A Mini Project Report On

DIABETES PREDICTION USING MACHINE LEARNING

Submitted to partial fulfilment of the requirements for the award of the degree of

BACHELOR OF TECHNOLOGY

in

INFORMATION TECHNOLOGY

By

B. PRAVALIKA	**20911A1213**
P. RAJASREE	**20911A1241**
T. BHAVANA	**20911A1251**

Under the Guidance of

Mrs. G. INDIRA PRIYADARSHINI
Associate professor

Department of Information Technology

VIDYA JYOTHI INSTITUTE OF TECHNOLOGY

(An Autonomous Institution)

(Approved by AICTE, New Delhi & Affiliated to JNTUH, Hyderabad)

Aziz Nagar Gate, C.B. Post, Hyderabad-500075 2023-2024

VIDYA JYOTHI INSTITUTE OF TECHNOLOGY

(An Autonomous Institution)

(Approved by AICTE, New Delhi & Affiliated to JNTUH, Hyderabad)

Aziz Nagar Gate, C.B. Post, Hyderabad-500075

DEPARTMENT OF INFORMATION TECHNOLOGY

CERTIFICATE

This is to certify that the project report titled **"DIABETES PREDICTION USING MACHINE LEARNING"** is being submitted by **B. PRAVALIKA (20911A1213), P. RAJASREE (20911A1241), T. BHAVANA (20911A1251)** in partial fulfilment for the award of the Degree of Bachelor of Technology in **INFORMATION TECHNOLOGY,** is a record of bonafide work carried out by us under my guidance and supervision.

Project Guide	**Head of the Department**
Mrs. G INDIRA PRIYADARSHINI	Dr. B. Srinivasulu (M.E - PHD)
Associate Professor	Professor

External Examiner

DECLARATION

We, **B. PRAVALIKA (20911A1213), P. RAJASREE (20911A1241), T. BHAVANA (20911A1251)** hereby declare that the project entitled, **"DIABETES PREDICTION USING MACHINE LEARNING"** submitted for the degree of Bachelor of Technology in Information Technology is original and has been done by us and this work is not copied and submitted anywhere for the award of any degree.

Team Members:

B. PRAVALIKA	20911A1213
P. RAJASREE	20911A1241
T. BHAVANA	20911A1251

ACKNOWLEDGEMENT

We express our sincere gratitude to the project guide **Mrs. G. INDIRA PRIYADARSHINI**, Associate Professor, Vidya Jyothi Institute of Technology, Hyderabad for her timely cooperation and valuable suggestions while carrying out this work. It is her kindness that made us learn more from her. We are grateful to **Dr. B. Srinivasulu**, Professor and HOD, Department of IT, for his help and support during our academic year. We whole-heartedly convey our gratitude to Principal **Dr. E. Saibaba Reddy** for his constructive encouragement.

We would thank faculty members and my parents who have contributed to our progress through the course to come to this stage.

Team Members:

B. PRAVALIKA	**20911A1213**
P. RAJASREE	**20911A1241**
T. BHAVANA	**20911A1251**

ABSTRACT

Diabetes is a disease caused due to the increase level of blood glucose. Various traditional methods based on physical and chemical tests are available for diagnosing diabetes. Our task is to make predictions on medical data. This system proposes a diabetes prediction model utilizing machine learning techniques with data from the Pima dataset. The machine learning model is trained using the Support Vector Machine (SVM) algorithm. Input features such as BMI, glucose levels, number of pregnancies, and blood pressure are utilized to build an accurate prediction model. The trained model is then evaluated using input data to assess whether an individual is likely to have diabetes or not. This research aims to enhance early diagnosis and management of diabetes through the implementation of advanced machine learning techniques

INDEX

CHAPTER -1

INTRODUCTION

According to International Diabetes Federation 382 million people are living with diabetes across the whole world. The data which is collected is trained using various algorithms. Healthcare professionals working in this area have their own limits and cannot forecast the probability of high accuracy in diabetes. The project aims to improve diabetes prediction using SVM algorithm of machine learning considering the healthcare dataset from the open source of PIMA dataset sets which classifies the patients whether they are having diabetes or not.

Problem Statement:

In this present world we could see importance of certificate. Here we can also see that many of them are duplicating the certificates which is creating a problem in the real world, here we are allocating a series number to the certificate so that with the help of series number we can remove the duplication of certificate.

