



**Tribhuvan University**

**Institute of Science and Technology**

**Bhaktapur Multiple Campus, Bhaktapur**

**Project Report On**

**Diabetes Prediction System for Female using Support Vector Machine and  
Logistic Regression**

Under Supervision of:

**Surya Bam**

Department of CSIT, BMC

**Submitted by:**

Manish Shrestha (23245/076)

Amresh Kumar Singh (24513/076)

Binod Paudel (23233/076)

**Submitted to:**

Tribhuvan University

Institute of Science and Technology

March, 2024



**Tribhuvan University**

**Institute of Science and Technology**

**Bhaktapur Multiple Campus, Bhaktapur**

## **SUPERVISOR'S RECOMMENDATION**

I hereby recommend that the project work report entitled **Diabetes Prediction for Female using Support Vector Machine and Logistic Regression** submitted by **Mr. Manish Shrestha** (23245/076), **Mr. Binod Paudel** (23233/076) and **Mr. Amresh Kumar Singh** (24513/076) of **Bhaktapur Multiple Campus, Doodhpati, Bhaktapur** is prepared under my supervision as per the procedure and format requirement laid by the Faculty of B.Sc. CSIT, Tribhuvan University, in the partial fulfillment of the requirements for the Bachelor's Degree in Computer Science and Information Technology (B. Sc. CSIT). I, therefore, recommend the project work report for evaluation.

.....

**Surya Bam**

Bhaktapur Multiple Campus

B.Sc. CSIT Department



**Tribhuvan University**

**Institute of Science and Technology**

**Bhaktapur Multiple Campus, Bhaktapur**

## **LETTER OF APPROVAL**

We certify that we have read this project work report and, in our opinion, it is appreciable for the scope and quality as a project work in the partial fulfillment of the requirements of Four Year Bachelor Degree of Science in Computer Science and Information Technology.

### **Evaluation Committee**

\_\_\_\_\_  
**Mr. Sushant Paudel**

Co-ordinator, B.Sc. CSIT Department

Bhaktapur Multiple Campus

\_\_\_\_\_  
Internal Examiner

\_\_\_\_\_  
**Surya Bam**

(Supervisor)

Bsc CSIT, BMC

\_\_\_\_\_  
External Examiner

## ACKNOWLEDGEMENT

We owe my most profound appreciation to Bhaktapur Multiple Campus for giving us a chance to work on this project as part of our syllabus.

Special thanks to our supervisor, **Mr. Surya Bam** for his consistent support, guidance and feedback throughout the report's creation. We are generously obligated to him for providing this excellent opportunity to expand our knowledge. It helped a lot to realize what we study for.

We want to sincerely thank our respected coordinator, **Mr. Sushant Paudel**. We are thankful to all the faculty members and lecturer/teachers who helped us, directly or indirectly, in completing this project.

We would also to extend our appreciation to our team members for their hard work and dedication in completing this project.

Finally, last but by no means least, we would like to express our sincere gratitude to all those individuals, families, friends, and colleagues for supporting and helping us a lot in finalizing this project within the limited time frame by providing valuable insights and feedback on the report.

Thank You,

Manish Shrestha (23245/076)

Binod Paudel (23233/076)

Amresh Kumar Singh (24513)

## ABSTRACT

Diabetes mellitus, especially Type 2, is becoming a major global public health concern because it affects women more frequently than men for a variety of physiological and socioeconomic reasons. Diabetes-related problems must be managed and prevented, and this requires early identification and prediction. In order to predict diabetes in females, this study provides an in-depth analysis of two well-known machine learning techniques, Support Vector Machine (SVM) and Logistic Regression (LR). We deeply trained, evaluated SVM and LR models using a dataset that included several physiological and medical characteristics of female patients, such as pregnancies, glucose, blood pressure, skin thickness, insulin, body mass index (BMI), Diabetes pedigree function and age. Accuracy, precision, recall and F1-score were used to evaluate the models' performance. Our results suggest that although both models demonstrate excellent prediction ability, SVM exceeded LR slightly in terms overall accuracy and precision. But LR showed to be more interpretable and simple to use, which made it a useful tool for clinical decision-making. The study emphasizes how crucial it is to choose the right machine learning models depending on the particular demands of the job at hand and the trade-offs between usability and efficiency measures in healthcare contexts. It also demonstrates how machine learning algorithms could enhance predictive healthcare and open up the possibilities to active and individualized management of diabetes in women.

*Keywords: Diabetes Mellitus, Type 2 Diabetes, Machine Learning, Support Vector Machine (SVM), Logistic Regression (LR), Prediction Model, Accuracy, Clinical Decision-Making, Predictive Healthcare*

# TABLE OF CONTENT

<b>ACKNOWLEDGEMENT .....</b>	<b>i</b>
<b>ABSTRACT .....</b>	<b>ii</b>
<b>LIST OF FIGURE.....</b>	<b>v</b>
<b>LIST OF TABLE.....</b>	<b>vi</b>
<b>LIST OF ABBREVIATIONS.....</b>	<b>vii</b>
<b>Chapter 1: Introduction.....</b>	<b>1</b>
1.1    Introduction .....	1
1.2    Problem Statement.....	1
1.3    Objectives .....	2
1.4    Scope and Limitation.....	2
1.4.1    Scope .....	2
1.4.2    Limitation .....	2
1.5    Methodology .....	2
1.6    Report Organization .....	4
<b>Chapter 2: Background Study and Literature Review .....</b>	<b>5</b>
1.1    Background Study .....	5
1.2    Literature Review .....	5
<b>Chapter 3: System Analysis.....</b>	<b>8</b>
3.1 System Analysis .....	8
3.1.1 Requirement Analysis .....	8
3.1.2 Feasibility Analysis .....	9
3.1.3 Analysis Details.....	11
<b>Chapter 4: System Design.....</b>	<b>14</b>
4.1    Design.....	14
4.1.1 System Architecture Diagram .....	14
4.1.2 Refinement of Class and Object Diagram, Sequence Diagram and Activity Diagram .....	15
4.1.3 Component Diagram .....	18
4.1.4 Deployment Diagram .....	19
4.1.5 Interface Design.....	20
4.2 Algorithm Details .....	21

<b>Chapter 5: System Implementation and Testing .....</b>	<b>25</b>
4.2 Implementation.....	25
4.2.1 Tools used.....	25
4.2.2 Implementation of Module .....	25
4.3 Testing .....	34
4.3.1 Test Cases for Unit Testing .....	34
4.3.2 Test cases for System Testing .....	37
5.3 Result Analysis.....	40
<b>Chapter 6: Conclusion .....</b>	<b>43</b>
6.1 Conclusion.....	43
6.2 Future Recommendation .....	43
<b>Reference .....</b>	<b>44</b>
<b>Appendix .....</b>	<b>45</b>

## LIST OF FIGURE

Figure 1.1: Agile Methodology.....	3
Figure 3.1: Use-Case Diagram for Functional Requirement .....	9
Figure 3.2: Class and Object Diagram for Object Modeling .....	11
Figure 3.3: Sequence Diagram for Dynamic Modeling .....	12
Figure 3.4: Activity Diagram for Process Modeling .....	13
Figure 4.1: System Architecture .....	14
Figure 4.2: Refinement of Class and Object Diagram.....	15
Figure 4.3: Refinement of Sequence Diagram.....	16
Figure 4.4: Refinement of Activity Diagram.....	17
Figure 4.5: Component Diagram .....	18
Figure 4.6: Deployment Diagram .....	19
Figure 4.7: Home Page .....	20
Figure 4.8: Prediction Page.....	20
Figure 4.9: SVM Diagram .....	22
Figure 5.1: Correlation Matrix .....	27
Figure 5.2: Redirecting to the Homepage .....	35
Figure 5.3: Redirected to the About us Page .....	36
Figure 5.4: Redirected to the Prediction Page .....	36
Figure 5.5: Redirected to Accuracy Page .....	37
Figure 5.6: Invalid Input .....	39
Figure 5.7: Showing diabetes result.....	39
Figure 5.8: Showing No diabetes result.....	39
Figure 5.9: Confusion Matrix of SVM .....	40
Figure 5.10: Confusion Matrix of LR.....	40



## LIST OF TABLE

Table 5. 1: Unit Testing for the System.....	34
Table 5. 2: System Testing for Diabetes Prediction .....	37
Table 5. 3: Comparison Table between SVM and LR.....	42

## **LIST OF ABBREVIATIONS**

BMI	Body Mass Index
LR	Logistic Regression
PCOs	Polycystic Ovary Syndrome
SVM	Support Vector Machine

# **Chapter 1: Introduction**

## **1.1 Introduction**

Diabetes is a major health problem that affects millions of people around the world, and it's especially tricky because it can lead to other serious problems if not managed or care properly. Type 2 Diabetes is the most common form which often sneaks up on people because of factors like being overweight, not exercising, or having a family history of diabetes. Women face unique risk when it comes to diabetes, including certain situation like polycystic ovary syndrome also called PCOs (a hormonal disorder) and pregnancy-related diabetes, which make it really important to figure out who might get diabetes early on.

In the world of medical and healthcare, there is growing use of smart computer programs, known as machine learning, to help doctors predict disease like diabetes before they happen. Two of these smart machine learning programs, called Support Vector Machine (SVM) and Logistic Regression (LR), are really good classifier to predict the disease using medical information to find patterns that might suggest people who is at risk of developing diabetes.

Even though these machine learning program can be super helpful, it's not always clear that classify the best result. That's why in this study, we decided to compare how well SVM and LR can predict diabetes in women by using medical information like pregnancies, glucose, blood pressure, skin thickness, insulin, body mass index (BMI), Diabetes pedigree function and age. We wanted to see which program does better job and how they can be used by doctors to help prevent diabetes in women. By doing the comparison, we hope to make it easier for medical field professional to pick the right tool for predicting diabetes early on in women, leading to better health management and preventing serious problem.

## **1.2 Problem Statement**

Diabetes is a big health issue that's growing fast, especially for women. Prediction it early is super important to stop it from getting worse. Normally, doctors check blood sugar to find diabetes, but this method doesn't always predict it early enough. Women have some special risks linked to their hormones, which these tests might not always spot. That's where Machine learning model like Support Vector Machine (SVM) and Logistic Regression (LR)

come in which detect early. They can look at a lot of health information and find early signs of diabetes. These models work differently and might be better at seeing why women might get diabetes. We want to see which models, SVM or LR, is better at finding diabetes in women early. This could help doctors help women manage their health better from the start.

### **1.3 Objectives**

- To predict the diabetes for female using SVM and Logistic Regression.

### **1.4 Scope and Limitation**

#### **1.4.1 Scope**

In this project, we aim to develop a comprehensive Diabetes Prediction System using machine learning techniques. The scope includes the creation of predictive model capable of identifying individuals at a risk of developing diabetes based on various physiological and lifestyle factors. The primary objective is to design the accurate and reliable system to early detection of diabetes in women so they can get early treatment. In this project, we compare the Support Vector Machine and Logistic regression for classification to getting more accurate system which predicts diabetes or not in female patient.

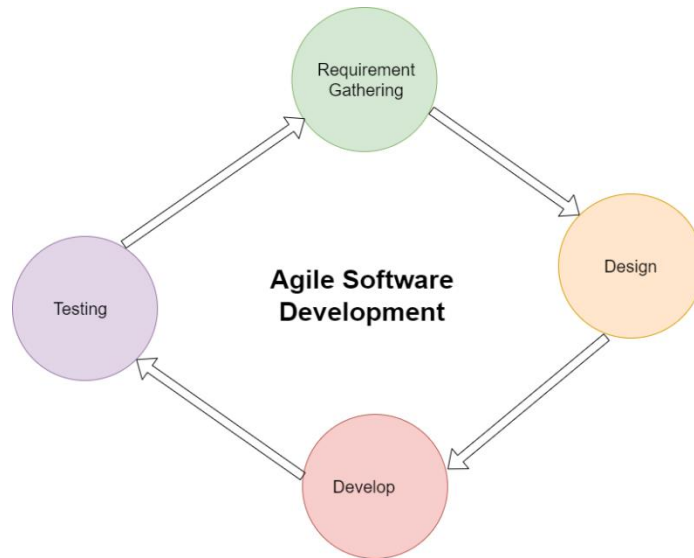
#### **1.4.2 Limitation**

In this project, we only use eight feature such as pregnancies, glucose, blood pressure, skin thickness, insulin, body mass index (BMI), Diabetes pedigree function and age which may does not give accurate prediction or classification. This project only focused on female rather than both gender i.e. female and male. We have used small dataset (769) around for training our model which may not provide more accurate results.

