],
[<matplotlib.axes._subplots.AxesSubplot object at 0x000001B78CF589C8>,
<matplotlib.axes._subplots.AxesSubplot object at 0x000001B78CF90B08>,
<matplotlib.axes._subplots.AxesSubplot object at 0x000001B78CFAC48>]],
dtype=object)
```

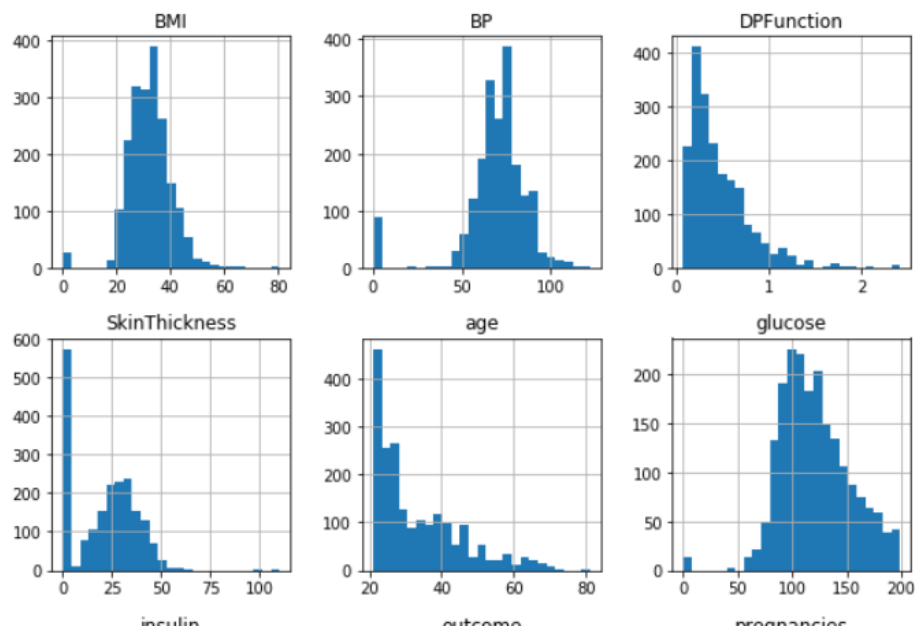


Figure 38 Histogram Plot

The above visualization is a histogram which has shown the distribution of data in a dataset. As the deviation is more in data, more the model seems to be not working properly. Hence, we need to clean the data properly so that we can create a low deviation in a dataset. The statistical theory should need to solve these problems. It is possible to find data in whether left or right skewness but extra tail should be removed e.g. like of glucose.

```
In [12]: # To check the missing (0) values which are in this dataset which needed to be
#cleaned to get better result

print("number of missing pregnancies i.e. having (0) are : {}".format(len(data
print("number of missing glucose i.e. having (0) are : {}".format(len(dataset.
print("number of missing BP i.e. having (0) are : {}".format(len(dataset.loc[d
print("number of missing SkinThickness i.e. having (0) are : {}".format(len(dat
print("number of missing insulin i.e. having (0) are : {}".format(len(dataset.l
print("number of missing BMI i.e. having (0) are : {}".format(len(dataset.loc[d
print("number of missing DiabetesPedigreeFunction i.e. having (0) are : {}".for
print("number of missing age i.e. having (0) are : {}".format(len(dataset.loc[d

number of missing pregnancies i.e. having (0) are : 301
number of missing glucose i.e. having (0) are : 13
number of missing BP i.e. having (0) are : 90
number of missing SkinThickness i.e. having (0) are : 573
number of missing insulin i.e. having (0) are : 956
number of missing BMI i.e. having (0) are : 28
number of missing DiabetesPedigreeFunction i.e. having (0) are : 0
number of missing age i.e. having (0) are : 0
```

Figure 39 Handling Missing Values

The above function is printing the total number of records which are having an input of zero values. Some parameters i.e. glucose, Skin thickness, BP, insulin, etc. are given wrong input. Zero value is a wrong value as per the medical terms which are whether missing values or wrong inputs. All these first needed to be converted into **np.nan** and have to be filled with either mean or median as per the property of data.



```
: sn.pairplot(dataset, diag_kind='kde')
: <seaborn.axisgrid.PairGrid at 0x1b78d527988>
```

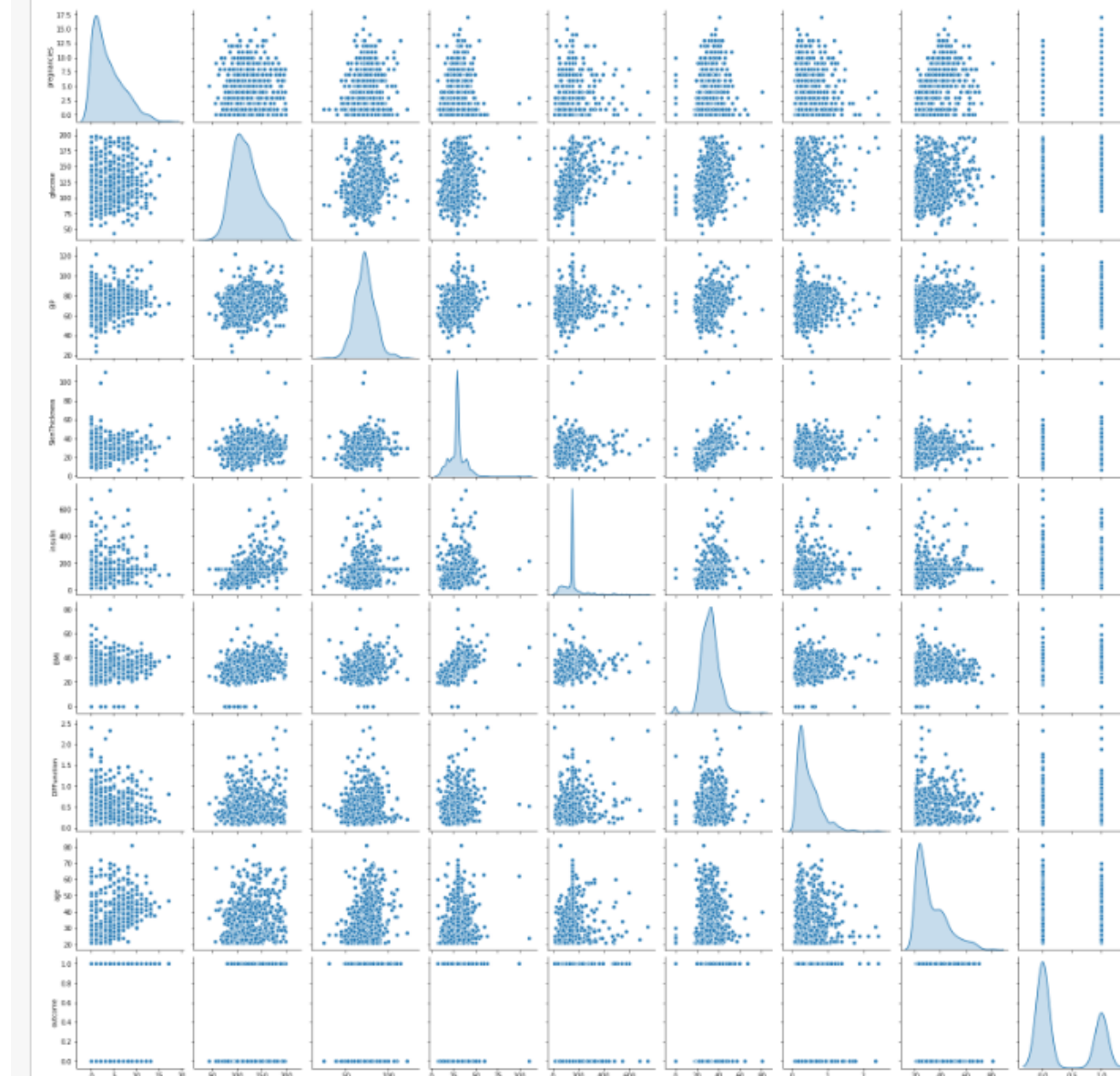


Figure 40 Pair plot

It is a pair plot that provides the correlation of different factors in a dataset. The middle one provides a graph that defines the distribution of data.

To be the best dataset to build a model, two different factors should have no relationship. It doesn't exist in real-time hence we have to minimize the correlation value and should make near to zero.

```
23]: co_relations=dataset.corr()
      print(co_relations)
```

pregnancies	1.000000	0.121537	0.198110	0.087605	0.054230
glucose	0.121537	1.000000	0.199498	0.208309	0.406556
BP	0.198110	0.199498	1.000000	0.202711	0.073282
SkinThickness	0.087605	0.208309	0.202711	1.000000	0.179486
insulin	0.054230	0.406556	0.073282	0.179486	1.000000
BMI	0.019475	0.243739	0.228055	0.465059	0.170183
DPFunction	-0.025453	0.124141	0.012466	0.091822	0.096155
age	0.539457	0.259805	0.323659	0.133270	0.089810
outcome	0.224437	0.488020	0.174184	0.205527	0.207696

	BMI	DPFunction	age	outcome
pregnancies	0.019475	-0.025453	0.539457	0.224437
glucose	0.243739	0.124141	0.259805	0.488020
BP	0.228055	0.012466	0.323659	0.174184
SkinThickness	0.465059	0.091822	0.133270	0.205527
insulin	0.170183	0.096155	0.089810	0.207696
BMI	1.000000	0.125719	0.038987	0.276726
DPFunction	0.125719	1.000000	0.026569	0.155459
age	0.038987	0.026569	1.000000	0.236509
outcome	0.276726	0.155459	0.236509	1.000000

Figure 41 Corelation

It seems that the correlations between each factor seem to be minimum as possible so that we further can use this dataset to make a model splitting it to training and testing sets.

### 3.5.3. Data splitting:

It is the process of dividing the data from the dataset into two different parts i.e. training set and testing set. Trained data is fitted in a model and the testing dataset is used over the trained dataset i.e. to examine the performance of the built model. We can visualize the accuracy and other different measures of perfection of a system after comparing the training and testing set.

```

from sklearn.model_selection import train_test_split
featured_columns = ['pregnancies', 'glucose', 'BP', 'SkinThickness', 'insulin', 'BMI', 'DPFunction', 'age']
predicted_class = ['outcome']

X = dataset[featured_columns].values
y = dataset[predicted_class].values

X_train, X_test, y_train, y_test = train_test_split(X, y, train_size=0.6, random_state=0)

```

Figure 42 Training Test Split

### 3.5.4. Building model:

It is a process of fitting a dataset in an algorithm to make a prediction, classification, etc. and so on. It feeds data as an input and finds some patterns in data so that we can further use that generated pattern to solve problems in the future.

```

knn = KNeighborsClassifier(1)