Objective:

The main objective to analyze how machine learning algorithms are used to identify the diabetes mellitus at an early stage, which is one of the most serious metabolic disorders in the world today.

CHAPTER-2

LITERATURE SURVEY

Survey -1

A new data preparation method based on clustering algorithms for diagnosis systems of diabetes diseases systems of diabetes diseases by **Yılmaz, Nihat & Inan, Onur & Uzer, Mustafa. (2014).**

A New Data Preparation Method Based on Clustering Algorithms for Diagnosis Systems of Diabetes Diseases. Journal of medical systems. 38. 48. 10.1007/s10916-014-0048-7.

The most important factors that prevent pattern recognition from functioning rapidly and effectively are the noisy and inconsistent data in databases. This presents a new data preparation method based on clustering algorithms for diagnosis of diabetes diseases. In this method, Support Vector Machines is used for classification. This newly developed approach was tested in the diagnosis of diabetes. The data sets used in the diagnosis of these diseases is Pima Indians Diabetes data sets obtained from the UCI database. The proposed system achieved 77.87 %, 78.18 %, 79.71 % classification success rates from these data sets. According to the results, the proposed method of performance is highly successful compared to other results attained, and seems very promising for pattern recognition applications.

Survey -2

Classification Of Diabetes Disease Using Support Vector Machine

By **Jegan, Chitra. (2013)** in International Journal of Engineering Research and Applications. 3. 1797 - 1801.

Diabetes is one of the most serious health challenges in both developing and developed countries. According to the International Diabetes Federation, there are 285 million diabetic people worldwide. This total is expected to rise to 380 million within 20 years. Due to its importance, a design of classifier for the detection of Diabetes disease with optimal cost and better performance is the need of the age. The Pima Indian diabetic database at the UCI machine learning laboratory has become a standard for testing data Machine Learning algorithms to see their prediction accuracy in diabetes data classification. The proposed method uses Support Vector Machine (SVM), a machine learning method as the classifier for diagnosis of diabetes. The machine learning method focus on classifying diabetes disease from high dimensional medical dataset. The experimental results obtained show that support vector machine can be successfully used for diagnosing diabetes disease

CHAPTER-3

FEASIBILITY STUDY

Feasibility studies are common across industries and disciplines. They are an important project planning tool that can help you identify points of failure in a project before any money or time gets invested. Feasibility studies are particularly useful for machine learning projects because ML projects are generally experimental in nature. They can fail for many reasons, some of which can be identified upfront with a feasibility study. In order to evaluate if the project can be done in the given time frame, we are using the TEL-evaluation methods, where we cover the feasibility of the project from a technological, economical and legal perspective. Those perspectives would help us have a broad vision on the requirements and implications related to the project. We also discuss in this section the methodology used in conducting the project.

3.1 Technological Feasibility

This project would be developed using technologies and libraries pertinent to Python. Python facilitates developers to increase the confidence and productivity about their developing software from development to deployment and maintenance. The benefits of making Python the perfect solution for machine learning and AI-driven projects include simplicity and consistency, flexibility, access to powerful AI and machine learning (ML) libraries and frameworks, platform independence, and large communities. These things increase the popularity of the language.

3.2 Economical Feasibility

This project will be based on Free and Open-Source Technologies and Libraries that are readily available to developers and scientists, free of cost. This means that we don’t have to worry about costs related to licensing or reusing source code. This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into research is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products has to be purchased.

3.3 Social Feasibility

The aspect of the study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make them familiar with it. Their level of confidence must be raised so that they are also able to make some constructive criticism, which is welcomed, as they are the final user of the system.

3.4 Operational Feasibility

Operational feasibility is the measure of how well a proposed system solves the problems, and takes advantage of the opportunities identified during scope definition and how it satisfies the requirements identified in the requirements analysis phase of system development. The operational feasibility assessment focuses on the degree to which the proposed development project fits in with the existing business environment and objectives with regard to development schedule, delivery date, corporate culture and existing business processes.

To ensure success, desired operational outcomes must be imparted during design and development. These include such design-dependent parameters as reliability, maintainability, supportability, usability, producibility, disposability, sustainability, affordability and others. These parameters are required to be considered at the early stages of design if desired operational behaviours are to be realised.