### **1.5 Methodology**

The Agile method was chosen as the development methodology for the project, emphasizing quick and flexible iterations in building software through small, manageable steps. It involved continuous collaboration, adaptability, improvements and feedback. The key consideration included: determining project scope, requirements, and planning the number

and duration of development iteration from the start. Agile allows quick adjustments based on changing needs to the project development.



**Figure 1.1: Agile Methodology**

Our project lifecycle consists of following phases:

1. Planning and Requirement Analysis:

In this phase, we have study the diabetes and their symptoms from different journal and research paper. We consider the medical data as the features for classification and prediction. According to our project needs, we research the dataset. We have studied different machine learning algorithm and selected SVM and logistic regression for our project.

2. Design:

In this phase, we design Use-case for determining the functional requirement and for the modeling of the system, we design the Class and object diagram for the object modeling, Sequence diagram for the dynamic modeling and activity diagram for the process modeling.

3. Development:

In this phase, we developed the User Interface using HTML, CSS and bootstrap for the project. Then we build the SVM and logistic regression model for predicting the diabetes

using Jupyter notebook. We select the Django framework for integrating the User Interface and machine learning model.

#### 4. Testing:

In this phase, various testing methods were gathered to evaluate the behavior of each task and it ensures everything works fine and fixing any problems.

### **1.6 Report Organization**

The documentation of the project has been prepared after the successful completion of the project. The initial section of the report consists of a Cover Page, Certificate Page, Acknowledgement, Abstract, Table of Contents, and lists of Abbreviations, Figures and Tables. The main report is organized in 6 Chapters, aligning with their respective headings and content. In Chapter 1: Introduction, it provides a brief overview of the project by outlining aspects such as introduction of project, problem statement, objectives, scope and limitation and developmental methodology. In Chapter 2: background Study and Literature Review, it provides the background study of the project and conducts a review of existing literature, summarizing similar projects, paper and articles. In Chapter 3: System Analysis, it focuses on analysis of system, covering requirement (i.e. functional and non-functional requirements) and feasibility analysis. Functional requirements of the system are defined through the use case diagram. A Gantt chart is used to visually illustrate the time taken for various tasks in the project. In Chapter 4: System Design, it goes deep into the system details, focusing on implementation process and designing the user interface, model architecture, and forms. It also provides insights in the algorithm employed in the system. In Chapter 5: Implementing and Testing, it discusses the implementation of the system and details of testing, including an overview of the dependencies and tools utilized for implementing the system. Chapter 6: Conclusion and Future Recommendation, it concludes and summarizes the projects and explores possibilities for future improvement of the project.

The report's final session includes the References following IEEE standards and Appendices containing system screenshots and necessary source code snippets.

## **Chapter 2: Background Study and Literature Review**

### **1.1 Background Study**

In the context of predicting diabetes in females, comparing Support Vector Machine (SVM) and Logistic Regression (LR) algorithms reveals distinct characteristics and applications of each method. SVM performs excellently in managing complex, nonlinear relationships within data, making it suitable for scenarios where the differentiation between having diabetes and not isn't straightforward. By finding the optimal boundary to separate cases, SVM can effectively classify individuals based on small patterns. However, its complexity and computational demands can be a drawback, particularly with large datasets. On the other hand, Logistic Regression offers more straightforward approach, estimating the probability that a given individual falls into a particular category (diabetes or not). This method's strength lies in its simplicity and interpretability, especially valuable in medical setting where explaining the basis of prediction is crucial. Logistic Regression tends to require fewer computational resources, making more benefits for larger datasets. Ultimately, the choice between SVM and LR for diabetes prediction in females hinges on the specific data characteristics, importance of interpretability, and available computational resources. Both algorithms have proven effective, but their optimal application depends on the context of the health data being analyzed.

### **1.2 Literature Review**

1. According to paper [2], diabetes is one of the dangerous diseases all over the world, it can cause many varieties of disorders which includes blindness etc. In this paper, they have used machine learning techniques to discover the diabetes disease as it is flexible and easy to determine whether the patient has illness or not. Their main goal of this analysis was to develop a system that can essential for people to detect the diabetes disease of the patient with accurate results. Here they used mainly 3 main algorithms Naïve Bayes, Decision Tree and SVM algorithms and compared their accuracy which is 77%, 85%, 77.3 % respectively. They also used ANN algorithm after the training process to know the reaction of the network which states whether the disease is classified correctly or not. Here they compared the precision recall and F1 score support and accuracy of all models. [2]

2. From the journal [3], one of the worst diseases in the world is diabetes. Not only is it an illness, but it also has the potential to cause various other illnesses, such as kidney, eye, and heart disorders. Patients typically have to visit a diagnostic facility, speak with their physician, then wait a day or longer to receive their reports. Furthermore, they are forced to spend money in vain each time they wish to receive their diagnosis report. A collection of metabolic diseases collectively referred to as diabetes mellitus (DM) are primarily brought on by aberrant insulin secretion and/or action. Hyperglycemia, or high blood sugar, and poor protein, lipid, and carbohydrate metabolism are the outcomes of insulin insufficiency. With a global impact on over 200 million individuals, diabetes mellitus is among the most prevalent endocrine illnesses. In the next years, there will likely be a significant increase in the start of diabetes. DM can be categorized into multiple unique kinds. However, based on the etiopathology of the condition, there are two main clinical types: type 1 diabetes (T1D) and type 2 diabetes (T2D). T2D, which accounts for 90% of all cases of diabetes, is primarily characterized by insulin resistance. [3]

3. By studying the research paper [4], a chronic illness is an ongoing disease or condition with severe consequences. One of the main negative effects of these disorders is their impact on quality of life. One of the most serious diseases, diabetes affects people all over the world. This chronic illness is one of the leading causes of adult fatalities worldwide. Chronic illnesses also have financial consequences. Governments and individuals spend a significant amount of their budgets on treating chronic illnesses. According to global diabetes data from 2013, 382 million people worldwide suffered from this illness. In 2012, it ranked as the eighth most common cause of mortality for both sexes and the fifth most common cause of death for women. Diabetes is more common in nations with higher incomes. It is predicted that there would be 693 million diabetic patients worldwide in 2045, with half of them going untreated. Furthermore, 850 million USD were spent in 2017 on diabetes patients. [4]

4. One of the most prevalent chronic metabolic illnesses in the world today is diabetes. Diabetes comes in two types: Type-1 and Type-2. A small quantity of insulin is released or none at all when the body develops Type-1 diabetes, which is caused by immune system injury to pancreatic beta cells ( $\beta$ -cells). An auto immune disorder known as type-2 diabetes occurs when the body's cells are unable to respond to insulin or when the pancreas cells are



unable to generate enough insulin to control blood glucose levels. Type-1 diabetes is brought on by a lack of insulin, which raises blood glucose levels and impairs the metabolism of proteins, lipids, and carbohydrate. This article presents a model using a fused machine learning approach for diabetes prediction. The conceptual framework consists of two types of models: Support Vector Machine (SVM) and Artificial Neural Network (ANN) models. These models analyze the dataset to determine whether a diabetes diagnosis is positive or negative. [5]

5. In this research [6], the authors use a computer-based method, specifically Artificial Intelligence (AI), to predict diabetes, a widespread and serious health issue affecting millions worldwide. They focus on a particular group, the Pima Indians, for their study. By employing a type of AI known as an Artificial Neural Network (ANN), which mimics the way human brains operate, they developed a system to foresee the risk of diabetes. Their findings were promising, showing that their system could accurately predict diabetes 92% of the time when tested on a small group. The researchers believe that if they could use more data for training their system, its accuracy could improve even further. This study highlights the potential of using AI to help in early detection of diabetes, which could significantly impact managing and treating this chronic condition. [6]

## **Chapter 3: System Analysis**

### **3.1 System Analysis**

This system is intended to predict diabetes in females based on their symptoms and personal health data. The core component of the system include: Medical Data Input Form, Diabetes Prediction Engine, etc.

#### **3.1.1 Requirement Analysis**

In our project, we required the information about the diabetes, symptoms and the relevant dataset from the trusted organization. We need a robust system which is capable of training the model, processing the data, deploying the model, etc.

Requirement Analysis consists of two types:

##### **1. Functional Requirement**

- Symptom Input Form: Users can input and update their health data for diabetes risk assessment.
- Diabetes Prediction: The system analyzes input data to predict diabetes risk and provides result.
- Diabetes Dataset: The system requires the diabetes dataset to predict the diabetes for the training the model.

##### **2. Non-Functional Requirement**

- Performance: The system responds to user requests and displays result within the seconds.
- Scalability: It capable of accommodating an increasing number of users and data without performance loss.
- Reliability: Available 24/7 with minimal downtime and capable of handling error easily.



**Figure 3.1: Use-Case Diagram for Functional Requirement**

### 3.1.2 Feasibility Analysis

#### a. Technical

The technical feasibility of creating a diabetes prediction system for female looks promising but comes with challenges. We have technology to build smart, user-friendly apps that can analyze health data to predict diabetes risks. However, getting high-quality health data while keeping it safe and privacy is tricky. Also making sure the app works well with other health systems and follows strict health data laws can be tough. Despite these problems, advances in machine learning and system development offer a strong foundation to overcome these issues and creates the helpful tool for predicting diabetes in women.

#### b. Operational

Operational feasibility for our diabetes prediction system for women looks at whether it can smoothly fit into daily use. It's about making sure everyone who needs to use or support the system is ready and willing to do so. This means the system must be easy to use, helpful, and not make more work for healthcare workers. It also depends on training people well and making sure the system works well with how healthcare is already done. If the system meets these needs and people find it useful and not too hard to add their routines, it has a good chance of being successfully adopted and used.

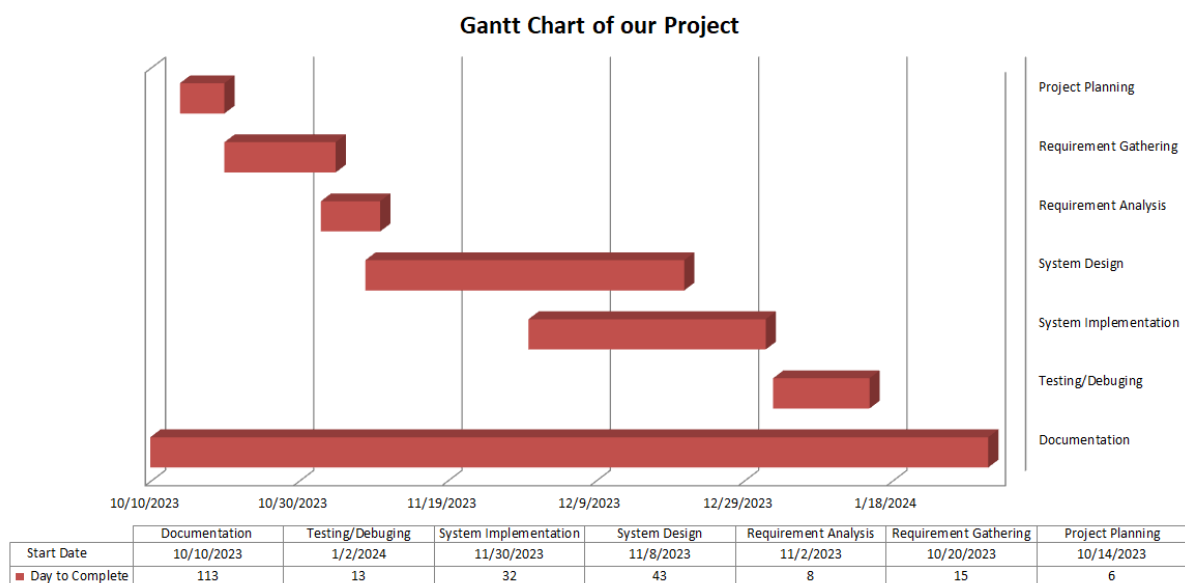
#### c. Economical

Economic feasibility looks at whether the diabetes prediction system for women makes financial sense. It's about comparing the money spent on building and running the system to money it could save or earn back by helping prevent diabetes and manage it better. That includes the cost of making the system, keeping it going, and teaching people how to use it if the system can help save on healthcare costs by predicting diabetes early or making care better, then it could be worth the investment. Finding funding and making sure the benefits outweigh the cost are keys to deciding if this project is a good financial move.