model=knn.fit(X_train,y_train)
knn.score(X_test,y_test)
# knn.score(x_train,y_train)

```

Figure 43 Model building

### 3.6. Implementation (diabetes prediction system):

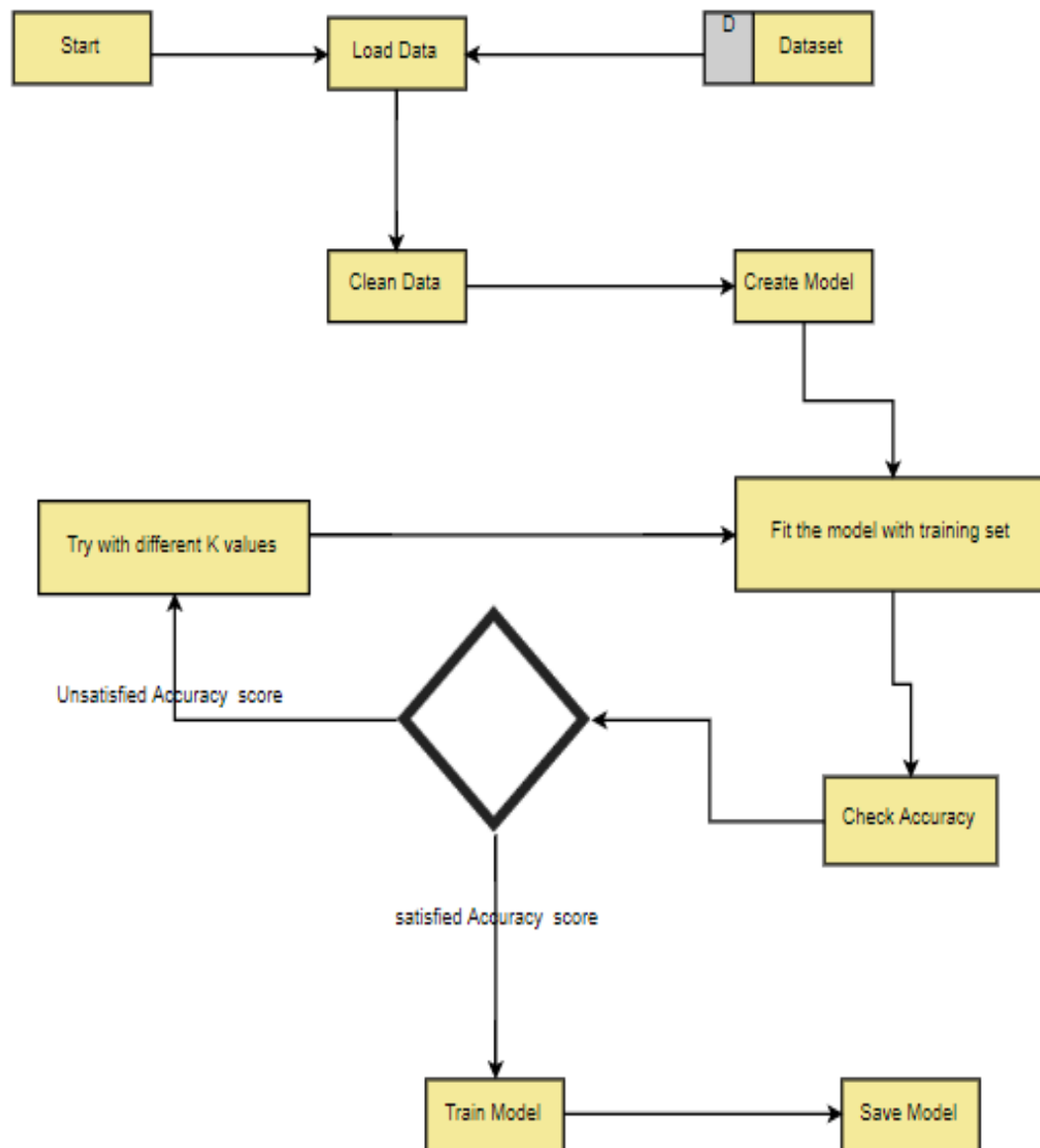


Figure 44 Implementation of Model

This web application is mainly designed to predict the status of diabetes of the patients taking an input of different parameters which are mainly responsible to carry some properties to detect diabetes. Appearance of changes in these factors instead of

normal value can be said to be the symptoms of having diabetes. This application will detect the diabetes in different patients feeding the past data fetched out from the hospital. The data fed to the model will set some guidance the data and upcoming all data will be classified as per the experience gained by the model from past data. Being of medical system, model is prepared to provide more value of recall to predict result as 1 instead of 0. It is because patient can check once in the hospital to confirm his possibility of diabetes either he is free of it. But if system shows result “NO” to the non-diabetic patient than patient life will be in a high risk.

It moreover functions to save upcoming data from the patients, keep the contact details and subjects regarding the issues.

In case of detecting diabetes in a patient, two different forms will be provided i.e. API form which is made so that it can further be accessed by any types of platforms and frameworks whereas in another, diabetes prediction system is integrated in web app itself. Users are allowed to use both kind of system where API system stores the data provided by users in a database for further research and analysis whereas web app integrated system only detects possibility of diabetes in patients whether keeping their diabetic information.

Some sort of functions is distinguished specifically to super users which will not be allowed by normal users. Such functions are to add contents and information, uploading images, removing of users, etc. will all be the function that will be used by superusers.

Diabetes Management system is an online system which can be said to be SaaS system that responds via internet to the patients.

This system consists of some security services as registration and login. Some important services such as displaying of important images and information, prediction of diabetes cannot be done without authorized registration and login. User can only get into the login system after his proper registration to the system.

Being sensitive system, few interaction functions are developed within an application but must of the function will be allowed to only admins as shown by use-case diagram.

This system is built under different computer languages such as python, JavaScript, html, CSS, SQL, bootstrap and so on. Django is the framework which is providing the base for the development of this system.

### 3.7. Wireframes:

#### 3.7.1. Home and About Us:

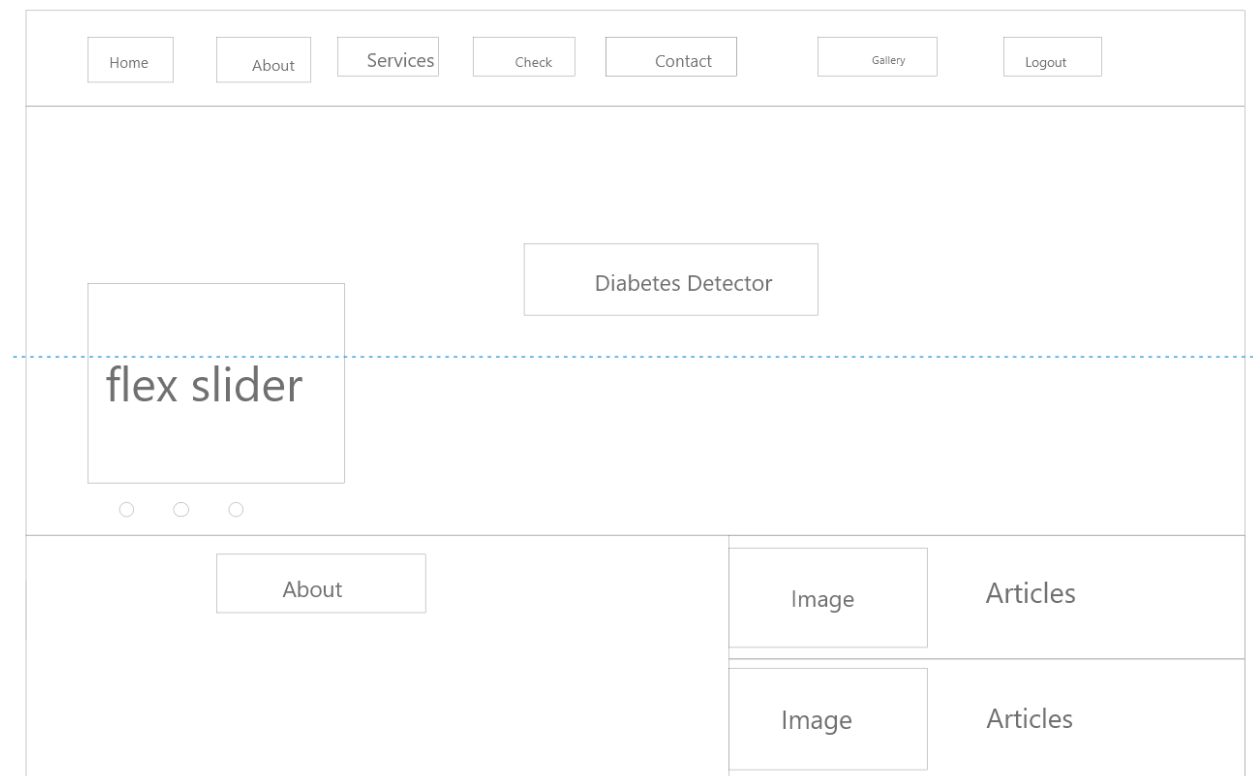


Figure 45 Home and About Us

### 3.7.2. Services and detect:

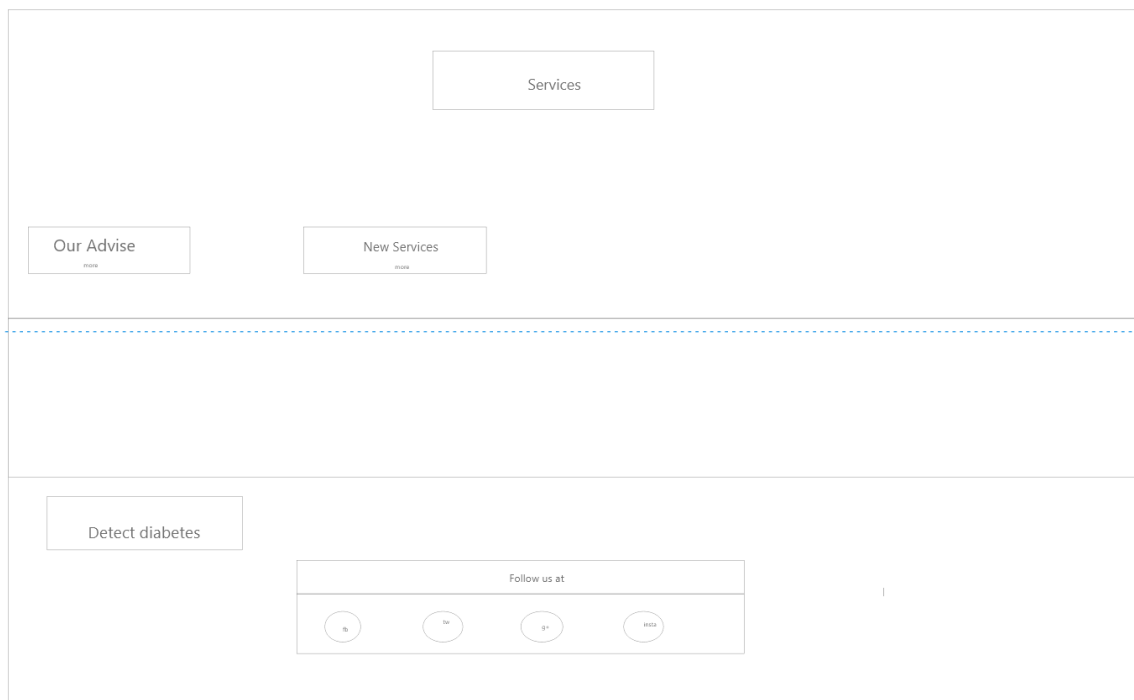


Figure 46 Services and detect

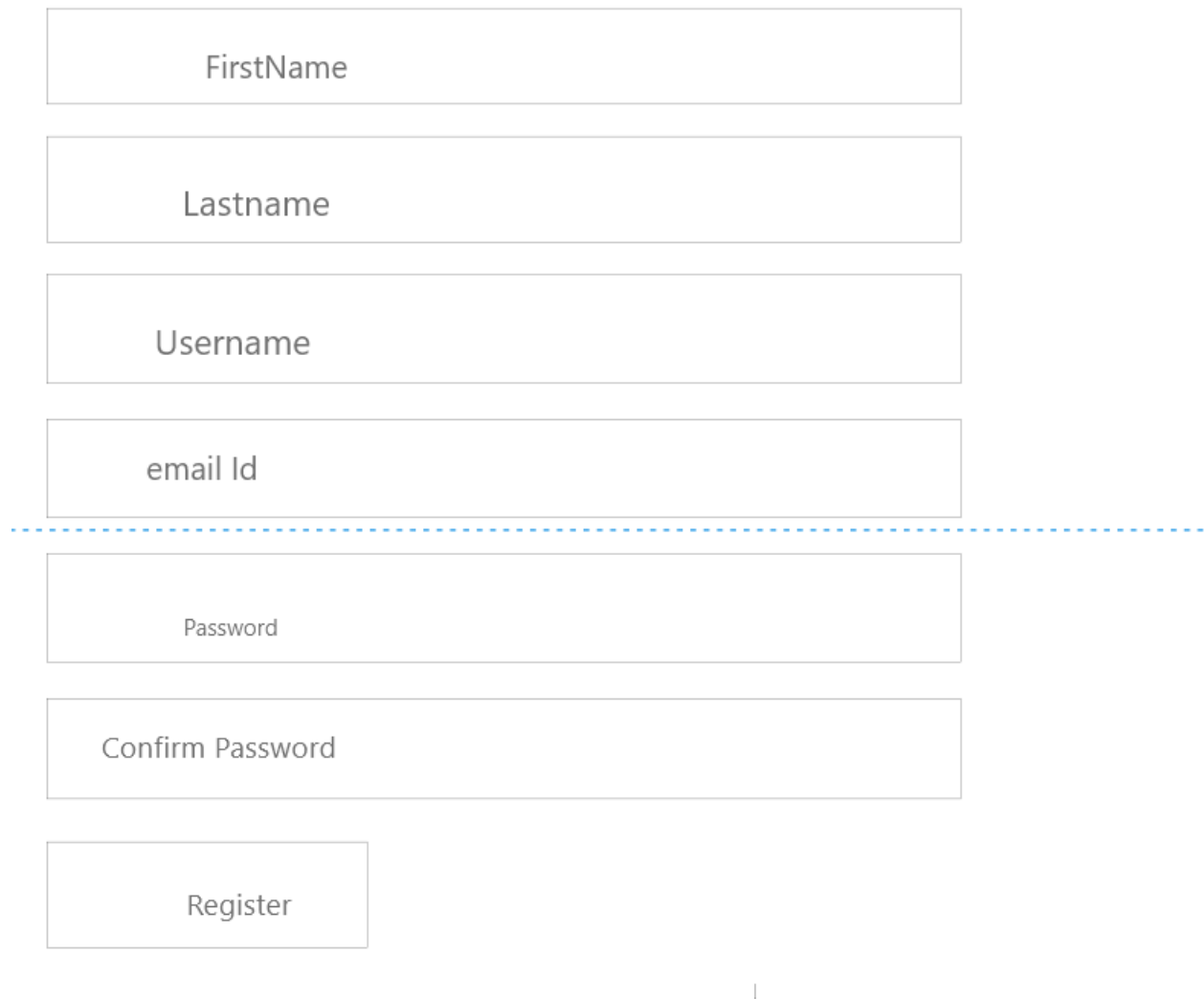
### 3.7.3. Contact:

The image shows a contact form layout within a light gray container. At the top center is a button labeled "Contact". Below it, on the left side, are three stacked input fields labeled "Name", "Email", and "Subject". To the right of these is a larger text area labeled "Message". A horizontal dashed blue line passes through the "Email" field and the "Message" area. Below the "Subject" field is a button labeled "Message Us". At the bottom of the container is a footer box containing the text "© 2020 All right reserved. Sandesh Lamsal".

*Figure 47 Contact*

#### **3.7.4. Register:**





The image shows a registration form with the following fields and elements:

- FirstName
- Lastname
- Username
- email Id

---

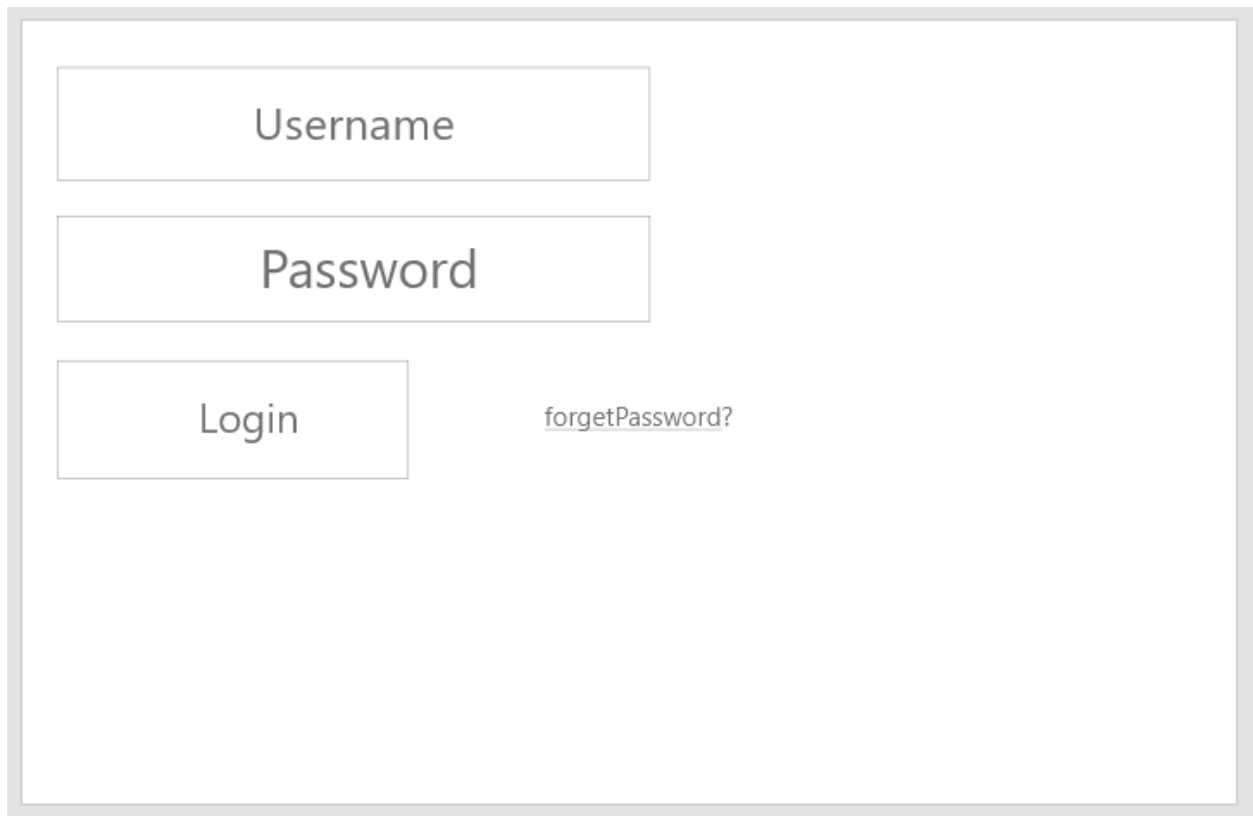
- Password
- Confirm Password

Register

|

*Figure 48 Register*

### 3.7.5. Logout:



A login form UI mockup enclosed in a light gray border. It features three input fields: 'Username' at the top, 'Password' in the middle, and 'Login' at the bottom left. To the right of the 'Login' button is a text link labeled 'forgetPassword?'.

Username

Password

Login

[forgetPassword?](#)

*Figure 49 Logout*

### **3.8. Page Views:**

### 3.8.1. Homepage:

When a user goes to access the system, first he goes towards the home page and can able to view some contents of this application. It shows limited number of pages and the user must first register and then do login to enter main dashboard of the system and could do prediction and accessible with essential contents.



Figure 50 Home Page I



Figure 51 Home Page II

### 3.8.2. Register:

Those users who are not superuser and not registered earlier are allowed to register to an application. This page can be found as a link to the left corner of an application.

Admin also can register to new users by using add user function provided to the admin panels.

The screenshot shows a web browser window with the address bar displaying "127.0.0.1:8000/accounts/register". The browser's tab bar shows three tabs: "Apps", "Free Ethical Hackin...", and "How to Crack a Pas...". The main content area displays a registration form with the following fields: "First Name", "Last Name", "Username", "Email ID", "Password", and "Confirm Password". Each field is represented by a light gray rectangular input box. Below these fields is a dark gray button labeled "Register".

Figure 52 Register

### 3.8.3. Login:

This page is developed for security purposes which provides page to login a user. User can only accessible to all functions assigned after his proper login.

A login form with two light blue input fields. The first field contains the text 'pujadear'. The second field contains a series of dots, indicating a masked password. Below the input fields, there is a dark grey button with the text 'login' and a purple underlined link that says 'forget password?'.

Figure 53 Login

#### 3.8.4. Database:

In this application, I have used SQLite as the database storage server in my project. It will record all the details of person, information given by the users and subscribers and all the contents added from the administrative site.

#### 3.9. Testing:

Testing means a practice to judge any implementations done to conclude that all the functional and non- functional requirements implemented in a system are working properly and either succeed or failed to meet the objectives of the proposed system. The testing must be done so that we again can reimplement the important modules in a system which helps to increase the systems efficiency and make the product more reliable, accurate and secure so any intruders and crackers do not harm to the system and costumers can fully rely on the system for their vital tasks.

There are many types of testing which are been practiced during the development of systems. Different types of testing in practices are:

### **3.9.1. White-Box Testing:**

In this kind of testing, the inner structure, plan, and usage of the thing are being tried. It is applicable for various degrees of programming testing like:

### **3.9.2. Unit testing:**

It is used for testing paths within a unit.

### **3.9.3. Integration testing:**

It is used for testing path between units.

### **3.9.4. Black-box testing:**

This project is using Django framework and SQLite to its system implementation. All the performances should be measured so that it will help users to find whether the system is working correctly or not.

#### **3.9.4.1. Test plan**

Test Case	Objectives
Case 1	User must register to the system
Case 2	System should not give permission to use same username and same email Id to the different users as well as check password confirmation while registration

Case 3	User can view main page with some contents without login and register
Case 4	User should login to view more information
Case 5	Admin should add user
Case 6	Admin should remove user
Case 7	System must store contact information given by user
Case 8	Registered user should able to predict diabetes
Case 9	User must able to view contact us data
Case 10	User without login must not able to use API
Case11	User able to predict diabetes in System without registering data to a database system

### Test 1:

This testing is done with purpose to identify whether the user able to register his details in a system.

Case 1: User must register to the system	
Steps required	<ul style="list-style-type: none"> <li>• User must click on register button</li> <li>• Fill all required columns</li> <li>• Click on register button</li> </ul>
Expected result	Should open login page with successful registration



	<div> <div>animesh</div> <div>.....</div> <div>login</div> </div>
Actual test result	<div> <div>Animesh</div> <div>Timilsina</div> <div>animesh</div> <div>animesh111@gmail.com</div> <div>.....</div> <div>..... </div> <div>Register</div> </div> <div> <div>animesh</div> <div>.....</div> <div>login</div> </div>
Conclusion	Successful

Table 1 Test case 1

**Test 2:**

Different registry blockages were taken while registering the user details such as denying use of same username, email id, matching of two different users, etc. these properties were tested.

Case 2: System should not give permission to use same username and same email Id to the different users	
Step required	<ul style="list-style-type: none"><li>• User must click on register button</li><li>• Fill all required columns with same username or email ID registered before</li><li>• Click on register button</li></ul>
Expected Test Result	<p>Different messages should be displayed on screen i.e.</p> <p>username: "Username taken. Please use next username"</p> <p>Email: "Email taken already.....Please use another email ID"</p> <p>Password: "Passwords not matching. Please confirm your password again"</p>

<b>Actual Result</b>	<div data-bbox="873 210 1123 247"><input type="text"/></div> <div data-bbox="873 289 1123 327"><input type="text"/></div> <div data-bbox="873 369 1123 407"><input type="text"/></div> <div data-bbox="873 449 1123 487"><input type="text"/></div> <div data-bbox="873 529 1123 567"><input type="text"/></div> <div data-bbox="873 609 1123 646"><input type="text"/></div> <div data-bbox="873 726 976 768"><input type="button" value="Register"/></div> <div data-bbox="873 848 1331 882"><p><b>Username taken. Please use next username</b></p></div> <div data-bbox="831 1058 1052 1096"><input type="text"/></div> <div data-bbox="831 1138 1052 1176"><input type="text"/></div> <div data-bbox="831 1218 1052 1255"><input type="text"/></div> <div data-bbox="831 1297 1052 1335"><input type="text"/></div> <div data-bbox="831 1377 1052 1415"><input type="text"/></div> <div data-bbox="831 1457 1052 1495"><input type="text"/></div> <div data-bbox="831 1575 922 1596"><input type="button" value="Register"/></div> <div data-bbox="831 1675 1331 1709"><p><b>Email taken already.....Please use another email ID</b></p></div>
----------------------	--

	<div> <input type="text"/>First Name         </div> <div> <input type="text"/>Last Name         </div> <div> <input type="text"/>Username         </div> <div> <input type="text"/>Email ID         </div> <div> <input type="password"/>Password         </div> <div> <input type="password"/>Confirm Password         </div> <div> <input type="button" value="Register"/> </div> <p><b>Passwords not matching. Please confirm your password again</b></p>
Conclusion	Successful

Table 2 Test Case 2

### Test 3:

Some contents were displayed without login done by the user. It tests such types of performances.

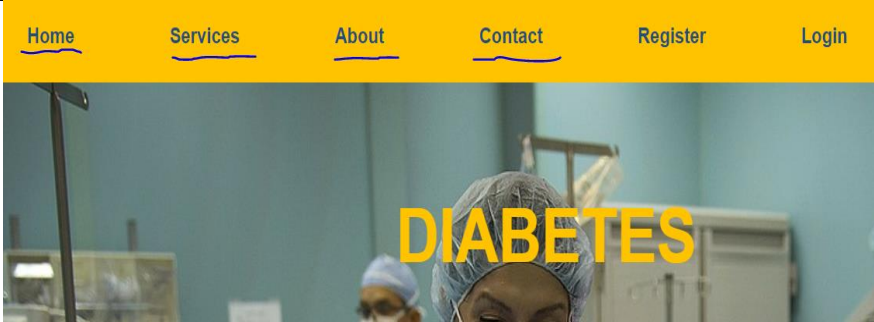
Case 3: User can view main page with some contents without login and register	
Step required	Open home page of the system
Expected result	System must display some content in home page without register and login
Actual result	
conclusion	Result successful

Table 3 Test case 3

**Test 4:**


Case 4: User should login to view more information	
Step required	Do register or login and go to main page
Expected result	Should display some more contents
Actual result	
Conclusion	Result successful

Table 4 Test Case 4

**Test 5:**

Case 5: Admin should add user	
Expected result	Admin should register user

## Actual result

### Add user

First, enter a username and password. Then, you'll be able to edit more user options.

Username:

kaku

Required. 150 characters or fewer. Letters, digits and @/./+/-/\_ only.

Password:

.....

Your password can't be too similar to your other personal information.  
Your password must contain at least 8 characters.  
Your password can't be a commonly used password.  
Your password can't be entirely numeric.

Password confirmation:

.....

Enter the same password as before, for verification.

Save and add another

Save and continue editing

SAVE

### Change user

Username:

kaku

Required. 150 characters or fewer. Letters, digits and @/./+/-/\_ only.

Password:

algorithm: pbkdf2\_sha256 iterations: 180000 salt: btg487\*\*\*\*\* hash: 3DMPqR\*\*\*\*\*

Raw passwords are not stored, so there is no way to see this user's password, but you can change the password using [this form](#).

#### Personal info

First name:

Kamal

Last name:

Kamala

Email address:

kamal11@gmail.com

#### Permissions

☒ Active

Designates whether this user should be treated as active. Unselect this instead of deleting accounts.

☐ Staff status

Designates whether the user can log into this admin site.

☐ Superuser status

Designates that this user has all permissions without explicitly assigning them.

✔ The user "kaku" was changed successfully.

Select user to change

Q  Search

Action:  Go 0 of 7 selected

<input type="checkbox"/>	USERNAME	EMAIL ADDRESS	FIRST NAME	LAST NAME	STAFF STATUS
<input type="checkbox"/>	BulletFlo	np03a17@heraldcollege.edu.np			●
<input type="checkbox"/>	animesh	animesh111@gmail.com	Animesh	Timilsina	○
<input type="checkbox"/>	bhakti	tbhakti@gmail.com	Bhakti	Timilsina	○
<input type="checkbox"/>	bullet1111	bullet@gmail.com	Bullet	Flo	○

## Conclusion

## Successful

Table 5 Test Case 5

## Test 6:

Case 6: Admin should remove user																																	
Steps required	<ul style="list-style-type: none"><li>• Go to admin page</li><li>• Click on change</li><li>• Choose to removing one</li><li>• Select on action to delete and press Go button</li><li>• Confirm to delete</li></ul>																																
Expected result	User kaku should be deleted																																
	<div>Action: <div>Delete selected users ▼</div> <div>Go</div> 1 of 7 selected</div> <table><tr><th><input type="checkbox"/></th><th>USERNAME</th><th>EMAIL ADDRESS</th><th>FIRST NAME</th></tr><tr><td><input type="checkbox"/></td><td>BulletFlo</td><td>np03a17@heraldcollege.edu.np</td><td></td></tr><tr><td><input type="checkbox"/></td><td>animesh</td><td>animesh111@gmail.com</td><td>Animesh</td></tr><tr><td><input type="checkbox"/></td><td>bhakti</td><td>tbhakti@gmail.com</td><td>Bhakti</td></tr><tr><td><input type="checkbox"/></td><td>bullet1111</td><td>bullet@gmail.com</td><td>Bullet</td></tr><tr><td><input checked="" type="checkbox"/></td><td>kaku</td><td>kamal11@gmail.com</td><td>Kamal</td></tr><tr><td><input type="checkbox"/></td><td>kitty</td><td>sandeshlamsal1111111@gmail.com</td><td>Sandesh</td></tr><tr><td><input type="checkbox"/></td><td>pujadear</td><td>puja@gmail.com</td><td>Puja</td></tr></table> <p>Are you sure?</p> <p>Are you sure you want to delete the selected user? All of the following objects and their related items will be deleted:</p> <p>Summary</p> <ul style="list-style-type: none"><li>▪ Users: 1</li></ul> <p>Objects</p> <ul style="list-style-type: none"><li>▪ User: kaku</li></ul> <div><div>Yes, I'm sure</div><div>No, take me back</div></div>	<input type="checkbox"/>	USERNAME	EMAIL ADDRESS	FIRST NAME	<input type="checkbox"/>	BulletFlo	np03a17@heraldcollege.edu.np		<input type="checkbox"/>	animesh	animesh111@gmail.com	Animesh	<input type="checkbox"/>	bhakti	tbhakti@gmail.com	Bhakti	<input type="checkbox"/>	bullet1111	bullet@gmail.com	Bullet	<input checked="" type="checkbox"/>	kaku	kamal11@gmail.com	Kamal	<input type="checkbox"/>	kitty	sandeshlamsal1111111@gmail.com	Sandesh	<input type="checkbox"/>	pujadear	puja@gmail.com	Puja
<input type="checkbox"/>	USERNAME	EMAIL ADDRESS	FIRST NAME																														
<input type="checkbox"/>	BulletFlo	np03a17@heraldcollege.edu.np																															
<input type="checkbox"/>	animesh	animesh111@gmail.com	Animesh																														
<input type="checkbox"/>	bhakti	tbhakti@gmail.com	Bhakti																														
<input type="checkbox"/>	bullet1111	bullet@gmail.com	Bullet																														
<input checked="" type="checkbox"/>	kaku	kamal11@gmail.com	Kamal																														
<input type="checkbox"/>	kitty	sandeshlamsal1111111@gmail.com	Sandesh																														
<input type="checkbox"/>	pujadear	puja@gmail.com	Puja																														

Action:

Go

0 of 6 selected

<input type="checkbox"/>	USERNAME	EMAIL ADDRESS	FIRST NAME	LA
<input type="checkbox"/>	BulletFlo	np03a17@heraldcollege.edu.np		
<input type="checkbox"/>	animesh	animesh111@gmail.com	Animesh	Ti
<input type="checkbox"/>	bhakti	tbhakti@gmail.com	Bhakti	Ti
<input type="checkbox"/>	bullet1111	bullet@gmail.com	Bullet	Fl
<input type="checkbox"/>	kitty	sandeshlamsal1111111@gmail.com	Sandesh	L
<input type="checkbox"/>	pujadear	puja@gmail.com	Puja	U

6 users

Absence of username kaku in new username list

Conclusion

Successful

Table 6 Test case 6

(note: These two above features are requirement of a system. This feature is found default on Django framework)

## Test 7:

A contact page is used in a system to collaborate with the clients and address their problems. They can post their queries regards various problems faced by them.

Case 7: System must store contact information given by user	
Steps required	Give some information on contact form
Expected result	Returns Message: "contact form submitted successfully"



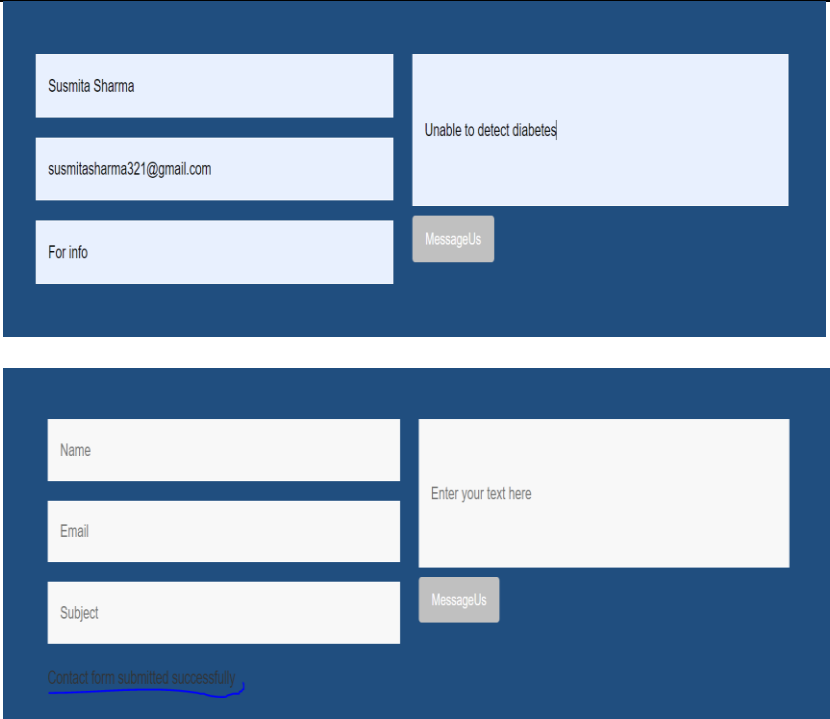
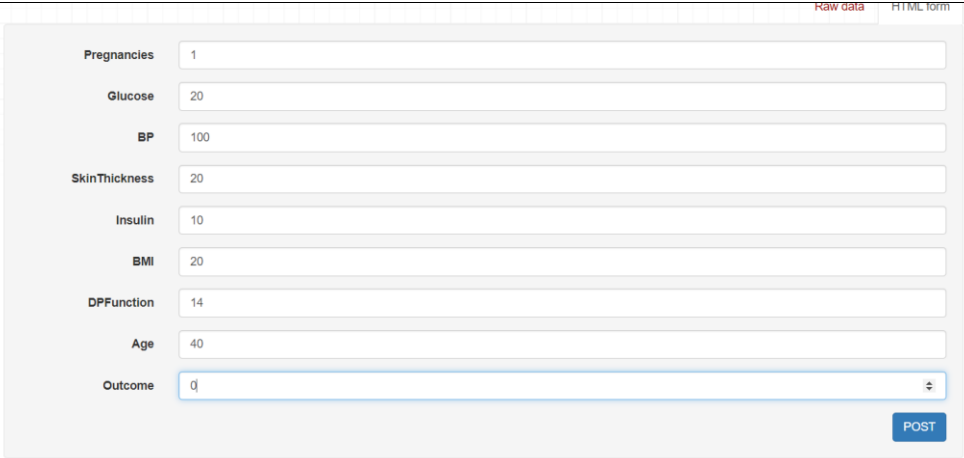
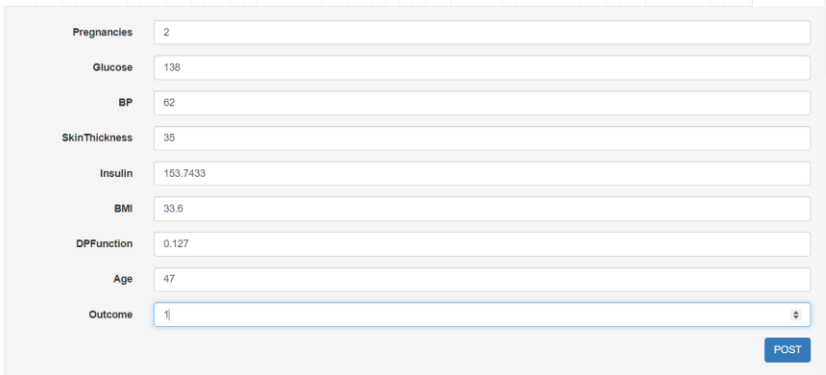
Actual result	
Conclusion	Successful

Table 7 Test Case 7

### Test 8:

The user only will be able to detect diabetes in API of diabetes detection system. User must register to use this API.