#### d. Schedule

Schedule feasibility is about figuring out if we can finish building and launching the diabetes prediction system for women in the time we've planned. It means looking at how complicated the project is, making sure we have enough people who know how to do the work, and thinking about any unexpected things that could slow us down. We need to be realistic about how long everything will take, including testing the system and teaching people how to use it. It's important to plan carefully and have some extra time just in case, to make sure we can get everything done on time.

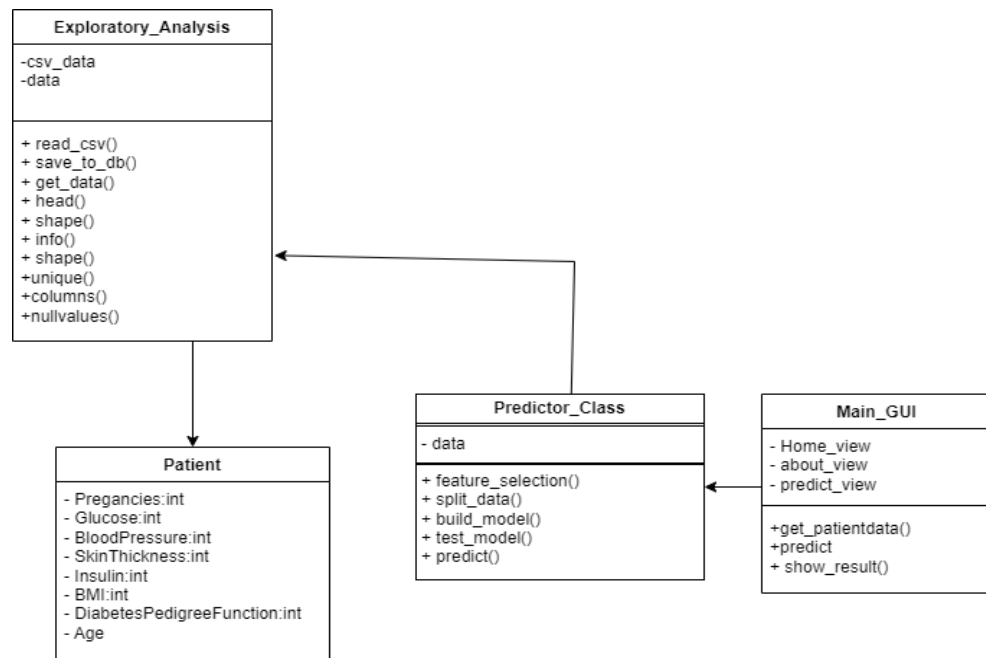
The requirements are not too complex so we complete the application development at the time of interval of 4-5 months. The work is divided as follows:



### 3.1.3 Analysis Details

We have design the three diagrams for the analysis of our system. For object modeling, we design the class and object diagram. We design the sequence diagram for dynamic modeling and activity diagram for process modeling.

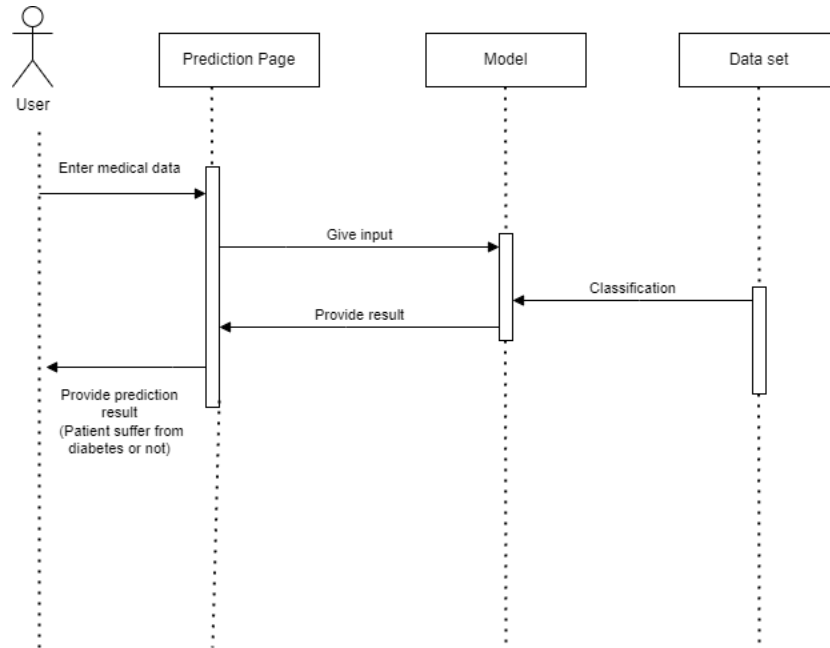
#### a. Object modeling using Class and Object Diagram



**Figure 2.2: Class and Object Diagram for Object Modeling**

We have designed the object modeling using Class and object diagram. In the above figure 3.2, there are 5 classes such as **Exploratory\_Analysis**, **Patient**, **Predictor\_Class** and **Main\_GUI**. **Exploratory\_Analysis** Class handle data analysis task such as statistical analysis, data visualization, and identifying patterns or anomalies in the data. **Patient\_Class** represents the patient information and the medical information. **Predictor\_class** implements predictive model to forecast patient result based on medical data. The Main GUI class acts as the user interface for interacting with the system integrating functionalities of the other classes.

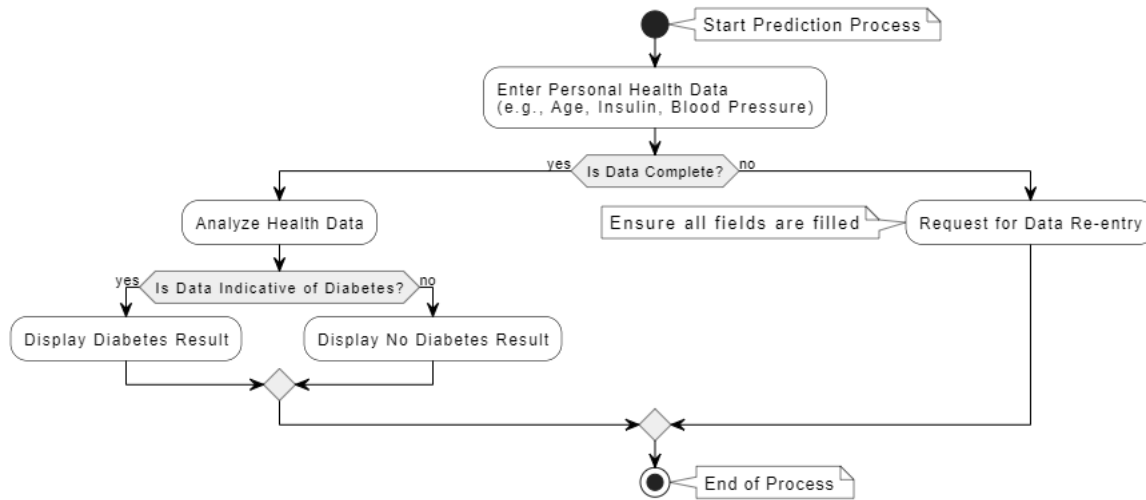
## b. Dynamic Modeling using Sequence Diagram



**Figure 3.3: Sequence Diagram for Dynamic Modeling**

We have design dynamic modeling using Sequence Diagram. In the above figure 3.3, the process starts when a user inputs medical data into the prediction page. This data is sent from the prediction page to the model. The model, equipped with a dataset, uses this information to classify and predict whether the patient is suffering from the diabetes. After that, the model completes its classification and prediction, it provides the results of its analysis. These results are then sent back to the prediction page. Finally, the prediction page displays the outcomes, informing the user whether the patient is suffering from diabetes or not. This sequence of the action illustrates a straightforward interaction flow from data entry to result presentation.

### c. Process modeling using Activity Diagram



**Figure 3.4: Activity Diagram for Process Modeling**

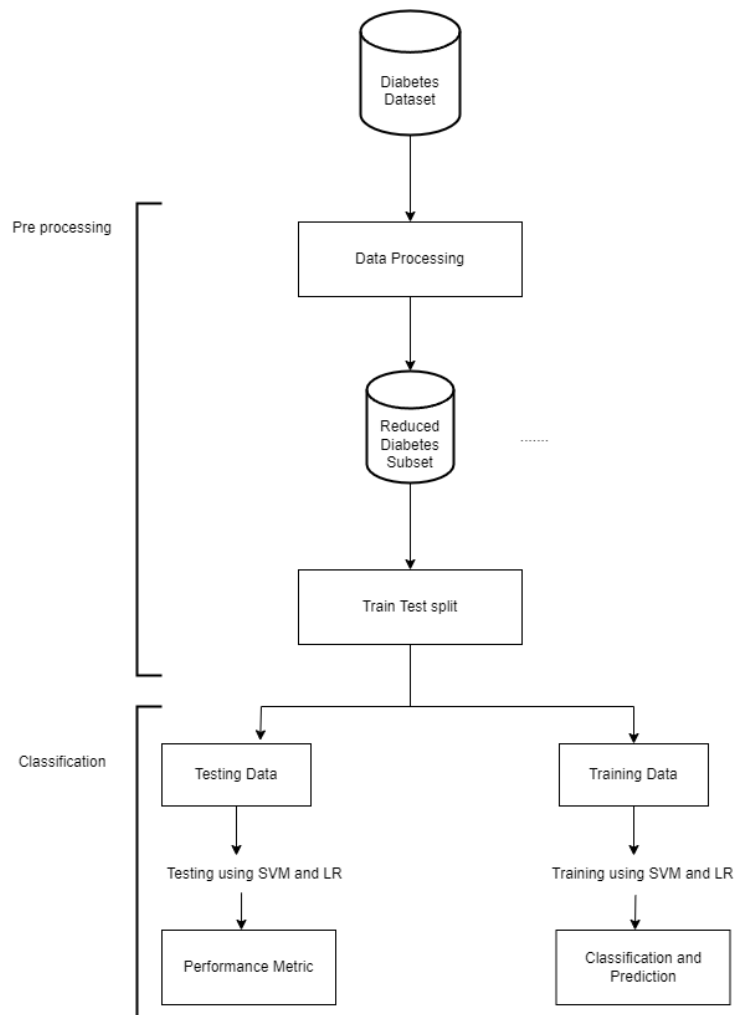
We have design process modeling using activity diagram. In the above figure, the process begins with the initiation of the diabetes prediction workflow. Then, the patient enters medical health information such as age, insulin, and blood pressure. The system checks if all required field have been completed accurately. If not, then patient should reenter the medical information. After checking, the system analyzes the entered medical data. If the Data indicate the diabetes result positively (Outcomes: 1) then display prediction results otherwise display no diabetes result. The process concludes after the prediction results are provided to user.