Case 8: Registered user should able to predict diabetes	
Steps required	Click check on navbar which drags you to check diabetes Onclick: detect your diabetes Fill up: Row data OR Fill up HTML form

	<p>Input data</p> <p>Click on: post</p> <p>View result</p>
Expected result	Possible output 0 and 1
Actual Result	 <pre> HTTP 200 OK Allow: GET, POST, HEAD, OPTIONS Content-Type: application/json Vary: Accept  {   "status": "Success",   "Diabetic status": [     0   ],   "tmp": [] } </pre> 

	<pre> POST /detector/api/detector/  HTTP 200 OK Allow: GET, POST, HEAD, OPTIONS Content-Type: application/json Vary: Accept  {   "status": "Success",   "Diabetic status": [     1   ],   "tmp": [] } </pre>
Conclusion	Successful

Table 8 Test Case 8

## Test 9:

Case 9: Admin should able to view contact Us data	
Steps required	<ul style="list-style-type: none"> <li>Go to admin page</li> <li>View contact Us details</li> </ul>
Expected result	Adding Contact feature to admin panel and giving all permission to admin
	

	<div>Select contact us to change</div> <div>Action: <div></div> Go 0 of 1 selected</div> <table><thead><tr><th><div></div></th><th>NAME</th><th>EMAIL</th><th>SUBJECT</th><th>MESSAGE</th><th>DATE</th></tr></thead><tbody><tr><td><div></div></td><td><a href="#">kabir</a></td><td>kabir@gmail.com</td><td>information</td><td>stay safe</td><td>May 21, 2020, 9:10 a.m.</td></tr></tbody></table> <div>1 contact us</div>	<div></div>	NAME	EMAIL	SUBJECT	MESSAGE	DATE	<div></div>	<a href="#">kabir</a>	kabir@gmail.com	information	stay safe	May 21, 2020, 9:10 a.m.
<div></div>	NAME	EMAIL	SUBJECT	MESSAGE	DATE								
<div></div>	<a href="#">kabir</a>	kabir@gmail.com	information	stay safe	May 21, 2020, 9:10 a.m.								
Conclusion	Successful												

Table 9 Test Case 9

## Test 10

This test is carried out to test security label of API router. Business logic is to built an API with such security so that it should be only possible to use by logged in user. However, logic is been carried out in HTML page i.e. only able to view by authenticated user, users may try to access an API through urls which is being protected.

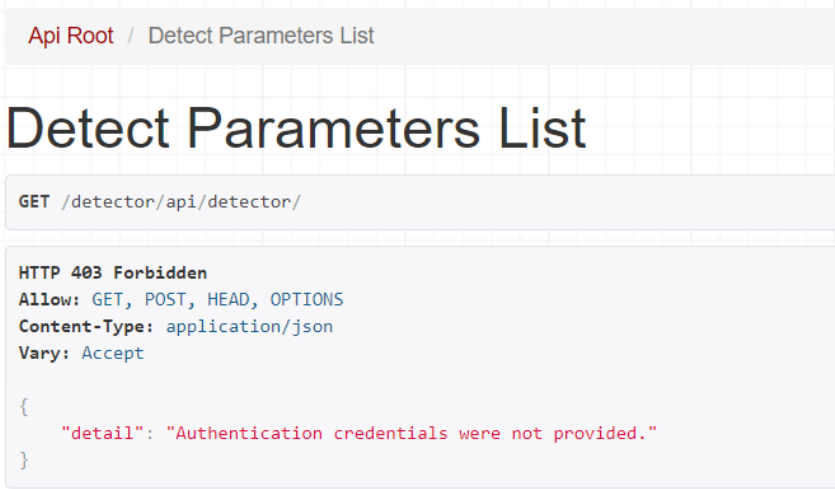
Case 10	User without login must not able to use API
Step required	Try to access the API router through url
Expected Result	Should not get permission to use API
Actual Result	
Conclusion	Successful

Table 10 Test case 10

## Test 11

One of the features is made to detect diabetes i.e. user is able to detect possibility of diabetes on him/her without giving their health data in a system.

Case 11	User able to predict diabetes in System without registering data to a database system
Step required	Go to check page and click on detect diabetes where it defines your data is not used for further analysis in a system
Expected Result	Should predict diabetes successfully
Actual Result	<div><b>Diabetes Prediction</b></div> <div><div>2</div><div>120</div><div>115</div><div>43</div><div>800</div><div>40</div><div>0.12</div></div>

	<p><b>Diabetes Prediction</b></p> <hr/> <p><b>You don't have diabetes disease.</b></p> <p>pregnancies</p> <p>glucose</p> <p>BP</p> <p>SkinThickness</p> <p>insulin</p> <p>BMI</p>
Conclusion	Successful

Table 11 Test Case 11

#### 4. Answering academic question:

*How this application detects diabetes in a Patient and what are its features?*

Any academic questions in a report is responsible to answer the major question about the project that is going to be build. This is a web application which is mainly designed with an objective to detect the presence of diabetes in a person by using some general factors on diabetes such as Insulin, glucose, BMI, Pregnancies, age, etc. To make system sensitive and detect the diabetes with correct data analysis and read the data pattern, it is equipped with some older dataset tests to either diabetic or non-diabetic person and a KNN algorithm which have its own approach to analyze the data and filter some useful information through it. KNN algorithm will classify the patient whether be a diabetic or non-diabetic analyzing the data points in a dataset. This application will

provide a UI to fill details of factors of any patient and submission of result will provide its outcome. The model is used from one of the famous ML library i.e. scikit-learn and have used Euclidian distance to represent value of K. This application will be web based and will provides way to some important information about diabetes such as signs, symptoms, preventive measures, food habits, etc.

## **5. Conclusion:**

Finally, I want to conclude that this project was built to predict the presence of diabetes Meletus to any patient. This project was built under many circumstances. First of all, some factors responsible to detect the presence of diabetes are classified from raw data. After this process, different ML and ANN models are feed with training and testing datasets and accuracy between them is compared. All compared models are Logistic regression, KNN, Decision tree, Random Forest Classifier and ANN where decision tree, KNN and Random Forest classifiers have given better accuracy, ANN with remarkable values of hyperparameter is good while logistic regression has given poor result. All these researches were done for model analysis and proposed model i.e. KNN was finally used to detect diabetes in a patient.

Apart from diabetes prediction system, the system is also featured with a web application which provides User interface for prediction and as well consists of authentication system, user contacts and some facility of dynamically adding of some images and important information dynamically from the database. It also consists of some static information that will provide important information about diabetes to the users and non-members.

The project was completed doing project plans, following SDLC methodologies and using project management tool to monitor and control the project. Detailed explanations and evaluation of project are done explaining the use of tools and technologies. It also includes some information of architectural and procedural designs made to complete the project. Evaluating the project clearly explaining all fact-finding techniques, functional and non-functional requirements and testing of the project is fulfilled in this report.

### **5.1. Future Escalation:**

According to WHO there are almost 436000 diabetes patients in Nepal in 2009 and predicted to rise up to 1328000 by year 2030. Hence, this field is one of the most important field which should needed to solved by using different types of technologies as possible. Artificial

intelligence is an emerging field which can be one of the efficient solutions to detect diabetic and non-diabetic patients.

This system will help to identify presence of diabetes in a patient. People from rural and remote areas where seems to be shortage of health services and hospitals will take better advantage of it. This will save time of users and as well as money which they used to spend for diabetes check-ups. If any user is unaware about his health (diabetic mellitus) status than he can view his result by using diabetes prediction system of this application. He will do further work as per the result given by the prediction system.

This application will be developed with other several ML and Neural Net models in a future. It is an earlier stage of diabetes detection and further it will be featured with prediction using other medical factors so that it can be used by health institutes. It also will develop function to detect diabetes using term diabetic retinopathy where image processing of human eye will be done and diabetes will be classified.

Thus, I can claim that such kinds of application will be in a high demand because people will be more curious about their health status and will always try to get their health information if the service is available at free of cost.

## **6. Critical Evaluation and Findings:**

During the time of project development, I have given proposal of two kinds of applications that will be integrated together to perform the common tasks related to diabetes. First one is a static and dynamic combination of web application which integrates some functional and non-functional requirements which will provide efficiency to the project. And another part is system that uses machine learning technique which feeds some factors and gives an outcome as per dependency on several factors that indicate the possibility of diabetes. I have tried different types of algorithms and gained varied value of accuracy and sensitivity.

Some of the algorithms I have tried on my project and compared their accuracy which has given different knowledges and uses about an algorithm.

Some algorithms I tried and compared in my project are:

- Decision tree



- Random Forest
- K Nearest Neighbors
- Logistic regression

As I have chosen to use K Neighbors classifier as an AI model in my project, hence I have used this model which provided me an accuracy of about 93% and sensitivity of 91% and f1-score of 0.89% to give result “Yes” when the value of K is 1. Value of an accuracy is gained while feeding the purposed dataset in a purposed model.

```
In [212]: # k-Nearest neighbour classifier
from sklearn.metrics import classification_report
print(classification_report(y_test, predictions))
```

	precision	recall	f1-score	support
0	0.96	0.94	0.95	547
1	0.87	0.91	0.89	253
accuracy			0.93	800
macro avg	0.91	0.92	0.92	800
weighted avg	0.93	0.93	0.93	800

Figure 54 KNN model

Similarly, other algorithms i.e. decision tree has given an accuracy of 89% where recall value is 0.83 and as well f1-score of 83% to predict “Yes” when random state is 2.

```
In [219]: #Decision tree classifier
print(classification_report(y_test, predictDic))
```

	precision	recall	f1-score	support
0	0.92	0.92	0.92	547
1	0.83	0.83	0.83	253
accuracy			0.89	800
macro avg	0.87	0.87	0.87	800
weighted avg	0.89	0.89	0.89	800

Figure 55 Decision tree model

Logistic regression had given an accuracy of 78% with 58% of recall value and 63% of f1-score which is not actually a good classification.

```
In [227]: #logistic regression
print(classification_report(y_test, LogModel))
```

	precision	recall	f1-score	support
0	0.82	0.87	0.84	547
1	0.68	0.58	0.63	253
accuracy			0.78	800
macro avg	0.75	0.73	0.74	800
weighted avg	0.78	0.78	0.78	800

Figure 56 Logistic regression model

Random Tree Classifier has given an accuracy of 93% with value of recall to 85% and f1-score of 88% which is also very good accuracy.

```
In [233]: #Random forest Classifier
print(classification_report(y_test, RandModel))
```

	precision	recall	f1-score	support
0	0.93	0.96	0.95	547
1	0.91	0.85	0.88	253
accuracy			0.93	800
macro avg	0.92	0.90	0.91	800
weighted avg	0.93	0.93	0.93	800

Figure 57 Random Forest Classifier Model

Finally, Artificial Neural Network is also used and has given an accuracy of 86% with 79% of sensitivity and 78% of f1-score to predict “Yes”. This is also a good accuracy.

```
In [240]: #Neural network MLP classifier result
print(classification_report(y_test, MLPdiabetes))
```

	precision	recall	f1-score	support
0	0.90	0.90	0.90	547
1	0.78	0.79	0.78	253
accuracy			0.86	800
macro avg	0.84	0.84	0.84	800
weighted avg	0.86	0.86	0.86	800

Figure 58 MLP model

Since, this project is related to critical field hence it should have high possibility to say “Yes” while detecting diabetes Meletus. In comparison between all these algorithms, I found K-NN, random forest and decision tree, etc. were the better approaches to use in such types of dataset.

As ANN is an advanced model which works on base of brain analogies, but have given lower accuracy comparing to some above common ML algorithms. This may because of low amount of data present in a dataset. Neural Networks are mainly efficient to use on those where dimension and amount of data are high e.g. Images processing, business analytics, government data records, etc. where data dimension and sources are high.

KNN seems to be suitable in this dataset according to the number of records, dimensions and distribution of data in a dataset. Hence, I have used this model in my project.

KNN is not suitable to those where number of records are very high, highly deviated data and with very high dimensions.

We can even use decision tree and random forest classifier which seems having similar types of accuracy. They can give best predicted value for similar kind of data sets.

All of these tasks were performed as per the Gantt chart which was given above in a report. Project Management tools had given me guidelines to give continuity on works.

To come over with all these, lots of challenges i.e. ideas of using different tools for dataset analysis, working mechanisms of algorithms, etc. been faced as a challenge which I solved with time to time research and comparison. Similarly, Final report build according to the development of the system is trying to achieve each and every effort made during the development phase. It almost covering all the contents along with SRS documents, its architecture designs, system designs and as well as design of database in a schema level. This report reflects almost all the features of this system in mirror view.

## 7. References:

- A., R., 2014. Know the sign and symptoms of diabetes. *The Indian journal of medical research*, 140(5), pp. 579-581.
- Alaliyat, S., 2020. *Video -based Fall Detection in Elderly's Houses*. [Online] Available at: [https://www.researchgate.net/publication/267953942\\_Video -based Fall Detection in Elderly's Houses](https://www.researchgate.net/publication/267953942_Video_-based_Fall_Detection_in_Elderly's_Houses) [Accessed 10 5 2020].
- Alghamdi, D. A. A., Imran, M. & Ahmad, B., 2016. SOFTWARE ENGINEERING:. *International Journal of Computer Science and Mobile Computing*, 5(3), pp. 801-815.
- altexsoft, 2019. *Web Application Architecture: How the Web Works*. [Online] Available at: <https://www.altexsoft.com/blog/engineering/web-application-architecture-how-the-web-works/> [Accessed 21 5 2020].
- Azam & F., 2000. *Biologically Inspired Modular Neural Networks*, s.l.: Virginia Tech.
- Brownlee, J., 2014. *A Gentle Introduction to Scikit-Learn: A Python Machine Learning Library*. [Online] Available at: <https://machinelearningmastery.com/a-gentle-introduction-to-scikit-learn-a-python-machine-learning-library/> [Accessed 22 5 2020].
- Cheng, D. et al., 2014. kNN Algorithm with Data-Driven k Value. *Springer International Publishing Switzerland*, pp. 499-512.
- EL\_Jerjawi, N. S. & Abu-Naser, S. S., 2018. Diabetes Prediction Using Artificial Neural Network. *International Journal of Advanced Science and Technology*, Volume 121, pp. 55-64.
- Guru99, 2020. *What is a Functional Requirement? Specification, Types, EXAMPLES*. [Online] Available at: <https://www.guru99.com/functional-requirement-specification-example.html> [Accessed 19 5 2020].
- Ivivity, 2019. *Iterative Model in Software Development: Pros and Cons*. [Online] Available at: <https://lvivity.com/iterative-model> [Accessed 15 5 2020].
- Javatpoint, 2018. *Coding*. [Online] Available at: <https://www.javatpoint.com/software-engineering-coding> [Accessed 20 5 2020].
- Javatpoint, 2018. *Iterative Model*. [Online] Available at: <https://www.javatpoint.com/software-engineering-iterative-model> [Accessed 17 5 2020].
- Johnson, J., n.d. *Python Numpy Tutorial (with Jupyter and Colab)*. [Online] Available at: <https://cs231n.github.io/python-numpy-tutorial/#numpy> [Accessed 21 5 2020].

Joshi, T. N. & Chawan, P. P. M., 2018. Logistic Regression and SVM based Diabetes Prediction System. *International Journal for Technological Research In Engineering*, 5(11), pp. 4347-4351.

K., K. et al., 2013. *Introduction to Diabetes Mellitus*. New York: Springer.

Kaur, G. & chhabra, A., 2014. Improved J48 Classification Algorithm for the. *International Journal of Computer Applications*, Volume 98, pp. 13-17.

Kenton, W., 2019. *Feasibility Study*. [Online] Available at: <https://www.investopedia.com/terms/f/feasibility-study.asp> [Accessed 18 5 2020].

Lakhtakia, R., 2013. The History of Diabetes Mellitus. *Sultan Qaboos Univ Med J.*, 13(doi: 10.12816/0003257), pp. 368-370.

Marr, B., 2016. *A Short History of Machine Learning -- Every Manager Should Read*. [Online] Available at: <https://www.forbes.com/sites/quora/2020/05/27/what-should-businesses-look-for-in-a-cloud-service/#43f3ba9b385f> [Accessed 11 5 2020].

McCarthy, J., 1955. *Dartmouth Artificial Intelligence*. New Hampshire, Dartmouth Artificial Intelligence (AI) Conference.

Nagyfi, R., 2018. *The differences between Artificial and Biological Neural Networks*. [Online] Available at: <https://towardsdatascience.com/the-differences-between-artificial-and-biological-neural-networks-a8b46db828b7> [Accessed 29 5 2020].

Paradism, V., 2020. *What is Class diagram?*. [Online] Available at: <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-class-diagram/> [Accessed 20 5 2020].

Peterson, L. E., 2009. *K-nearest neighbor*, s.l.: Scholarpedia.

Prasath, V. S. et al., 2019. *Effects of Distance Measure Choice on KNN Classifier Performance - A Review*, s.l.: s.n.

Saxena, K., Khan, D. & Singh, S., 2014. Diagnosis of Diabetes Mellitus using K Nearest. *International Journal of Computer Science Trends and Technology (IJCST)*, 2(4), p. 37.

sebastianraschka, 2020. *Why is Nearest Neighbor a Lazy Algorithm?*. [Online] Available at: <https://sebastianraschka.com/faq/docs/lazy-knn.html> [Accessed 11 5 2020].

Tutorialspoint, 2020. *Python Pandas - Introduction*. [Online] Available at: [https://www.tutorialspoint.com/python\\_pandas/python\\_pandas\\_introduction.htm](https://www.tutorialspoint.com/python_pandas/python_pandas_introduction.htm) [Accessed 22 5 2020].

Weinberger, K. Q., Blitzer, J. & Saul, L. K., n.d. *Distance Metric Learning for Large Margin*, Philadelphia: Department of Computer and Information Science, University of Pennsylvania.

World Helath Organization, 2016. *Global Report On Diabetes*, France: MEO Design & Communication, meomeo.ch.

## **8. Abbreviations:**

WHO: World Health Organizations

AI: Artificial Intelligence

ML: Machine Learning

MLP: Multi-Layer Perceptron

ANN: Artificial Neural Network

KNN: K-Nearest Neighbors

SDLC: Software Development Lifecycle

BMI: Body Mass Index

## **9. Appendix:**

### **9.1. Different analysis and visualizations done to develop models:**

```
In [31]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sn
```

```
In [32]: data = pd.read_csv(r"diabetes.csv")
```

```
In [33]: data.head(10)
```

```
Out[33]:
```

	pregnancies	glucose	BP	SkinThickness	insulin	BMI	DPFunction	age	outcome
0	2	138	62	35	0	33.6	0.127	47	1
1	0	84	82	31	125	38.2	0.233	23	0
2	0	145	0	0	0	44.2	0.630	31	1
3	0	135	68	42	250	42.3	0.365	24	1
4	1	139	62	41	480	40.7	0.536	21	0
5	0	173	78	32	265	46.5	1.159	58	0
6	4	99	72	17	0	25.6	0.294	28	0
7	8	194	80	0	0	26.1	0.551	67	0
8	2	83	65	28	66	36.8	0.629	24	0
9	2	89	90	30	0	33.5	0.292	42	0

```
In [34]: data.tail(10)
```

```
Out[34]:
```

	pregnancies	glucose	BP	SkinThickness	insulin	BMI	DPFunction	age	outcome
1990	3	111	90	12	78	28.4	0.495	29	0
1991	6	102	82	0	0	30.8	0.180	36	1
1992	6	134	70	23	130	35.4	0.542	29	1
1993	2	87	0	23	0	28.9	0.773	25	0
1994	1	79	60	42	48	43.5	0.678	23	0
1995	2	75	64	24	55	29.7	0.370	33	0
1996	8	179	72	42	130	32.7	0.719	36	1
1997	6	85	78	0	0	31.2	0.382	42	0
1998	0	129	110	46	130	67.1	0.319	26	1
1999	2	81	72	15	76	30.1	0.547	25	0



```
In [35]: data.shape
```

```
Out[35]: (2000, 9)
```

```
In [36]: data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2000 entries, 0 to 1999
Data columns (total 9 columns):
#   Column          Non-Null Count  Dtype
---  -
0   pregnancies      2000 non-null   int64
1   glucose          2000 non-null   int64
2   BP               2000 non-null   int64
3   SkinThickness    2000 non-null   int64
4   insulin          2000 non-null   int64
5   BMI              2000 non-null   float64
6   DPFunction       2000 non-null   float64
7   age              2000 non-null   int64
8   outcome          2000 non-null   int64
dtypes: float64(2), int64(7)
memory usage: 140.8 KB
```

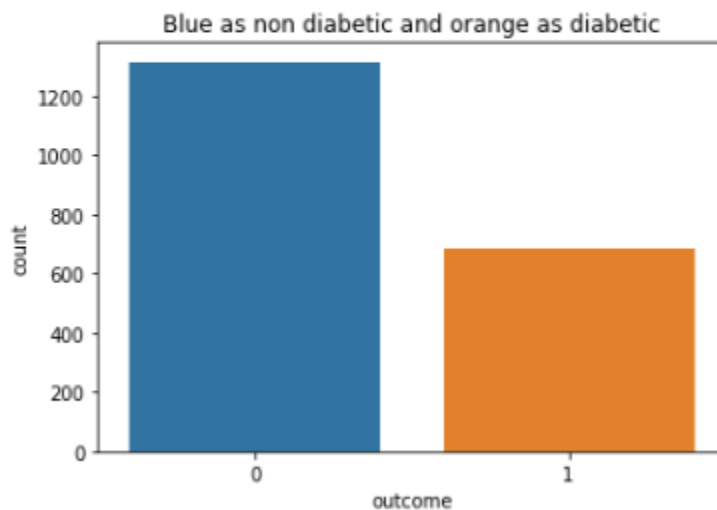
```
In [37]: data.describe()
```