## Chapter 4: System Design

### 4.1 Design

We have design the System for representing architecture, Interfaces, components that are included in the system. Hence the System design can be visualize as the application of the system theory to product development

#### 4.1.1 System Architecture Diagram



**Figure 4.1: System Architecture**

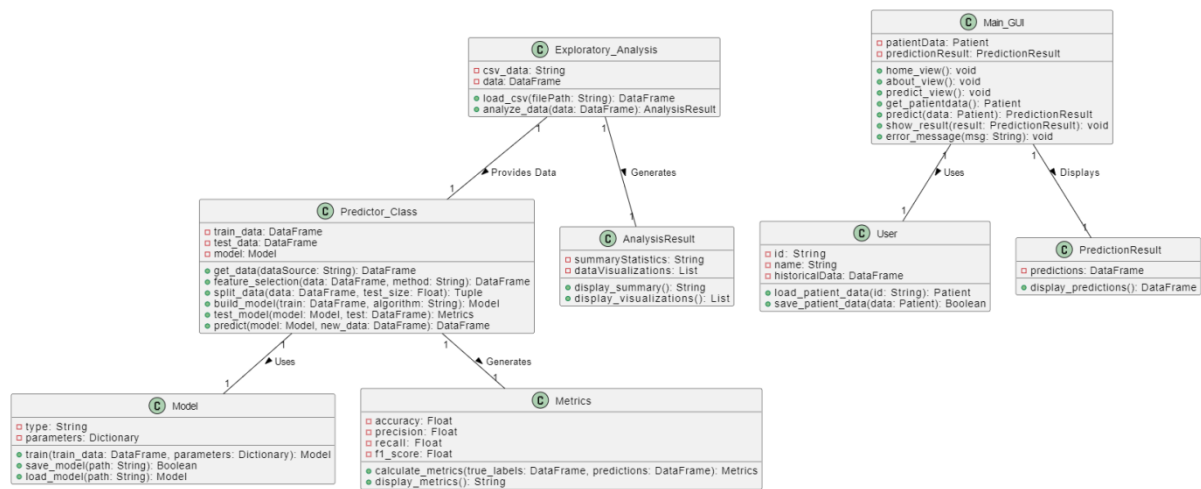
Our project's goal was to predict diabetes using two techniques: logistic regression (LR) and support vector machines (SVM). First, the system begins with a diabetes dataset. Then the data is preprocessed which may include cleaning the data, handling missing values, and



scaling the data. After preprocessing, the data is processed which may involve feature engineering, which is the process of creating new features from the existing data. The processed data is then split into a training set (80% data) and a testing set (20% data). The training set is used to train the machine learning model, and the test set is used to evaluate the performance of the model. Then the training data is used to train two machine learning models: a support vector machine (SVM) and logistic regression model. The performance of the models is evaluated using performance metrics which involve calculating the accuracy, precision, recall, and F1-score of the models. Once the best is selected , it is used to make prediction on the test data.

#### 4.1.2 Refinement of Class and Object Diagram, Sequence Diagram and Activity Diagram

##### 1. Refinement of Class and Object Diagram:

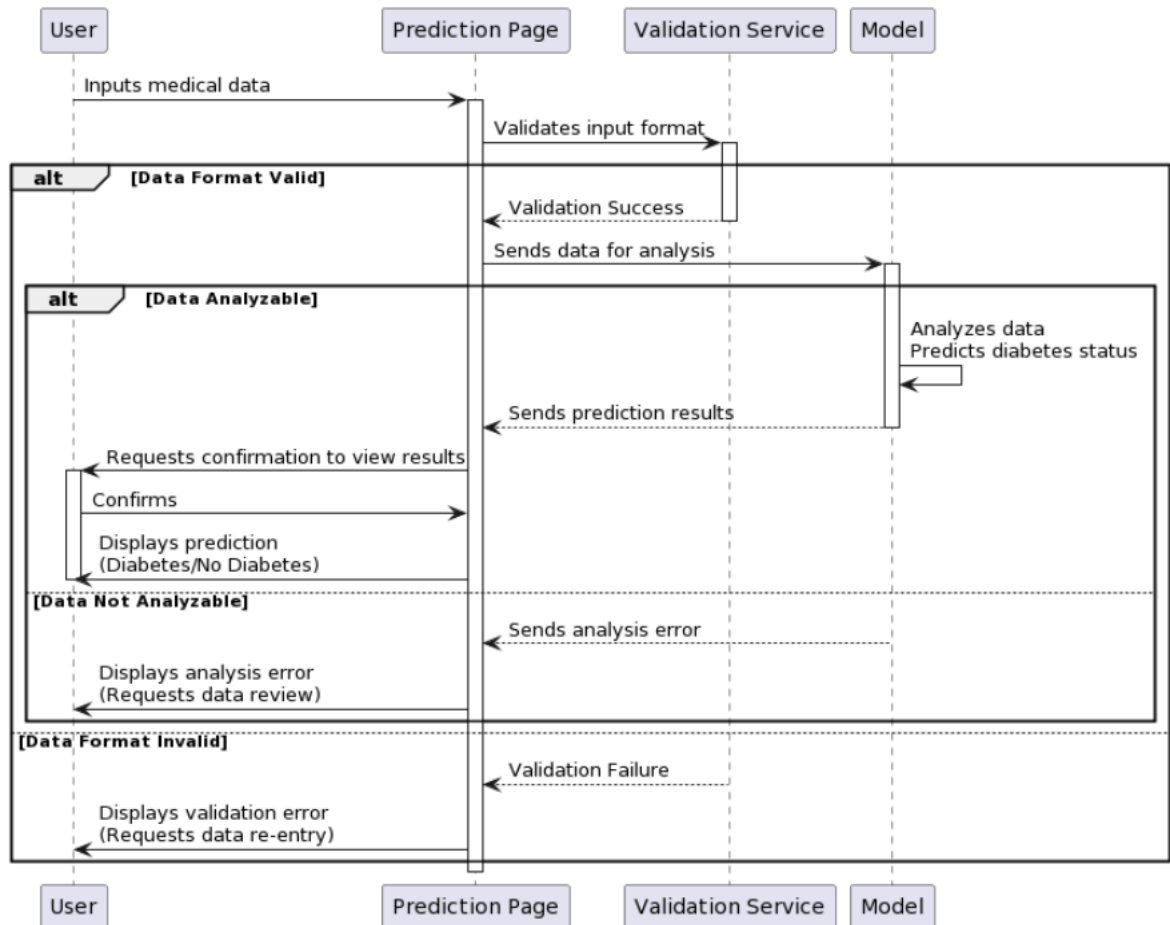


**Figure 4.2: Refinement of Class and Object Diagram**

In the above refined class and object diagram, the user interact with the Predictor class through the method like home(), view() and likely other methods that shown in diagram. These methods allow the user to navigate the application and initiate task. The predictor class calls the load\_csv() and analyze\_data() method, which likely performs some exploratory analysis on the data. The Predictor calls then calls the build\_model() method of the Model class, providing the training data and the algorithm to use for training. The Predictor class

calls the predict() method of the Model class, providing the data to use for prediction. The model class then uses the trained model to generate prediction on the data. The Predictor class calls the show\_result() method to display the prediction results to the User.

### 3. Refinement of the Sequence Diagram:

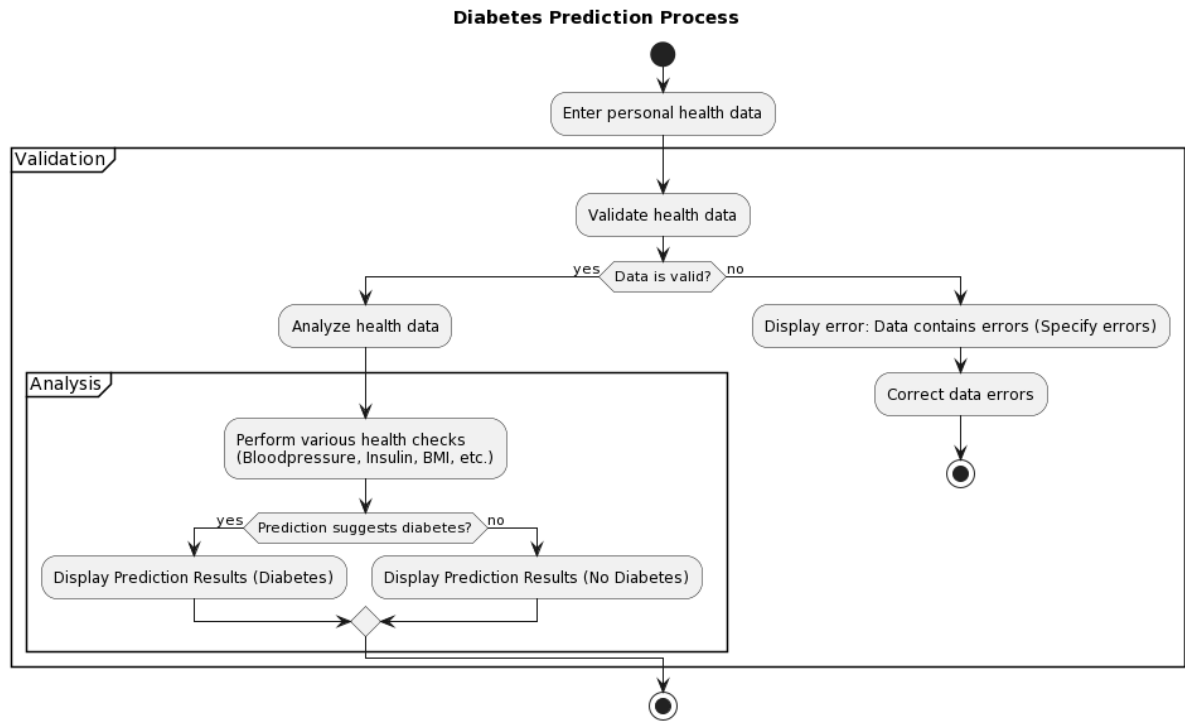


**Figure 4.3: Refinement of Sequence Diagram**

In above refined sequence diagram, the user initiates the process by sending the request to the Prediction Page. For Data gathering, the prediction pages prompt the Use to enter medical data relevant to diabetes prediction (e.g. age, insulin, blood pressure). The Prediction page pre-processes the medical data relevant the medical data to ensure it's compatible with the

Model's requirements (e.g., handling missing values, normalization). The Prediction page sends the processed medical data to the Model. The model performs any necessary preprocessing on the data. The model checks if a pre-trained model or diabetes prediction already exists. The model sends the prediction result back to the Prediction Page. After that, prediction page displays the prediction result (User suffer from diabetes or not) to the user.