```
Out[37]:
```

	pregnancies	glucose	BP	SkinThickness	insulin	BMI	DPFunction	age	outcome
count	2000.000000	2000.000000	2000.000000	2000.000000	2000.000000	2000.000000	2000.000000	2000.000000	2000.000000
mean	3.703500	121.182500	69.145500	20.935000	80.254000	32.193000	0.470930	33.090500	0.342000
std	3.306063	32.068636	19.188315	16.103243	111.180534	8.149901	0.323553	11.786423	0.474498
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.078000	21.000000	0.000000
25%	1.000000	99.000000	63.500000	0.000000	0.000000	27.375000	0.244000	24.000000	0.000000
50%	3.000000	117.000000	72.000000	23.000000	40.000000	32.300000	0.376000	29.000000	0.000000
75%	6.000000	141.000000	80.000000	32.000000	130.000000	36.800000	0.624000	40.000000	1.000000
max	17.000000	199.000000	122.000000	110.000000	744.000000	80.600000	2.420000	81.000000	1.000000

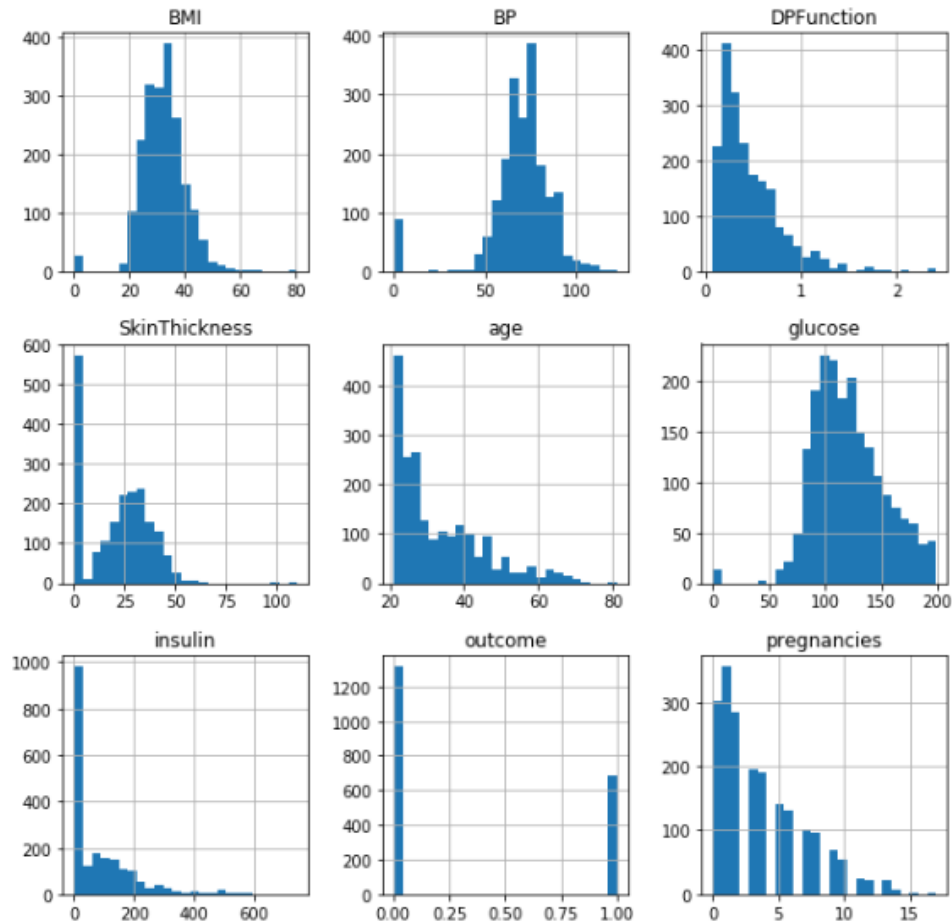
```
In [38]: sn.countplot(data['outcome'])
plt.title('Blue as non diabetic and orange as diabetic')
```

```
Out[38]: Text(0.5, 1.0, 'Blue as non diabetic and orange as diabetic')
```



```
In [39]: data.hist(figsize = (10,10), bins=25)
```

```
Out[39]: array([[<matplotlib.axes._subplots.AxesSubplot object at 0x000002281A946208>,  
<matplotlib.axes._subplots.AxesSubplot object at 0x000002281C4604C8>,  
<matplotlib.axes._subplots.AxesSubplot object at 0x000002281C47D708>],  
[<matplotlib.axes._subplots.AxesSubplot object at 0x000002281C484C08>,  
<matplotlib.axes._subplots.AxesSubplot object at 0x000002281C4F0148>,  
<matplotlib.axes._subplots.AxesSubplot object at 0x000002281C525608>],  
[<matplotlib.axes._subplots.AxesSubplot object at 0x000002281C55CD08>,  
<matplotlib.axes._subplots.AxesSubplot object at 0x000002281C596948>,  
<matplotlib.axes._subplots.AxesSubplot object at 0x000002281C5A1448>]],  
dtype=object)
```



```
In [40]: dataset= data.copy(deep = True)
```

```
In [41]: #to choose certain numbers of columns in a data  
# dataset[dataset.columns[0:7]]
```

```
In [42]: # To check the missing (0) values which are in this dataset which needed to be  
#cleaned to get better result  
  
print("number of missing pregnancies i.e. having (0) are : {}".format(len(dataset.loc[dataset['pregnancies'] == 0])))  
print("number of missing glucose i.e. having (0) are : {}".format(len(dataset.loc[dataset['glucose'] == 0])))  
print("number of missing BP i.e. having (0) are : {}".format(len(dataset.loc[dataset['BP'] == 0])))  
print("number of missing SkinThickness i.e. having (0) are : {}".format(len(dataset.loc[dataset['SkinThickness'] == 0])))  
print("number of missing insulin i.e. having (0) are : {}".format(len(dataset.loc[dataset['insulin'] == 0])))  
print("number of missing BMI i.e. having (0) are : {}".format(len(dataset.loc[dataset['BMI'] == 0])))  
print("number of missing DiabetesPedigreeFunction i.e. having (0) are : {}".format(len(dataset.loc[dataset['DPFunction'] == 0])))  
print("number of missing age i.e. having (0) are : {}".format(len(dataset.loc[dataset['age'] == 0])))
```

```
number of missing pregnancies i.e. having (0) are : 301  
number of missing glucose i.e. having (0) are : 13  
number of missing BP i.e. having (0) are : 90  
number of missing SkinThickness i.e. having (0) are : 573  
number of missing insulin i.e. having (0) are : 956  
number of missing BMI i.e. having (0) are : 28  
number of missing DiabetesPedigreeFunction i.e. having (0) are : 0  
number of missing age i.e. having (0) are : 0
```

```
In [43]: dataset[['pregnancies', 'glucose', 'BP', 'SkinThickness', 'insulin']] = dataset[['pregnancies', 'glucose', 'BP', 'SkinThickness',
```

```
In [44]: dataset['glucose'].fillna(dataset['glucose'].mean(), inplace = True)  
dataset['SkinThickness'].fillna(dataset['SkinThickness'].mean(), inplace = True)  
dataset['BP'].fillna(dataset['BP'].median(), inplace = True)  
dataset['insulin'].fillna(dataset['insulin'].mean(), inplace = True)  
dataset['pregnancies'].fillna(dataset['pregnancies'].mean(), inplace = True)
```

```
In [45]: data.head(20)
```

```
Out[45]:
```

	pregnancies	glucose	BP	SkinThickness	insulin	BMI	DPFunction	age	outcome
0	2	138	62	35	0	33.6	0.127	47	1
1	0	84	82	31	125	38.2	0.233	23	0
2	0	145	0	0	0	44.2	0.630	31	1
3	0	135	68	42	250	42.3	0.365	24	1
4	1	139	62	41	480	40.7	0.536	21	0
5	0	173	78	32	265	46.5	1.159	58	0
6	4	99	72	17	0	25.6	0.294	28	0
7	8	194	80	0	0	26.1	0.551	67	0
8	2	83	65	28	66	36.8	0.629	24	0
9	2	89	90	30	0	33.5	0.292	42	0
10	4	99	68	38	0	32.8	0.145	33	0
11	4	125	70	18	122	28.9	1.144	45	1
12	3	80	0	0	0	0.0	0.174	22	0
13	6	166	74	0	0	26.6	0.304	66	0
14	5	110	68	0	0	26.0	0.292	30	0
15	2	81	72	15	76	30.1	0.547	25	0
16	7	195	70	33	145	25.1	0.163	55	1
17	6	154	74	32	193	29.3	0.839	39	0
18	2	117	90	19	71	25.2	0.313	21	0
19	3	84	72	32	0	37.2	0.267	28	0

In [46]: dataset.head(20)

Out[46]:

	pregnancies	glucose	BP	SkinThickness	insulin	BMI	DPFunction	age	outcome
0	2.000000	138.0	62.0	35.000000	153.743295	33.6	0.127	47	1
1	4.359623	84.0	82.0	31.000000	125.000000	38.2	0.233	23	0
2	4.359623	145.0	72.0	29.341275	153.743295	44.2	0.630	31	1
3	4.359623	135.0	68.0	42.000000	250.000000	42.3	0.365	24	1
4	1.000000	139.0	62.0	41.000000	480.000000	40.7	0.536	21	0
5	4.359623	173.0	78.0	32.000000	265.000000	46.5	1.159	58	0
6	4.000000	99.0	72.0	17.000000	153.743295	25.6	0.294	28	0
7	8.000000	194.0	80.0	29.341275	153.743295	26.1	0.551	67	0
8	2.000000	83.0	65.0	28.000000	66.000000	36.8	0.629	24	0
9	2.000000	89.0	90.0	30.000000	153.743295	33.5	0.292	42	0
10	4.000000	99.0	68.0	38.000000	153.743295	32.8	0.145	33	0
11	4.000000	125.0	70.0	18.000000	122.000000	28.9	1.144	45	1
12	3.000000	80.0	72.0	29.341275	153.743295	0.0	0.174	22	0
13	6.000000	166.0	74.0	29.341275	153.743295	26.6	0.304	66	0
14	5.000000	110.0	68.0	29.341275	153.743295	26.0	0.292	30	0
15	2.000000	81.0	72.0	15.000000	76.000000	30.1	0.547	25	0
16	7.000000	195.0	70.0	33.000000	145.000000	25.1	0.163	55	1
17	6.000000	154.0	74.0	32.000000	193.000000	29.3	0.839	39	0
18	2.000000	117.0	90.0	19.000000	71.000000	25.2	0.313	21	0
19	3.000000	84.0	72.0	32.000000	153.743295	37.2	0.267	28	0

```
In [47]: # To check the missing (0) values which are in this dataset which needed to be
#cleaned to get better result

print("number of missing pregnancies i.e. having (0) are : {}".format(len(dataset.loc[dataset['pregnancies'] == 0])))
print("number of missing glucose i.e. having (0) are : {}".format(len(dataset.loc[dataset['glucose'] == 0])))
print("number of missing BP i.e. having (0) are : {}".format(len(dataset.loc[dataset['BP'] == 0])))
print("number of missing SkinThickness i.e. having (0) are : {}".format(len(dataset.loc[dataset['SkinThickness'] == 0])))
print("number of missing insulin i.e. having (0) are : {}".format(len(dataset.loc[dataset['insulin'] == 0])))
print("number of missing BMI i.e. having (0) are : {}".format(len(dataset.loc[dataset['BMI'] == 0])))
print("number of missing DiabetesPedigreeFunction i.e. having (0) are : {}".format(len(dataset.loc[dataset['DPFunction'] == 0])))
print("number of missing age i.e. having (0) are : {}".format(len(dataset.loc[dataset['age'] == 0])))

number of missing pregnancies i.e. having (0) are : 0
number of missing glucose i.e. having (0) are : 0
number of missing BP i.e. having (0) are : 0
number of missing SkinThickness i.e. having (0) are : 0
number of missing insulin i.e. having (0) are : 0
number of missing BMI i.e. having (0) are : 28
number of missing DiabetesPedigreeFunction i.e. having (0) are : 0
number of missing age i.e. having (0) are : 0
```

```
In [48]: dataset['pregnancies']= dataset['pregnancies'].astype('int64')
```

```
In [49]: dataset.head(20)
```

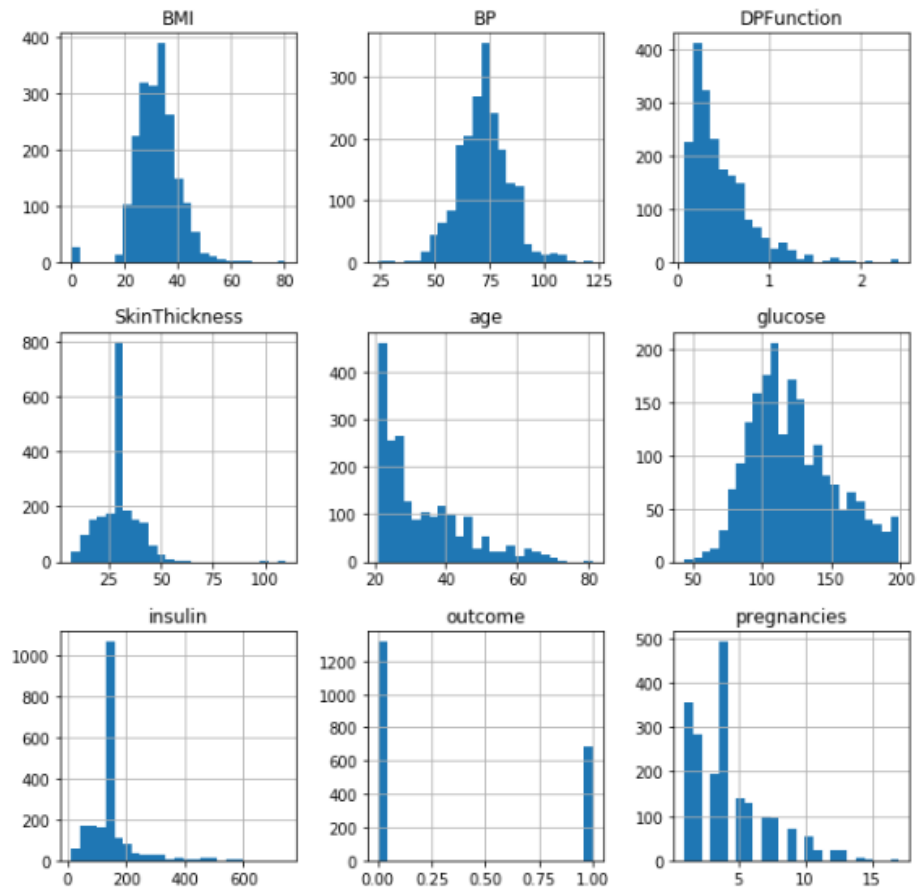
```
Out[49]:
```

	pregnancies	glucose	BP	SkinThickness	insulin	BMI	DPFunction	age	outcome
0	2	138.0	62.0	35.000000	153.743295	33.6	0.127	47	1
1	4	84.0	82.0	31.000000	125.000000	38.2	0.233	23	0
2	4	145.0	72.0	29.341275	153.743295	44.2	0.630	31	1
3	4	135.0	68.0	42.000000	250.000000	42.3	0.365	24	1
4	1	139.0	62.0	41.000000	480.000000	40.7	0.536	21	0
5	4	173.0	78.0	32.000000	265.000000	46.5	1.159	58	0
6	4	99.0	72.0	17.000000	153.743295	25.6	0.294	28	0
7	8	194.0	80.0	29.341275	153.743295	26.1	0.551	67	0
8	2	83.0	65.0	28.000000	66.000000	36.8	0.629	24	0
9	2	89.0	90.0	30.000000	153.743295	33.5	0.292	42	0
10	4	99.0	68.0	38.000000	153.743295	32.8	0.145	33	0
11	4	125.0	70.0	18.000000	122.000000	28.9	1.144	45	1
12	3	80.0	72.0	29.341275	153.743295	0.0	0.174	22	0
13	6	166.0	74.0	29.341275	153.743295	26.6	0.304	66	0
14	5	110.0	68.0	29.341275	153.743295	26.0	0.292	30	0
15	2	81.0	72.0	15.000000	76.000000	30.1	0.547	25	0
16	7	195.0	70.0	33.000000	145.000000	25.1	0.163	55	1
17	6	154.0	74.0	32.000000	193.000000	29.3	0.839	39	0
18	2	117.0	90.0	19.000000	71.000000	25.2	0.313	21	0
19	3	84.0	72.0	32.000000	153.743295	37.2	0.267	28	0