#### 4. Refinement of Activity Diagram



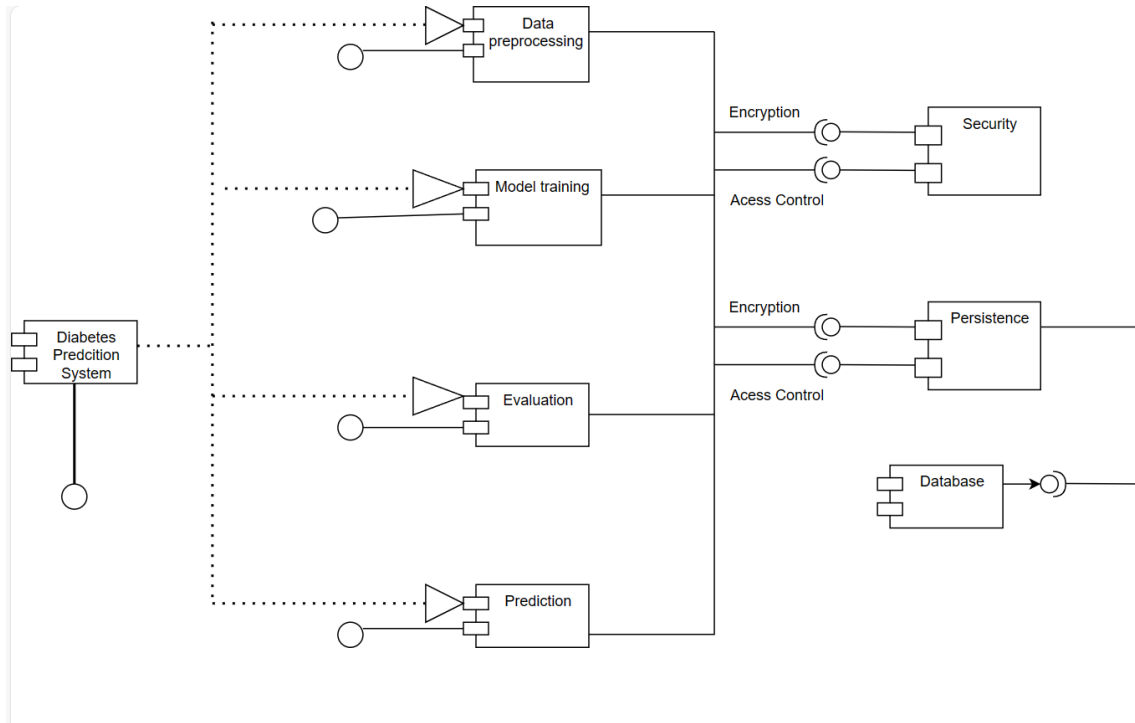
**Figure 4.4: Refinement of Activity Diagram**

In above refined activity diagram, it all starts when a user inputs their personal health information, such as blood pressure, insulin, levels, BMI and other relevant data. This information enters the validation phase where it's checked for completeness and accuracy. If everything looks good, then process moves forward to a detailed and analysis stage. In this stage, the system examines the provided health data more closely, performing series of health check to evaluate the diabetes risk.

Based on these checks, the system reaches a conclusion. If the indicator suggests a likelihood of the diabetes, it informs the user that the results indicate the diabetes. Otherwise, it

reassures the user by displaying results that suggest no diabetes. However, if there is an issue with the data initially provided – maybe some information is missing or incorrect – the system finds errors and asks the user to correct them. This feedback loop is crucial as it ensures the analysis is based on accurate and complete information. Once the errors are pointed out, the process halts, implying that the user needs to address these issues before proceeding.

### 4.1.3 Component Diagram

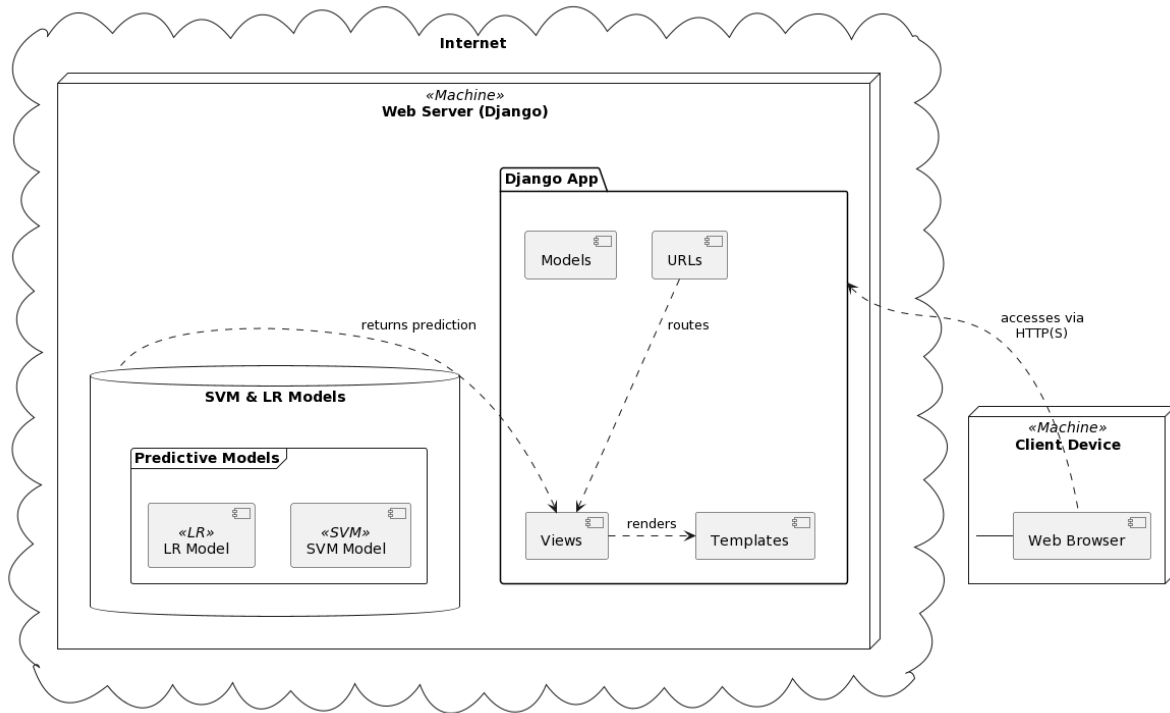


**Figure 4.5: Component Diagram**

The system’s component diagram highlights the key components and their security elements which highlighting the system’s overall structure. ‘Data Preprocessing’ for preparing and transforming raw data into suitable format and ‘Model training’ for training dataset to teach the model to make prediction or classification, ‘Evaluation’ for evaluating using a separate dataset to assess its performance, an ‘Prediction’ for predicting on new or unseen data. Access control and encryption are two of the specific security mechanism in place to secure data throughout the system. The “Persistence” component interfaces with a “Database,”

pointing to reliable data management and storage systems. It also includes encryption and access control.

#### 4.1.4 Deployment Diagram

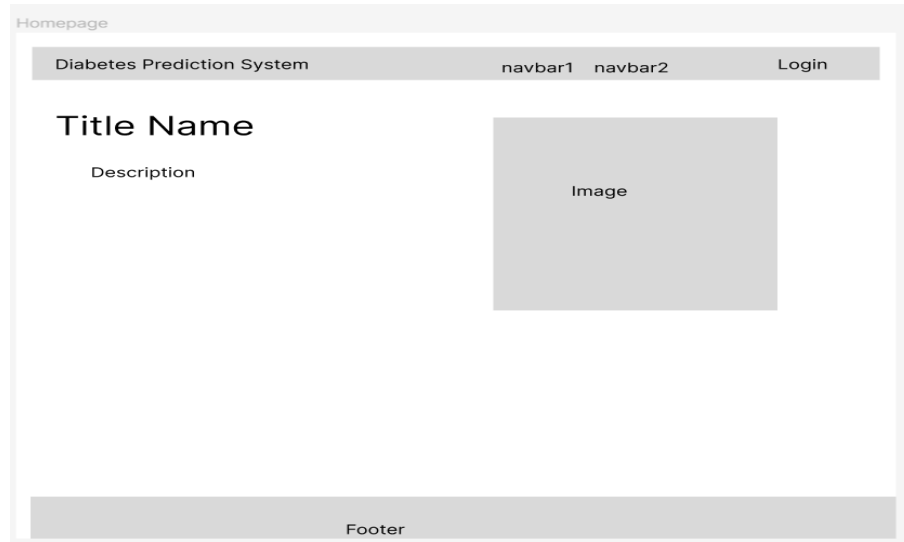


**Figure 4.6: Deployment Diagram**

In above deployment diagram, for predicting the diabetes, the user goes to a website that powered from Django framework, a kind of software that helps build websites. They fill out a form with some health details and click Predict button. This information goes over the internet to a server, a powerful computer that runs the website and the Django software. Django looks at the information and decides what to do with it. It uses two special models, called SVM and LR models, which are smart model that use the health details to predict if the user might get diabetes. Django takes prediction from these tools and puts it into the webpage that it displays it into the webpage that showing them the result that they are suffer from diabetes or not. So, from the user's action of submitting their details to getting their diabetes risk, it's all handled smoothly by Django and these smart models, making it easy for users to get insights about their health.

#### 4.1.5 Interface Design

We use wireframes in our project that shows basic visual representations of a web page or application interface. They describe the layout and features of your product without being weighed down with particular hues, graphics, or text. [7]. We have built the wireframe for our project using the design tool Figma. Below the some samples of design for Home page and Prediction page:



**Figure 4.7: Home Page**



**Figure 4.8: Prediction Page**

## 4.2 Algorithm Details

We focus on accurately identifying if data represents someone with diabetes or not. While using more samples doesn't always make the prediction better, we've noticed that sometimes the methods are fast but not very accurate. Our main goal is to be as accurate as possible. We found that using a large part of our data for training and a smaller part for testing can help improve accuracy. After reviewing different methods, we learned that Support Vector Machine, Logistic Regression, and Artificial Neural Networks are the best choices for predicting diabetes. In our system, we use Logistic Regression and Support vector Machine to predict diabetes.

### a. Support Vector Machine:

The Support Vector Machine (SVM) concept was initially introduced by Vapnik. It is a group of supervised learning methods commonly applied in medical diagnosis for both classification and regression tasks. The main strength of SVM lies in its ability to minimize classification errors while also maximizing the distance between the separating boundary and the closest samples. This is why SVMs are known as Maximum Margin Classifiers. The underlying theory of SVM is based on minimizing the risk of incorrect predictions, a principle known as structural risk minimization, which provides a solid theoretical foundation for its algorithm.

The maximizing of the margin between two distinct classes is the primary objective of SVM. This indicates that you should aim to have the same number of points in one class on one side of the decision boundary and the same number of points in the other class on the other. All points with a higher degree of separation will be correctly classified in this scenario, whereas all points with a lower degree of separation will be incorrectly classified. [8]

The equation for soft margin SVM algorithm is as follow:

$$\begin{aligned} \max_{||w||} \frac{2}{||w||} &= \max_{||w||} \frac{1}{||w||} \\ &= \min ||w|| = \min \frac{1}{2} ||w||^2 \end{aligned} \quad \dots\dots\dots(i)$$

Where w represents weight vector that defines the orientation of the hyperplane used to

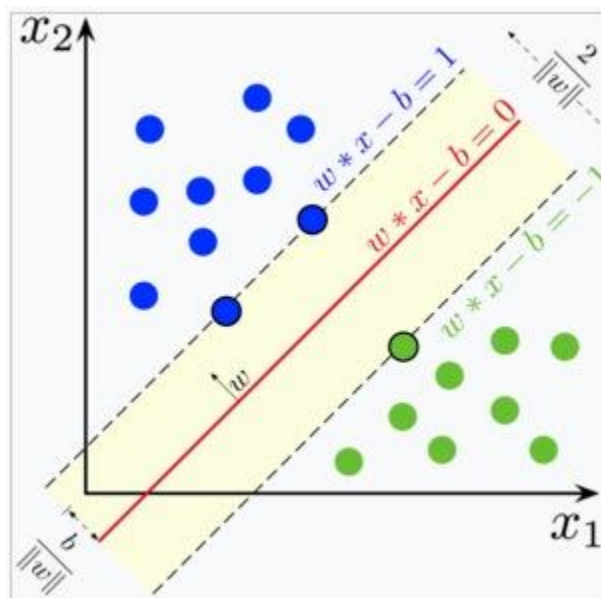
separate different classes in the feature space. The SVM algorithm aims to find the optimal hyperplane that separates the classes of data points with the maximum margin, and  $w$  is central of defining this hyperplane. The decision boundary (or hyperplane) in an SV is described mathematically as follows:

$$w \cdot x + b = 0 \dots\dots\dots(ii)$$

Where

$w$  is the weight vector,  $x$  represents the feature vectors of the data points and  $b$  is bias term, which offsets the hyperplane from the origin.

The direction of  $w$  is orthogonal (perpendicular) to the hyperplane, and the magnitude of  $w$  influences the margin between the classes. Specifically, the margin is inversely proportional to the norm of  $w$ , so minimizing the norm of  $w$  (while enforcing the correct classification of training samples) leads to the maximization of the margin, which is the primary goal in the optimization problem solved by SVMs.