```
In [50]: dataset.to_csv('file1.csv')
```

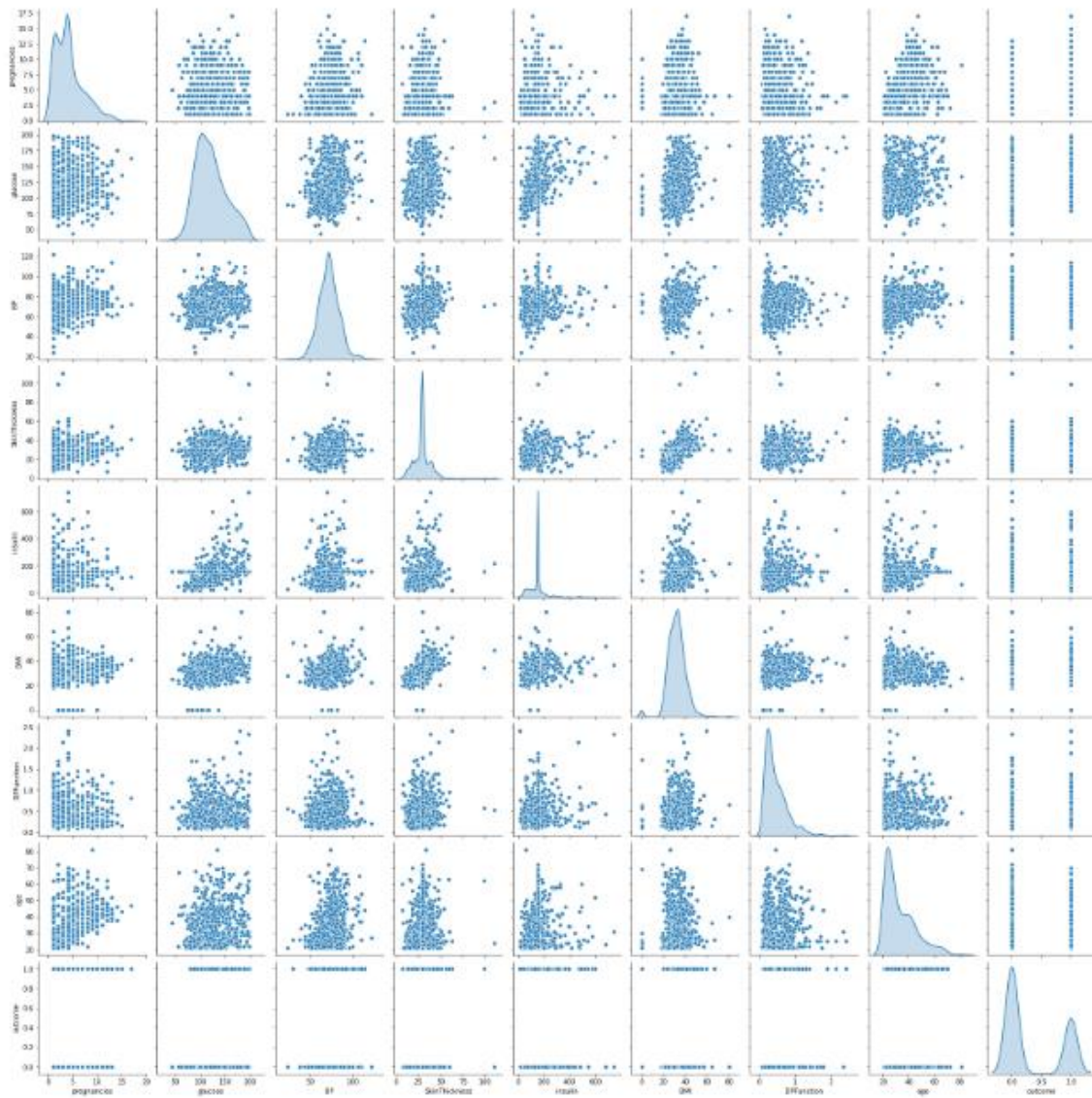
```
In [51]: dataset.hist(figsize = (10,10), bins=25)
```

```
Out[51]: array([[<matplotlib.axes._subplots.AxesSubplot object at 0x000002281CC1DC48>,  
<matplotlib.axes._subplots.AxesSubplot object at 0x000002281C9587C8>,  
<matplotlib.axes._subplots.AxesSubplot object at 0x000002281C96CA48>],  
[<matplotlib.axes._subplots.AxesSubplot object at 0x000002281C99AC48>,  
<matplotlib.axes._subplots.AxesSubplot object at 0x000002281C9D9E88>,  
<matplotlib.axes._subplots.AxesSubplot object at 0x000002281CA0E248>],  
[<matplotlib.axes._subplots.AxesSubplot object at 0x000002281CA46288>,  
<matplotlib.axes._subplots.AxesSubplot object at 0x000002281CA7F388>,  
<matplotlib.axes._subplots.AxesSubplot object at 0x000002281CA86F48>]],  
dtype=object)
```



```
In [52]: sns.pairplot(dataset, diag_kind='kde')
```

```
Out[52]: <seaborn.axisgrid.PairGrid at 0x2281d2d3648>
```





```
In [54]: dataset.isnull().values.any()
```

```
Out[54]: False
```

```
In [55]: X=dataset.iloc[:, 0:8].values  
y=dataset.iloc[:, 8].values
```

```
In [56]: X
```

```
Out[56]: array([[2.00e+00, 1.38e+02, 6.20e+01, ..., 3.36e+01, 1.27e-01, 4.70e+01],  
                [4.00e+00, 8.40e+01, 8.20e+01, ..., 3.82e+01, 2.33e-01, 2.30e+01],  
                [4.00e+00, 1.45e+02, 7.20e+01, ..., 4.42e+01, 6.30e-01, 3.10e+01],  
                ...,  
                [6.00e+00, 8.50e+01, 7.80e+01, ..., 3.12e+01, 3.82e-01, 4.20e+01],  
                [4.00e+00, 1.29e+02, 1.10e+02, ..., 6.71e+01, 3.19e-01, 2.60e+01],  
                [2.00e+00, 8.10e+01, 7.20e+01, ..., 3.01e+01, 5.47e-01, 2.50e+01]])
```

```
In [57]: y
```

```
Out[57]: array([[2.00e+00, 1.38e+02, 6.20e+01, ..., 3.36e+01, 1.27e-01, 4.70e+01],  
                [4.00e+00, 8.40e+01, 8.20e+01, ..., 3.82e+01, 2.33e-01, 2.30e+01],  
                [4.00e+00, 1.45e+02, 7.20e+01, ..., 4.42e+01, 6.30e-01, 3.10e+01],  
                ...,  
                [6.00e+00, 8.50e+01, 7.80e+01, ..., 3.12e+01, 3.82e-01, 4.20e+01],  
                [4.00e+00, 1.29e+02, 1.10e+02, ..., 6.71e+01, 3.19e-01, 2.60e+01],  
                [2.00e+00, 8.10e+01, 7.20e+01, ..., 3.01e+01, 5.47e-01, 2.50e+01]])
```

```
In [58]: co_relations=dataset.corr()  
print(co_relations)
```

	pregnancies	glucose	BP	SkinThickness	insulin	\
pregnancies	1.000000	0.147316	0.247341	0.131514	0.063534	
glucose	0.147316	1.000000	0.199498	0.208309	0.406556	
BP	0.247341	0.199498	1.000000	0.202711	0.073282	
SkinThickness	0.131514	0.208309	0.202711	1.000000	0.179486	
insulin	0.063534	0.406556	0.073282	0.179486	1.000000	
BMI	0.067306	0.243739	0.228055	0.465059	0.170183	
DPFunction	-0.011554	0.124141	0.012466	0.091822	0.096155	
age	0.517086	0.259805	0.323659	0.133270	0.089810	
outcome	0.250023	0.488020	0.174184	0.205527	0.207696	

	BMI	DPFunction	age	outcome
pregnancies	0.067306	-0.011554	0.517086	0.250023
glucose	0.243739	0.124141	0.259805	0.488020
BP	0.228055	0.012466	0.323659	0.174184
SkinThickness	0.465059	0.091822	0.133270	0.205527
insulin	0.170183	0.096155	0.089810	0.207696
BMI	1.000000	0.125719	0.038987	0.276726
DPFunction	0.125719	1.000000	0.026569	0.155459
age	0.038987	0.026569	1.000000	0.236509
outcome	0.276726	0.155459	0.236509	1.000000

```

In [59]: Having_diabetes=len(dataset.loc[dataset['outcome']==1])
        Not_having_diabetes=len(dataset.loc[dataset['outcome']==0])

        #Viewing the value of either having or not having diabetes

        (Having_diabetes,
         Not_having_diabetes)

Out[59]: (684, 1316)

In [60]: # from sklearn.preprocessing import StandardScaler
        # stdScaler=StandardScaler()
        # x = stdScaler.fit_transform(x)

In [61]: from sklearn.model_selection import train_test_split
        featured_columns = ['pregnancies', 'glucose', 'BP', 'SkinThickness', 'insulin', 'BMI', 'DPFunction', 'age']
        predicted_class = ['outcome']

        X = dataset[featured_columns].values
        y = dataset[predicted_class].values

        X_train, X_test, y_train, y_test = train_test_split(X,y, train_size=0.6, random_state=0)

In [62]: from sklearn.preprocessing import StandardScaler
        stdScaler=StandardScaler()
        X_train = stdScaler.fit_transform(X_train)
        X_test= stdScaler.fit_transform(X_test)

In [63]: X_train

Out[63]: array([[ -0.76955595, -0.90973344, -0.37938655, ..., -0.74023999,
         0.27718691, -1.02576914],
        [ 1.64609053,  1.35564633,  0.45734174, ...,  0.05041387,
        -1.00344915,  1.09590217],
        [ -0.76955595, -1.65407251, -0.2120409 , ..., -0.50769473,
         0.35470725, -0.93736617],
        ...,
        [ -0.07937124, -0.87737087, -0.71407787, ..., -0.04260423,
        -0.96313858, -0.14173943],
        [ 2.33627524, -1.20099655,  0.12265042, ..., -0.26352222,
        -0.53212547,  0.21187245],
        [ 0.26572111,  0.44949442,  0.79203306, ..., -3.76332828,
         0.52215119,  3.21757348]])

```

```
In [64]: X_train.shape
```

```
Out[64]: (1200, 8)
```

```
In [65]: X_test.shape
```

```
Out[65]: (800, 8)
```

```
In [66]: y_test.shape
```

```
Out[66]: (800, 1)
```

```
In [67]: y_train.shape
```

```
Out[67]: (1200, 1)
```

```
In [ ]:
```

```
In [68]: data= dataset[['glucose','insulin','BMI','BP','outcome',]]  
data.head(20)
```

```
Out[68]:
```

	glucose	insulin	BMI	BP	outcome
0	138.0	153.743295	33.6	62.0	1
1	84.0	125.000000	38.2	82.0	0
2	145.0	153.743295	44.2	72.0	1
3	135.0	250.000000	42.3	68.0	1
4	139.0	480.000000	40.7	62.0	0
5	173.0	265.000000	46.5	78.0	0
6	99.0	153.743295	25.6	72.0	0
7	194.0	153.743295	26.1	80.0	0
8	83.0	66.000000	36.8	65.0	0
9	89.0	153.743295	33.5	90.0	0
10	99.0	153.743295	32.8	68.0	0
11	125.0	122.000000	28.9	70.0	1
12	80.0	153.743295	0.0	72.0	0
13	166.0	153.743295	26.6	74.0	0
14	110.0	153.743295	26.0	68.0	0
15	81.0	76.000000	30.1	72.0	0
16	195.0	145.000000	25.1	70.0	1
17	154.0	193.000000	29.3	74.0	0
18	117.0	71.000000	25.2	90.0	0
19	84.0	153.743295	37.2	72.0	0

```
In [70]: data= dataset[['BP','outcome']]
data.head(20)
```

Out[70]:

	BP	outcome
0	62.0	1
1	82.0	0
2	72.0	1
3	68.0	1
4	62.0	0
5	78.0	0
6	72.0	0
7	80.0	0
8	65.0	0
9	90.0	0
10	68.0	0
11	70.0	1
12	72.0	0
13	74.0	0
14	68.0	0
15	72.0	0
16	70.0	1
17	74.0	0
18	90.0	0
19	72.0	0

```
In [72]: X_train
```

```
Out[72]: array([[ -0.76955595, -0.90973344, -0.37938655, ..., -0.74023999,
         0.27718691, -1.02576914],
        [ 1.64609053,  1.35564633,  0.45734174, ...,  0.05041387,
        -1.00344915,  1.09590217],
        [ -0.76955595, -1.65407251, -0.2120409 , ..., -0.50769473,
         0.35470725, -0.93736617],
        ...,
        [ -0.07937124, -0.87737087, -0.71407787, ..., -0.04260423,
        -0.96313858, -0.14173943],
        [ 2.33627524, -1.20099655,  0.12265042, ..., -0.26352222,
        -0.53212547,  0.21187245],
        [ 0.26572111,  0.44949442,  0.79203306, ..., -3.76332828,
         0.52215119,  3.21757348]])
```