**Figure 5.1: SVM Diagram**

#### b. Logistic Regression:

Logistic regression is a powerful statistical method used primarily for binary classification problems, enabling the prediction of an outcome based on one or more independent



variables. It operates by estimating the probabilities using a logistic function, which maps any real-valued number into a value between 0 and 1, representing the likelihood of the dependent variable falling into one of two categories. This function is characterized by its S-shaped curve, which ensures that predictions are confined to the range of 0 to 1, making it particularly suited for scenarios where the outcome is dichotomous, such as disease/no disease, pass/fail, or click/no click.

At the heart of logistic regression is the logistic or sigmoid function, defined as:

$$\sigma(z) = \frac{1}{1+e^{-z}} \dots\dots\dots\textbf{(iii)}$$

This S-shaped curve maps any real-valued number  $z$  to the  $(0, 1)$  interval, making it interpretable as a probability. The variable  $z$  represent the linear combination of input features ( $\mathbf{x}$ ) and their corresponding weights ( $\mathbf{w}$ ), plus a bias term ( $\mathbf{b}$ ), such that  $\mathbf{z} = \mathbf{w}^T \mathbf{x} + \mathbf{b}$ .

This probability that a instances  $\mathbf{x}$  belongs to the positive class ( $y=1$ ) can be modeled as:

$$\mathbf{P}(y = 1 | \mathbf{x}) = \sigma(\mathbf{w}^T \mathbf{x} + \mathbf{b}) \dots\dots\dots\textbf{(iv)}$$

And consequently, the probability of the negative class ( $y=0$ ) is:

$$\mathbf{P}(y=0 | \mathbf{x}) = 1 - \mathbf{P}(y = 1|\mathbf{x}).\dots\dots\dots\textbf{(v)}$$

To measure how well our model fits the data, we use a cost function. For logistic regression, this is typically the log loss (or binary cross-entropy), which for a single observation is:

$$\textbf{Cost} = -[y \log(\hat{y}) + (1 - y) \log(1 - \hat{y})].\dots\dots\dots\textbf{(vi)}$$

Where  $y$  is the actual label and  $\hat{y}$  is the predicted probability  $\mathbf{P}(y = 1|\mathbf{x})$ . The cost function for the entire training dataset is the average cost over all observations.

To find the best parameters ( $\mathbf{w}$  and  $\mathbf{b}$ ) for our model, we aim to minimize the cost function. This is usually done through optimization algorithms like Gradient Descent. The gradient of the cost function with respect to each parameter is computed to update the parameters in the direction that reduces the cost.

In each iteration, parameters are updated as follows:

$$\mathbf{w} := \mathbf{w} - \alpha \frac{\partial J}{\partial \mathbf{w}}$$

$$b := b - \alpha \frac{\partial J}{\partial b} \dots\dots\dots(\text{vi})$$

where,  $\alpha$  is the learning rate,  $J$  is the cost function and  $\partial J / \partial \mathbf{w}$  and  $\partial J / \partial b$  are the gradient of  $J$  with respect to  $\mathbf{w}$  and  $b$ , respectively.

## **Chapter 5: System Implementation and Testing**

### **4.2 Implementation**

Implementing System involved a series of steps to design, develop and launch the web application such as defining requirements and plan, selecting methodology, choosing a platform, designing, building the user interface (Frontend components), implementing backend functionality, testing and quality assurance, creation of the content and optimization and so on. During this phase, the actual project became perceptible to external stakeholders.

#### **4.2.1 Tools used**

- Draw.io: For creating diagrams like use case, class and object, component diagram etc.
- HTML and CSS: For creating UI of web application.
- Python: Backend programming language and used for implementing the machine learning model.
- SQL Lite: SQLite is database that is default provided in Django Framework.
- VS Code and Jupyter Notebook: For writing and running the code.
- Microsoft Excel: To prepare Gantt charts.
- Microsoft Word: To prepare document.

#### **4.2.2 Implementation of Module**

##### **1. Data Collection**

As part of our project, we collected the diabetes dataset from Kaggle. We found Pima Indian Diabetes database to serve as the foundation for my machine learning analysis. This dataset is originally from the National Institute of Diabetes and Digestive and Kidney Diseases. The objective of the dataset is to diagnostically predict whether or not a patient has diabetes, based on certain diagnostic measurements included in the dataset. Several constraints were placed on the selection of these instances from a larger database. In particular, all patients here are females at least 21 years old of Pima Indian heritage.

## 2. Data analysis

We have observed the column of our dataset by displaying the first five rows of our dataset:

#printing the first 5 rows of the dataset

```
diabetes_dataset.head()
```

Output:

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

Then we have observed the number of rows and column in this dataset:

```
diabetes_dataset.shape
```

Output:

(768, 9)

We have observed the statistical measures of the data:

```
diabetes_dataset.describe()
```

Output:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
count	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000
mean	3.845052	120.894531	69.105469	20.536458	79.799479	31.992578	0.471876	33.240885	0.348958
std	3.369578	31.972618	19.355807	15.952218	115.244002	7.884160	0.331329	11.760232	0.476951
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.078000	21.000000	0.000000
25%	1.000000	99.000000	62.000000	0.000000	0.000000	27.300000	0.243750	24.000000	0.000000
50%	3.000000	117.000000	72.000000	23.000000	30.500000	32.000000	0.372500	29.000000	0.000000
75%	6.000000	140.250000	80.000000	32.000000	127.250000	36.600000	0.626250	41.000000	1.000000
max	17.000000	199.000000	122.000000	99.000000	846.000000	67.100000	2.420000	81.000000	1.000000

We have observed the distribution of outcome

```
diabetes_dataset['Outcome'].value_counts()
```

Output:

```
0    500
```

```
1    268
```

Name: Outcome, dtype: int64

We have visualize correlation matrix:

```
corr_matrix = diabetes_dataset.corr()
```

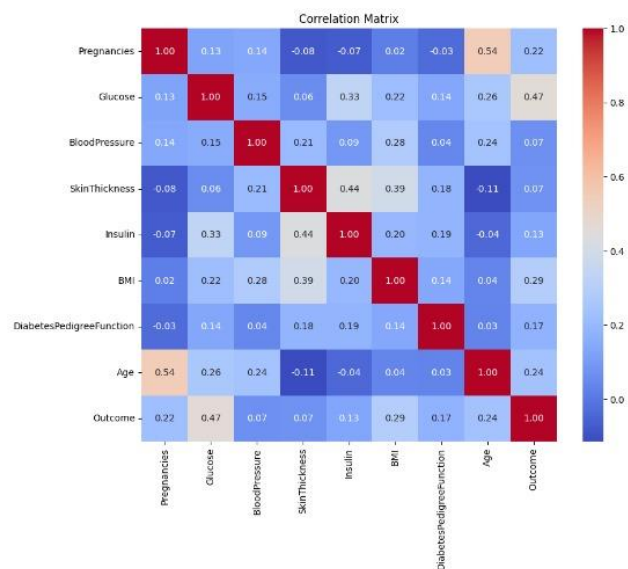
```
plt.figure(figsize=(10, 8))
```

```
sns.heatmap(corr_matrix, annot=True, cmap='coolwarm', fmt=".2f")
```

```
plt.title('Correlation Matrix')
```

```
plt.show()
```

Output:



**Figure 5.1: Correlation Matrix**

We have checked the missing values in our dataset:

```
print("\nStatistical summary of the dataset:")
```

```
print(diabetes_dataset.describe())
```

Output:

Missing values in the dataset:

Pregancies	0
------------	---

Glucose	0
---------	---

BloodPressure	0
---------------	---

Skin Thickness	0
----------------	---

Insullin	0
----------	---

BMI	0
-----	---

DiabetesPedigreeFunction	0
--------------------------	---

Age	0
-----	---

Outcomes	0
----------	---

dtype: Int

### 3. Data processing

We have separated the target variable and features from the dataset. Dataset contains 8 features such as pregnancies, glucose, blood pressure, skin thickness, insulin, BMI , diabetes pedigree function, age and last column contains outcome class i.e. 1 and 0. 1 means diabetes and 0 means non-diabetes.

```
#separating the data and labels
```

```
X=diabetes_dataset.drop(columns='Outcome',axis=1)
```

```
Y=diabetes_dataset['Outcome']
```

Features:

```
      Pregnancies  Glucose  BloodPressure  SkinThickness  Insulin   BMI   \
0                6    148             72             35         0   33.6
1                1     85             66             29         0   26.6
2                8    183             64              0         0   23.3
3                1     89             66             23        94   28.1
4                0    137             40             35       168   43.1
..            ...    ...             ...             ...     ...   ...
763             10    101             76             48       180   32.9
764              2    122             70             27         0   36.8
765              5    121             72             23       112   26.2
766              1    126             60              0         0   30.1
767              1     93             70             31         0   30.4
```

```
      DiabetesPedigreeFunction  Age
0                        0.627   50
1                        0.351   31
2                        0.672   32
3                        0.167   21
4                        2.288   33
..                        ...    ...
763                       0.171   63
764                       0.340   27
765                       0.245   30
766                       0.349   47
767                       0.315   23
```

```
[768 rows x 8 columns]
```

Target variable:

```
0      1
1      0
2      1
3      0
4      1
..
763    0
764    0
765    0
766    1
767    0
Name: Outcome, Length: 768, dtype: int64
```

Then we have performed data standardization. Data standardization is done so that all features have a similar scale, preventing algorithms from being biased towards features with larger magnitudes. Data standardization is done with the help of mean and standard deviation.

```
# Data standardization
```

```
mean = X.mean(axis=0)
```

```
std_dev = X.std(axis=0)
```

```
standardized_data = (X - mean) / std_dev
```

```
X = standardized_data
```

Before standardization of data:

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

After standardization of data:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age
0	0.639530	0.847771	0.149543	0.906679	-0.692439	0.203880	0.468187	1.425067
1	-0.844335	-1.122665	-0.160441	0.530556	-0.692439	-0.683976	-0.364823	-0.190548
2	1.233077	1.942458	-0.263769	-1.287373	-0.692439	-1.102537	0.604004	-0.105515
3	-0.844335	-0.997558	-0.160441	0.154433	0.123221	-0.493721	-0.920163	-1.040871
4	-1.141108	0.503727	-1.503707	0.906679	0.765337	1.408828	5.481337	-0.020483

After Standardization, we have splited the data for training and testing. 80% of data are training data and 20% of data are testing data.

```
X = standardized_data
```

```
Y = diabetes_dataset['Outcomes']
```

```
#train test split
```

```
X_train, X_test, Y_train, Y_test = train_test_split(X,Y, test_size = 0.2, stratify=Y,  
random_state=2)
```

```
print(X.shape, X_train.shape, X_test.shape)
```



#### 4. Model training

The SVM and Logistic Regression model is trained with the training data. We have separated earlier in data processing stage. 80% of the data is trained data and 20% data is for testing. We have used polynomial kernel for both model.

For training SVM model:

# SVM Classifier

class PolynomialSVM:

```
def __init__(self, degree=3, learning_rate=0.01, lambda_param=0.1, n_iters=1000):
```

```
    self.degree = degree
```

```
    self.lr = learning_rate
```

```
    self.lambda_param = lambda_param
```

```
    self.n_iters = n_iters
```

```
    self.w = None
```

```
    self.b = None
```

```
def _polynomial_kernel(self, x, y):
```

```
    return (1 + np.dot(x, y)) ** self.degree
```

```
def fit(self, X, y):
```

```
    n_samples, n_features = X.shape
```

```
    self.w = np.zeros(n_features)
```

```
    self.b = 0
```

```
    for _ in range(self.n_iters):
```

```
        for idx, x_i in enumerate(X):
```

```
            condition = y[idx] * (self._polynomial_kernel(x_i, x_i) - self.b) >= 1
```

```
            if condition:
```

```
                self.w -= self.lr * (2 * self.lambda_param * self.w)
```

```
            else:
```

```
                self.w -= self.lr * (2 * self.lambda_param * self.w - np.dot(x_i, y[idx]))
```

```
self.b -= self.lr * y[idx]
```

```
def predict(self, X):  
    approx = np.dot(X, self.w) - self.b  
    return np.sign(approx)
```

```
svm_classifier_scratch = PolynomialSVM(degree=10, learning_rate=0.001,  
lambda_param=0.01, n_iters=2000)  
svm_classifier_scratch.fit(X_train, Y_train)
```

For training logistic regression:

```
class LogisticRegression:  
    def _init_(self, learning_rate=0.001, n_iters=1000):  
        self.lr = learning_rate  
        self.n_iters = n_iters  
        self.weights = None  
        self.bias = None  
  
    def fit(self, X, y):  
        n_samples, n_features = X.shape  
        self.weights = np.zeros(n_features)  
        self.bias = 0  
  
        for _ in range(self.n_iters):  
            linear_model = np.dot(X, self.weights) + self.bias  
            y_pred = self._sigmoid(linear_model)  
            dw = (1 / n_samples) * np.dot(X.T, (y_pred - y))  
            db = (1 / n_samples) * np.sum(y_pred - y)  
            self.weights -= self.lr * dw  
            self.bias -= self.lr * db
```

```

def predict(self, X):
    linear_model = np.dot(X, self.weights) + self.bias
    y_pred = self._sigmoid(linear_model)
    y_pred_class = [1 if i > 0.5 else 0 for i in y_pred]
    return np.array(y_pred_class)

def _sigmoid(self, x):
    return 1 / (1 + np.exp(-x))

lr_classifier_scratch = LogisticRegression()
lr_classifier_scratch.fit(X_train, Y_train)

```

## 5. Making Predictive System

We give medical information as a input data in Prediction system. Prediction system sends that input data to Model. Then the model changes the input data to numpy array. Then It reshape the array as we are predicting for one instance. Then we standardize the input data. After the standardization, we print the prediction result i.e. whether the person is not diabetic otherwise the person is diabetes. The program for the making predictive system is given below:

```

input_data = (5,166,72,19,175,25.8,0.587,51)

# changing the input_data to numpy array
input_data_as_numpy_array = np.asarray(input_data)

# reshape the array as we are predicting for one instance
input_data_reshaped = input_data_as_numpy_array.reshape(1,-1)

# standardize the input data
std_data = scaler.transform(input_data_reshaped)
print(std_data)

```

```
prediction = classifier.predict(std_data)
print(prediction)
```

```
if (prediction[0] == 0):
    print('The person is not diabetic')
else:
    print('The person is diabetic')
```

Output:

The person is diabetes.

### 4.3 Testing

Software testing is a process of evaluating software to find defects and ensures that it meets the requirement specified by the customer. It is an important part of the software development process and helps to ensure the software meets the needs of the users.

#### 4.3.1 Test Cases for Unit Testing

In unit testing, the smallest testable part of an application, called units, is individually scrutinized for proper operation. Individual units of code, typically at the function or method level, were tested to ensure their correctness and proper behavior.

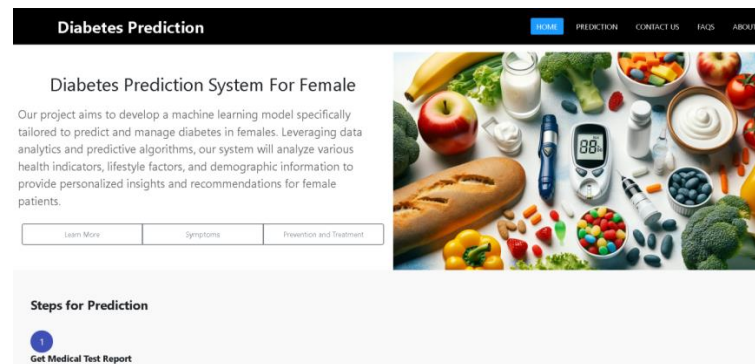
The following test scenarios were employed for conducting unit testing.

**Table 5. 1: Unit Testing for the System**

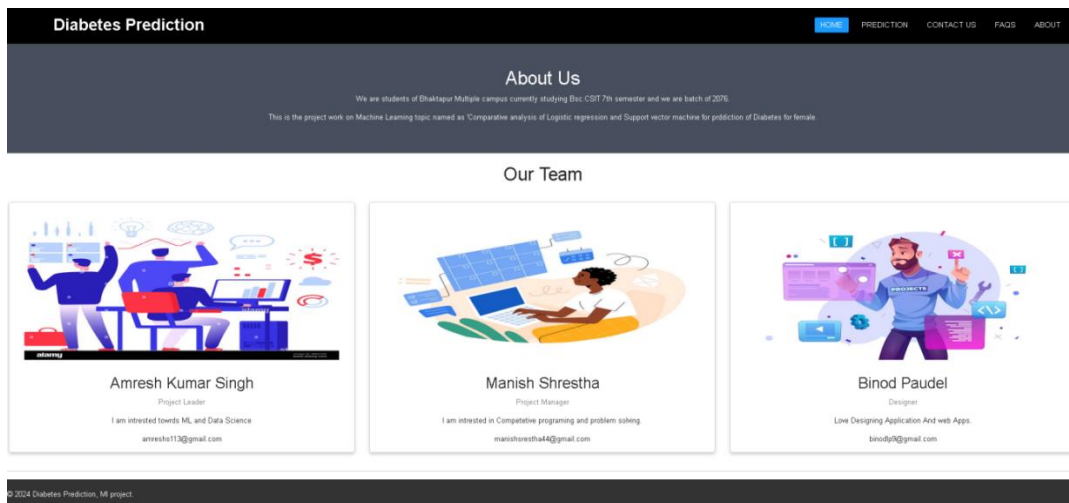
Test ID	Description	Action	Expected Result	Actual Result	Test Status
UT1	Clicking the Home navbar.	Click on the “Home” link	User is redirected to the homepage.	User is redirected to the homepage.	Pass

UT2	Clicking the About navbar.	Click on the “About” link	User is redirected to the About Page.	User is redirected to the About Page	Pass
UT3	Clicking the Prediction navbar.	Click on the “Prediction” link	User is redirected to the Prediction Page.	User is redirected to the Prediction Page	Pass
UT3	Clicking the Accuracy Button	Click the accuracy after showing the result	User is redirected to Accuracy Page	User is redirected to Accuracy Page.	Pass

The Table 5.1 above shows the test cases for unit testing. At first, when we clicked the Home page navbar, it redirected the Homepage. Similarly, When we clicked the About navbar and Prediction navbar and accuracy button then it redirect to About page and Prediction Page. The Figure 5.2, 5.3 and 5.4 show the home page, about page and prediction page as expected result.



**Figure 5.2: Redirecting to the Homepage**



**Figure 5.3: Redirected to the About us Page**

Diabetes Prediction

---

**Pregnancies:**

**Glucose:**

**Blood Pressure:**

**Skin Thickness:**

**Insulin:**

**BMI:**

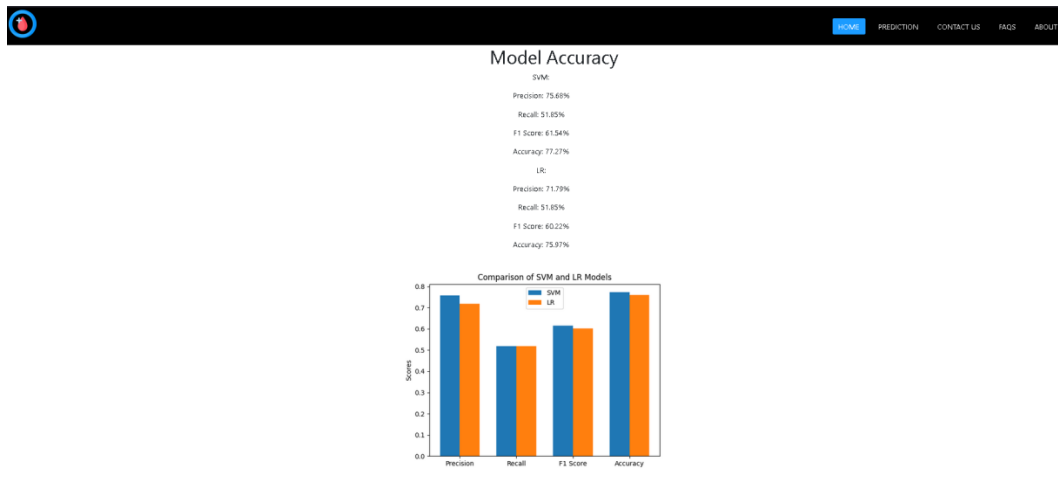
**Diabetes Pedigree Function:**

**Age:**

Select Model: SVM

Predict

**Figure 5.4: Redirected to the Prediction Page**



**Figure 5.5: Redirected to Accuracy Page**

### 4.3.2 Test cases for System Testing

System testing is like doing a final check on an entire web application to make sure everything works together perfectly. It's done to see if the web application does what it's supposed to do and if it's ready to be given to the users. This testing happens after checking the individual parts and before making sure the users are happy with it.

**Table 5. 2: System Testing for Diabetes Prediction**

Test ID	Test Scenario	Test Data	Expected Result	Actual Result	Test Status
ST1	User provides invalid inputs.	Pregnancies: 0 Blood Pressure: 0 Skin Thickness: 0 Insulin: 0 Diabetes Pedigree Function: 0 Age: 0	Invalid Input	Invalid Input	Pass

ST2	User provides valid inputs	Pregnancies: 6 Glucose: 148 Blood Pressure: 72 Skin Thickness: 35 Insulin: 0 BMI: 33.6 Diabetes Pedigree Function: 0.627 Age: 50	Output shows that the female patient suffers from diabetes.	Output shows that the female patient suffers	Pass
ST3	User provides valid inputs	Pregnancies: 1 Glucose: 85 Blood Pressure: 66 Skin Thickness: 29 Insulin: 0 BMI: 26.6 Diabetes Pedigree Function: 0.351 Age: 31	Output shows that the female patient does not suffer from diabetes.	Output shows that the female patient does not suffer from diabetes.	Pass

The above Table 5.2 shows the test cases for system testing. If the user provides the invalid input, then it pop up the invalid message box. If the user provide valid the input, it resulted that if the patient from the diabetes or not. Figure 5.5, 5.6 and 5.7 shows the result of above test case.



A screenshot of a web application for diabetes prediction. The form contains input fields for Skin Thickness, Insulin, BMI, Diabetes Pedigree, and Age, each with a value of 0. A modal dialog box is displayed in the center, showing a warning icon and the text "Invalid input: All values cannot be zero." with an "OK" button. Below the form is a "Select Model" dropdown menu set to "SVM" and a "Predict" button. The footer includes copyright information "© 2024 Diabetes Prediction, ML project." and navigation links: Home, Prediction, Contact Us, FAQs, About.

**Figure 5.6: Invalid Input**

A screenshot of the prediction result page. It features a green header with the text "Prediction Result". Below it, a green box contains the text "The predicted outcome is: From SVM: The person is diabeticFrom LR:The person is diabetic". At the bottom, a green bar contains the text "Show Model Accuracy".

**Figure 5.7: Showing diabetes result**

A screenshot of the prediction result page. It features a green header with the text "Prediction Result". Below it, a green box contains the text "The predicted outcome is: From SVM: The person is not diabeticFrom LR: The person is not diabetic". At the bottom, a green bar contains the text "Show Model Accuracy".

**Figure 5.8: Showing No diabetes result**

### 5.3 Result Analysis

The system was tested through unit testing and proved to be effective in executing its intended functions. The results showed that the project was able to meet its goals, but there is still some issue for improvements in term of expanding the system's capabilities.

For evaluating the performance matrix such as accuracy, recall, precision and F1-score we need to find the confusion matrix of each model.