```
In [73]: X_train
dataset.head()
```

```
Out[73]:
```

	pregnancies	glucose	BP	SkinThickness	insulin	BMI	DPFunction	age	outcome
0	2	138.0	62.0	35.000000	153.743295	33.6	0.127	47	1
1	4	84.0	82.0	31.000000	125.000000	38.2	0.233	23	0
2	4	145.0	72.0	29.341275	153.743295	44.2	0.630	31	1
3	4	135.0	68.0	42.000000	250.000000	42.3	0.365	24	1
4	1	139.0	62.0	41.000000	480.000000	40.7	0.536	21	0

```
In [74]: # from sklearn.preprocessing import Imputer
# fill_to_zero= Imputer(missing_values=0, strategy="median")
# X_train = fill_to_zero.fit_transform(X_train)
```

```
In [75]: from sklearn.neighbors import KNeighborsClassifier
```

```
test_scores = []
train_scores = []

for i in range(1,70):

    knn = KNeighborsClassifier(i)
    knn.fit(X_train,y_train)

    train_scores.append(knn.score(X_train,y_train))
    test_scores.append(knn.score(X_test,y_test))
```

```
C:\ProgramData\Anaconda3\lib\site-packages\ipykernel_launcher.py:10: DataConversionWarning: A column-vector y was passed whe
n a 1d array was expected. Please change the shape of y to (n_samples, ), for example using ravel().
# Remove the CWD from sys.path while we load stuff.
C:\ProgramData\Anaconda3\lib\site-packages\ipykernel_launcher.py:10: DataConversionWarning: A column-vector y was passed whe
n a 1d array was expected. Please change the shape of y to (n_samples, ), for example using ravel().
# Remove the CWD from sys.path while we load stuff.
C:\ProgramData\Anaconda3\lib\site-packages\ipykernel_launcher.py:10: DataConversionWarning: A column-vector y was passed whe
n a 1d array was expected. Please change the shape of y to (n_samples, ), for example using ravel().
# Remove the CWD from sys.path while we load stuff.
C:\ProgramData\Anaconda3\lib\site-packages\ipykernel_launcher.py:10: DataConversionWarning: A column-vector y was passed whe
n a 1d array was expected. Please change the shape of y to (n_samples, ), for example using ravel().
# Remove the CWD from sys.path while we load stuff.
C:\ProgramData\Anaconda3\lib\site-packages\ipykernel_launcher.py:10: DataConversionWarning: A column-vector y was passed whe
n a 1d array was expected. Please change the shape of y to (n_samples, ), for example using ravel().
# Remove the CWD from sys.path while we load stuff.
C:\ProgramData\Anaconda3\lib\site-packages\ipykernel_launcher.py:10: DataConversionWarning: A column-vector y was passed whe
n a 1d array was expected. Please change the shape of y to (n_samples, ), for example using ravel().
# Remove the CWD from sys.path while we load stuff.
```

```
In [76]: max_train_score = max(train_scores)
train_scores_ind = [i for i, v in enumerate(train_scores) if v == max_train_score]
print('Max_train_score {} % and k = {}'.format(max_train_score*100,list(map(lambda x: x+1, train_scores_ind))))

Max_train_score 100.0 % and k = [1]
```

```
In [77]: max_test_score = max(test_scores)
test_scores_ind = [i for i, v in enumerate(test_scores) if v == max_test_score]
print('Max test score {} % and k = {}'.format(max_test_score*100,list(map(lambda x: x+1, test_scores_ind))))

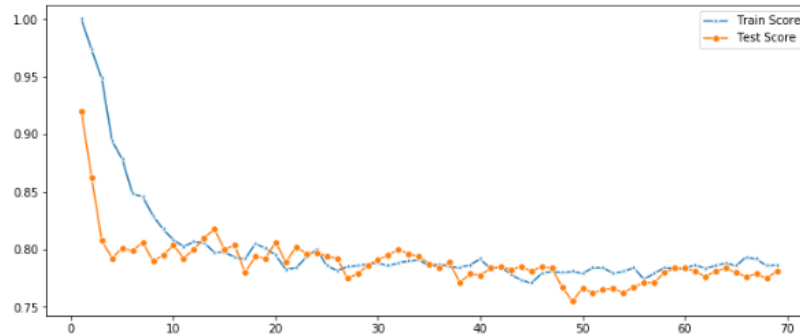
Max test score 92.0 % and k = [1]
```

```
In [78]: import seaborn as sns
plt.figure(figsize=(12,5))
p = sns.lineplot(range(1,70),train_scores,marker='*',label='Train Score')
p = sns.lineplot(range(1,70),test_scores,marker='o',label='Test Score')
```

```
In [77]: max_test_score = max(test_scores)
test_scores_ind = [i for i, v in enumerate(test_scores) if v == max_test_score]
print('Max test score {} % and k = {}'.format(max_test_score*100, list(map(lambda x: x+1, test_scores_ind))))

Max test score 92.0 % and k = [1]
```

```
In [78]: import seaborn as sns
plt.figure(figsize=(12,5))
p = sns.lineplot(range(1,70),train_scores,marker='*',label='Train Score')
p = sns.lineplot(range(1,70),test_scores,marker='o',label='Test Score')
```



```
In [79]: knn = KNeighborsClassifier(1)
model=knn.fit(X_train,y_train)
knn.score(X_test,y_test)
# knn.score(x_train,y_train)
```

C:\ProgramData\Anaconda3\lib\site-packages\ipykernel\_launcher.py:3: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please change the shape of y to (n\_samples, ), for example using ravel().  
This is separate from the ipykernel package so we can avoid doing imports until

Out[79]: 0.92

```
In [80]: predictions = model.predict(X_test)
```

```
In [81]: from sklearn.metrics import confusion_matrix
matrix = confusion_matrix(y_test, predictions)
print(matrix)
```

```
[[510  37]
 [ 27 226]]
```

```
In [82]: # k-Nearest neighbour classifier
from sklearn.metrics import classification_report
print(classification_report(y_test, predictions))
```

	precision	recall	f1-score	support
0	0.95	0.93	0.94	547
1	0.86	0.89	0.88	253
accuracy			0.92	800
macro avg	0.90	0.91	0.91	800
weighted avg	0.92	0.92	0.92	800

```

In [83]: from sklearn.tree import DecisionTreeRegressor

In [84]: regressor= DecisionTreeRegressor(random_state=2)

In [85]: modelDic=regressor.fit(X_train,y_train)

In [ ]:

In [86]: regressor.score(X_train,y_train)

Out[86]: 1.0

In [87]: predictDic = modelDic.predict(X_test)

In [88]: matrix = confusion_matrix(y_test, predictDic)
print(matrix)

[[500  47]
 [ 47 206]]

In [89]: #Decision tree classifier
print(classification_report(y_test, predictDic))


              precision    recall  f1-score   support

     0       0.91       0.91       0.91         547
     1       0.81       0.81       0.81         253

 accuracy          0.88          0.88          0.88         800
 macro avg       0.86       0.86       0.86         800
 weighted avg    0.88       0.88       0.88         800

In [90]: dictions = model.predict([[1, 79, 60, 42, 48, 43.5, 0.678, 23], [17,163,72,41,114,40.9,0.817,41], [3,148,66,25,0,32.5,0.256,22]])
dictions

```



```

Out[90]: array([0, 0, 1], dtype=int64)

```



```
In [91]: # predictions = model.predict(x_test)
# predictions
```

```
In [92]: dataset.tail(20)
```

Out[92]:

	pregnancies	glucose	BP	SkinThickness	insulin	BMI	DPFunction	age	outcome
1980	17	163.0	72.0	41.000000	114.000000	40.9	0.817	47	1
1981	4	151.0	90.0	38.000000	153.743295	29.7	0.294	36	0
1982	7	102.0	74.0	40.000000	105.000000	37.2	0.204	45	0
1983	4	114.0	80.0	34.000000	285.000000	44.2	0.167	27	0
1984	2	100.0	64.0	23.000000	153.743295	29.7	0.368	21	0
1985	4	131.0	88.0	29.341275	153.743295	31.6	0.743	32	1
1986	6	104.0	74.0	18.000000	156.000000	29.9	0.722	41	1
1987	3	148.0	66.0	25.000000	153.743295	32.5	0.256	22	0
1988	4	120.0	68.0	29.341275	153.743295	29.6	0.709	34	0
1989	4	110.0	66.0	29.341275	153.743295	31.9	0.471	29	0
1990	3	111.0	90.0	12.000000	78.000000	28.4	0.495	29	0
1991	6	102.0	82.0	29.341275	153.743295	30.8	0.180	36	1
1992	6	134.0	70.0	23.000000	130.000000	35.4	0.542	29	1
1993	2	87.0	72.0	23.000000	153.743295	28.9	0.773	25	0
1994	1	79.0	60.0	42.000000	48.000000	43.5	0.678	23	0
1995	2	75.0	64.0	24.000000	55.000000	29.7	0.370	33	0
1996	8	179.0	72.0	42.000000	130.000000	32.7	0.719	36	1
1997	6	85.0	78.0	29.341275	153.743295	31.2	0.382	42	0
1998	4	129.0	110.0	46.000000	130.000000	67.1	0.319	26	1
1999	2	81.0	72.0	15.000000	76.000000	30.1	0.547	25	0

```
In [ ]:
```



```

In [98]: from sklearn.ensemble import RandomForestClassifier

In [99]: classifier=RandomForestClassifier(random_state=1)

In [100]: DiabRand=classifier.fit(X_train,y_train)
C:\ProgramData\Anaconda3\lib\site-packages\ipykernel_launcher.py:1: DataConversionWarning: A column-vector y was passed when a
1d array was expected. Please change the shape of y to (n_samples,), for example using ravel().
"""Entry point for launching an IPython kernel.

In [101]: RandModel=DiabRand.predict(X_test)

In [102]: matrix = confusion_matrix(y_test, RandModel)
print(matrix)

[[530  17]
 [ 40 213]]

In [103]: #Random forest Classifier
print(classification_report(y_test, RandModel))

```

	precision	recall	f1-score	support
0	0.93	0.97	0.95	547
1	0.93	0.84	0.88	253
accuracy			0.93	800
macro avg	0.93	0.91	0.92	800
weighted avg	0.93	0.93	0.93	800

```
In [105]: MLP_Classifier = MLPClassifier(hidden_layer_sizes = (2000,),
                                         activation='relu',
                                         verbose=1, solver='adam',
                                         batch_size=30,
                                         learning_rate = 'constant',
                                         learning_rate_init = 0.0001,
                                         max_iter= 500)
```

```
In [106]: MLP_Classifier
```

```
Out[106]: MLPClassifier(activation='relu', alpha=0.0001, batch_size=30, beta_1=0.9,
                        beta_2=0.999, early_stopping=False, epsilon=1e-08,
                        hidden_layer_sizes=(2000,), learning_rate='constant',
                        learning_rate_init=0.0001, max_fun=15000, max_iter=500,
                        momentum=0.9, n_iter_no_change=10, nesterovs_momentum=True,
                        power_t=0.5, random_state=None, shuffle=True, solver='adam',
                        tol=0.0001, validation_fraction=0.1, verbose=1, warm_start=False)
```

```
In [107]: NNmodel = MLP_Classifier.fit(X_train, y_train)
```

```
Iteration 485, loss = 0.20110920
Iteration 486, loss = 0.19986556
Iteration 487, loss = 0.19967747
Iteration 488, loss = 0.19944309
Iteration 489, loss = 0.19955415
Iteration 490, loss = 0.19929254
Iteration 491, loss = 0.19848740
Iteration 492, loss = 0.19871558
Iteration 493, loss = 0.19820332
Iteration 494, loss = 0.19764298
Iteration 495, loss = 0.19724642
Iteration 496, loss = 0.19675900
Iteration 497, loss = 0.19629814
Iteration 498, loss = 0.19658540
Iteration 499, loss = 0.19540141
Iteration 500, loss = 0.19530016
```

```
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\normalization\multilayer_perceptron.py:571: ConvergenceWarning: Stochastic Optimizer: Maximum iterations (500) reached and the optimization hasn't converged yet.
% self.max_iter, ConvergenceWarning)
```

```
In [108]: MLPdiabetes=NNmodel.predict(X_test)
MLPdiabetes
```

```
Out[108]: array([0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 1, 1,
0, 0, 0, 1, 1, 1, 0, 1, 1, 0, 0, 1, 0, 1, 1, 0, 0, 0, 0, 1, 1, 0,
1, 1, 0, 0, 1, 1, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0,
0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 1,
0, 1, 1, 1, 0, 0, 1, 0, 0, 0, 1, 1, 0, 1, 0, 0, 1, 0, 0, 0, 0, 1,
1, 1, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0,
0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 1, 0, 1, 0, 1, 0,
0, 0, 0, 0, 0, 1, 0, 1, 1, 0, 0, 1, 0, 1, 1, 0, 0, 0, 0, 1, 1, 0,
0, 0, 0, 1, 0, 0, 1, 0, 1, 0, 1, 1, 1, 0, 0, 1, 0, 1, 0, 0, 0,
1, 0, 1, 1, 1, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0,
1, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0,
1, 0, 0, 1, 1, 0, 0, 1, 0, 0, 1, 1, 0, 1, 0, 0, 0, 1, 0, 0, 1,
```

```
0, 1, 1, 1, 1, 0, 1, 0], dtype=int04)

In [109]: matrix = confusion_matrix(y_test, MLPdiabetes)
print(matrix)

[[482  65]
 [ 52 201]]
```

```
In [110]: #Neural network MLP classifier result
print(classification_report(y_test, MLPdiabetes))
```

	precision	recall	f1-score	support
0	0.90	0.88	0.89	547
1	0.76	0.79	0.77	253
accuracy			0.85	800
macro avg	0.83	0.84	0.83	800
weighted avg	0.86	0.85	0.85	800

```
In [111]: import pickle
from sklearn.externals import joblib
filename = 'diabetes_model.pk1'
joblib.dump(model, filename)
```

C:\ProgramData\Anaconda3\lib\site-packages\sklearn\externals\joblib\\_\_init\_\_.py:15: FutureWarning: sklearn.externals.joblib is deprecated in 0.21 and will be removed in 0.23. Please import this functionality directly from joblib, which can be installed with: pip install joblib. If this warning is raised when loading pickled models, you may need to re-serialize those models with scikit-learn 0.21+.

```
warnings.warn(msg, category=FutureWarning)
```

Out[111]: ['diabetes\_model.pk1']