The confusion matrix for SVM is given follow

		Actual values	
		Positive(1)	Negative(0)
Predicted Values	Positive(1)	91	9
	Negative(0)	26	28

**Figure 5.9: Confusion Matrix of SVM**

The confusion matrix for LR is given below

		Actual values	
		Positive(1)	Negative(0)
Predicted Values	Positive(1)	89	11
	Negative(0)	26	28

**Figure 5.10: Confusion Matrix of LR**

Performance evaluation:

- **Accuracy:** Measures the proportion of correctly classified instances out of the total number of instances in the dataset. It's a common metric for balanced datasets but may be misleading in the presence of class imbalance. The numerical formula of the accuracy is follows:

$$\text{Accuracy} = (\text{True Positive} + \text{True Negative}) / (\text{True Positive} + \text{False Positive} + \text{True Negative} + \text{False Negative})$$

For SVM:

$$\text{Accuracy} = (91+28) / (91+9+26+28) = 0.7727 = 77.27\%$$

For LR:

$$\text{Accuracy} = (89+28) / (89+11+26+28) = 0.7597 = 75.97\%$$

- **Precision:** Indicates the proportion of true positive predictions out of all positive predictions made by the model. It's useful when minimizing false positives is a priority, such as in medical diagnosis. The numerical formula of the precision is given below:

$$\text{Precision} = (\text{True Positive}) / (\text{True Positive} + \text{False Positive})$$

For SVM:

$$\text{Precision} = (91) / (91+28) = 0.7568 = 75.68\%$$

For LR:

$$\text{Precision} = (89) / (89+28) = 0.7179 = 71.79\%$$

- **Recall:** Measures the proportion of true positive predictions out of all actual positive instances in the dataset. It's important when capturing all positive instances is crucial, such as in fraud detection. The numerical formula of the recall is given below:

$$\text{Recall} = (\text{True Positive}) / (\text{True Positive} + \text{False Negative})$$

For SVM:

$$\text{Recall} = (91) / (91+26) = 0.5185 = 51.85\%$$

For LR:

Precision =  $(89) / (89 + 26) = 0.5185 = 51.85\%$

- **F1-score:** Harmonic mean of precision and recall, providing a balanced measure that considers both false positives and false negatives. It's particularly useful when dealing with imbalanced datasets. The numerical formula of F1-Score:

F1-Score =  $(2 * \text{Recall} * \text{Precision}) / (\text{Recall} + \text{Precision})$

For SVM:

F1-Score =  $(2 * 0.5185 * 0.7568) / (0.5185 + 0.7568) = 0.6154 = 61.54\%$

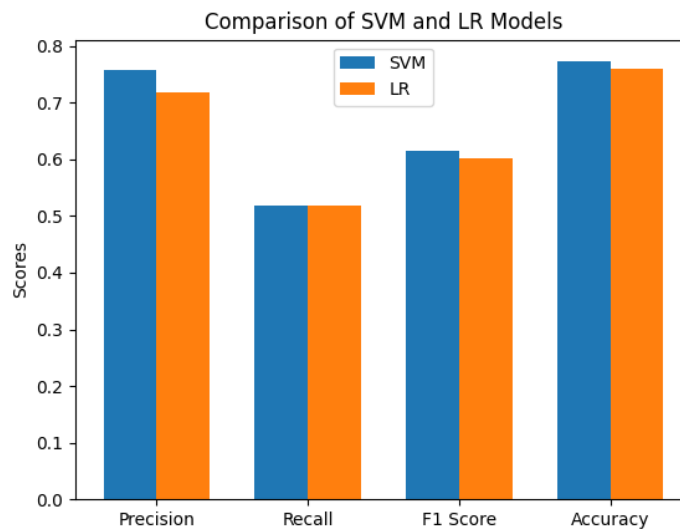
For LR:

F1-Score =  $(2 * 0.5185 * 0.7597) / (0.5185 + 0.7597) = 0.6022 = 60.22\%$

**Table 5. 3: Comparison Table between SVM and LR**

	Accuracy	Precision	Recall	F1-Score
SVM	77.27%	75.68%	51.85%	61.54%
LR	75.97%	71.79%	51.85%	60.22%

The Bar chart is as follow:



## **Chapter 6: Conclusion**

### **6.1 Conclusion**

In order to determine which computer algorithm is superior at early diabetes prediction in women, we conducted a comparison between Support Vector Machine (SVM) and Logistic Regression (LR) in our project. We found that while each has its own distinct advantages, SVM and LR are both effective. By integrating these algorithms, an improved diagnostic tool might be produced, highlighting the revolutionary potential of machine learning in diabetes diagnosis. In the end, this project demonstrates how customizing machine learning techniques can raise the precision of medical diagnoses and enhance the field of medical informatics. This system helps the female patient to detect their diabetes early so they can easily treat at first. Hence, It gives more accuracy and saves time for detecting diabetes. Hence this system only predicts the diabetes in female and not in male. This project is not only compare two machine learning algorithm even it assist the female that is unreachable to the doctor or hospital for checking diabetes or not.

### **6.2 Future Recommendation**

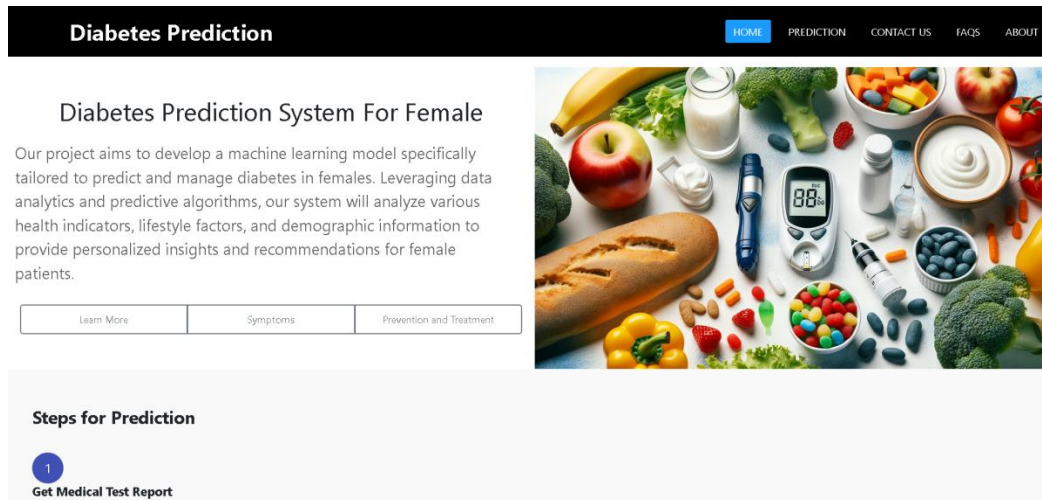
Our system only predicts the diabetes in women. So, In Future, we can also make a system that predicts the diabetes in both gender (i.e. male and female). Hence there is a certain limitation in the system. We can add more features in User Interface so the web application is more interactive to the system. In our system, we only predict the diabetes so in near future, we can add multiple disease such as heart disease, liver disease, etc..We can add more advance machine learning model to improve the accuracy of the disease prediction. This could involve deep learning algorithm that can analyze complex datasets, including medical images, to identify early sign of disease beyond what is explicitly reported by users. Hence, this system is only limited in web application. In near future, we can develop this system in the other platform such as mobile application. Hence, we can added more security to our system by using block-chain technology to enhance the security and privacy of the user data. This can assure users that their health information is stored in securely and shared only with their consent and making trust in digital health ecosystem.

## Reference

- [1] H. A. Samer Sawalha, "Agile Software Development: Methodologies and Trends," *International Journal of Interactive Mobile Technologies (IJM)*, p. 26, July 2020.
- [2] A. Y. S. S. P. R. N. Ankush Singh, "Multiple Disease Prediction System," *International Research Journal of Engineering and Technology (IRJET)*, Mar 2022.
- [3] P. P. M. C. Tejas N. Joshi, "Logistic Regression and SVM based diabetes prediction system," *International Journal For Technological Research In Engineering*, p. 5, 11 July 2018.
- [4] Computer Science and Engineering Department, University of Engineering and Technology Lahore, Pakistan, "A model for early prediction of diabetes," *ELSEVIER*, p. 6, 26 January 2019.
- [5] Riphah School of Computing and Innovation, Faculty of Computing, Riphah International University, Lahore 5400, Pakistan, "Prediction of Diabetes Empowered With Fused Machine Learning," *IEEE*, p. 10, December 9, 2021.
- [6] L. S. V. S. A. K. & H. D. Suyash Srivastava, "Prediction of Diabetes Using Artificial Neural Network," *LNEE*, vol. 478, 31 October 2018.
- [7] J. hannah, "What is A Wireframe? A Comprehensive Guide," CF BLOG, 25 Dec 2023. [Online]. Available: <https://careerfoundry.com/en/blog/ux-design/what-is-a-wireframe-guide/#:~:text=In%20simple%20terms%2C%20wireframes%20are,%2C%20visuals%2C%20or%20specific%20content..> [Accessed 30 Jan 2024].
- [8] Sidharth, "Implementing SVM from Scratch Python," 4 December 2022. [Online]. Available: <https://www.pycodemates.com/2022/10/implementing-SVM-from-scratch-in-python.html>. [Accessed 30 Jan 2024].
- [9] S. M. D. K. M. K. J. Mohammed Juned Shaikh, "Multiple Disease Prediction Webapp," *JETIR*, 2022 October 2022.
- [10] Sorchhaya Education Privated Limited, "Linear regression," 21 December 2023. [Online]. Available: <https://www.geeksforgeeks.org/ml-linear-regression/>. [Accessed 4 January 2024].

# Appendix

Screenshot:



## Homepage of our System

Diabetes Prediction

Pregnancies:

Glucose:

Blood Pressure:

Skin Thickness:

Insulin:

BMI:

Diabetes Pedigree Function:

Age:

Select Model:

## Prediction Page

### Prediction Result

The predicted outcome is: From SVM: The person is not diabetic Model Accuracy: 78.66%From LR: The person is not diabetic Model Accuracy: 78.50%



### Prediction Result Page

#### Frequently Asked Questions

- How accurate are the disease predictions?  
Our predictions are based on advanced machine learning algorithms trained on large datasets. While we strive for accuracy, please consult a medical professional for diagnosis and treatment.
- Can I rely solely on the website for medical advice?  
No, our website is not a substitute for professional medical advice. Always consult with a healthcare provider for personalized diagnosis and treatment.
- What diseases does the website predict?  
Currently, our website predicts heart disease, Parkinson's disease, and diabetes.

### FAQ Page



Diabetes Prediction


HOME PREDICTION CONTACT US FAQs ABOUT

### About Us

We are students of Bhaktapur Multiple campus currently studying Bsc.CSIT 7th semester and we are batch of 2076.

This is the project work on Machine Learning topic named as 'Comparative analysis of Logistic regression and Support vector machine for prediction of Diabetes for female.

### Our Team




**Amresh Kumar Singh**

Project Leader

I am intrested towards ML and Data Science

amresh113@gmail.com




**Manish Shrestha**

Project Manager

I am intrested in Competetive programing and problem solving.

manishshrestha44@gmail.com



**Binod Paudel**

Designer

Love Designing Application And web Apps.

binodp9@gmail.com

© 2024 Diabetes Prediction, MI project.

## About Us Page

Diabetes Prediction

HOME PREDICTION CONTACT US FAQs ABOUT

### Contact Us

Your Name:

Your Email:

Your Message:

Send Message

Thank you, binod paudell your message has been sent.

© 2024 Diabetes Prediction, MI project.

Home Prediction Contact Us FAQs About

## Contact Us Page

## Log of visits to Supervisor

**Table 1:Supervisor meet log**

Date	Remarks
6 <sup>th</sup> Jan , 2024	Initial Proposal submitted for review and supervision
20 <sup>th</sup> Jan , 2024	Discuss the project progress and supervision (Physical meetup)
12 <sup>th</sup> Mar, 2024	Shown the User Interface page and Model (Physical meetup)
18 <sup>th</sup> Mar, 2024	Submitted final defense report for review and supervision.
24 <sup>th</sup> Mar, 2024	Resubmitted final defense report for review and supervision.
2 <sup>nd</sup> April, 2024	Resubmitted final defense report and Presentation for review and supervision.