CHAPTER-4

SYSTEM REQUIREMENTS

4.1 Existing System

Many Existing research handled for diabetes detection. Data mining approach like

clustering, classification were studied in existing system .Diabetes prediction using algorithms such as KNN, k-means was proposed. A basic diabetic dataset is chosen for carrying out the comparative analysis The importance of feature analysis for predicting diabetes by employing machine learning technique is discussed. In this system the accuracy of detection is less.

Existing System Drawbacks:

- Implemented only conventional Machine Learning Algorithms and couldn't promise high accuracy and performance.
- This area of research still faces limitations related to lack of data preprocessing steps and issues related to validation.
- The performance of ML models does not necessarily show any improvement

4.2 Proposed System

The proposed system study is classification of Indian PIMA dataset for diabetes as binary classification problem. This is proposed to achieve through machine learning algorithm. For machine learning, SVM algorithm is proposed .The proposed system improves accuracy of prediction through Machine learning techniques.

Advantages:

- Individual user predictions are implemented.
- Accuracy is improved using machine learning

4.2.1 Modules

Data Collection:

Gather a dataset containing relevant features (e.g., age, BMI, family history, glucose levels) and the target variable (diabetes status - yes/no).

Data preprocessing

numpy: Useful for numerical operations and working with arrays, which are often used for storing and processing model predictions and evaluation metrics.

scikit-learn (sklearn): Scikit-learn provides a wide range of preprocessing tools, including:

StandardScaler: For standardizing numeric features to have mean 0 and variance 1.

train_test_split: For splitting your dataset into training and testing sets.

fit: To train your machine learning models on the training data.

predict: To make predictions on new data using the trained models.

score: To evaluate the model's performance, often using metrics like accuracy, precision, recall, F1-score, and ROC-AUC.

4.3 Software Requirements

• Operating System	: Windows 7 or higher
• Programming	: Python 3.6 and related libraries

4.4 Hardware Requirements

• RAM	:	4GB and Higher
• Processor	:	Intel i5 or above
• Hard Disk	:	2GB or above

4.5 Requirements Definition

After the severe continuous analysis of the problems that rise in the existing system, we are now familiar with the requirement that is required by the current system. The requirements that the system needs is categorized into the functional and non-functional requirements. These requirements are listed below:

4.5.1 Functional Requirements

Functional requirement defines which functions or features that are to be incorporated in any system to fulfill the business requirements and to be acknowledged by the clients. On the premise, the functional requirements specify relationship between the inputs and outputs. All the operations to be performed on the input data to obtain output are to be specified. This includes specifying the validity checks on the input and output data, parameters affected by the operations and the other operations, which must be used to transform the inputs into outputs. Functional requirements specify the behavior of the system for valid input and outputs.

4.5.2 Non Functional Requirements

Non-functional requirements provide a description of features, characteristics and capacity of the system and furthermore it may constraints the boundaries of the proposed system. The following are the non-functional requirements that are essential depending on the performance, cost, control and gives security efficiency and services.

Based on the above explained non-functional pre-requisites are as follows:

- User-friendly
- System should provide better accuracy
- To perform efficiently with better throughput and response time

CHAPTER-5

SYSTEM DESIGN

5.1 UML Diagrams

UML diagram is designed to let developers and customers view a system from a different perspective and in varying degrees of abstraction. UML diagrams commonly are created in visual modeling tools. In its simplest form, a use case can be described as a specific way of using the system from a User's (actor's) perspective. A more detailed description might characterize a use case as:

- a pattern of behaviour the system exhibits
- a sequence of related transactions performed by an actor and the system
- delivering something of value to the actor Use cases provide a means to:
- capture system requirements
- communicate with the end users and domain experts
- Test the system

Use cases are best discovered by examining the actors and defining what the actor will be able to do with the system. Since all the needs of a system typically cannot be covered in one use case, it is usual to have a collection of use cases. Together this use case collection specifies all the ways of using the system.

A UML system is represented using five different views that describe the system from distinctly different perspective. Each view is defined by a set of diagrams, which is as follows.

User Model View

- This view represents the system from the user's perspective.
- The analysis representation describes a usage scenario from the end-user's perspective.

Structural model view

- In this model the data and functionality are arrived from inside the system.
- This model view models the static structures.

Behavioural Model View

- It represents the dynamic of behavioural as parts of the system, depicting the interactions of collection between various structural elements described in the user model and structural model view.

Implementation Model View

- In this the structural and behavioural as parts of the system are represented as they are to be built.

Environmental Model View

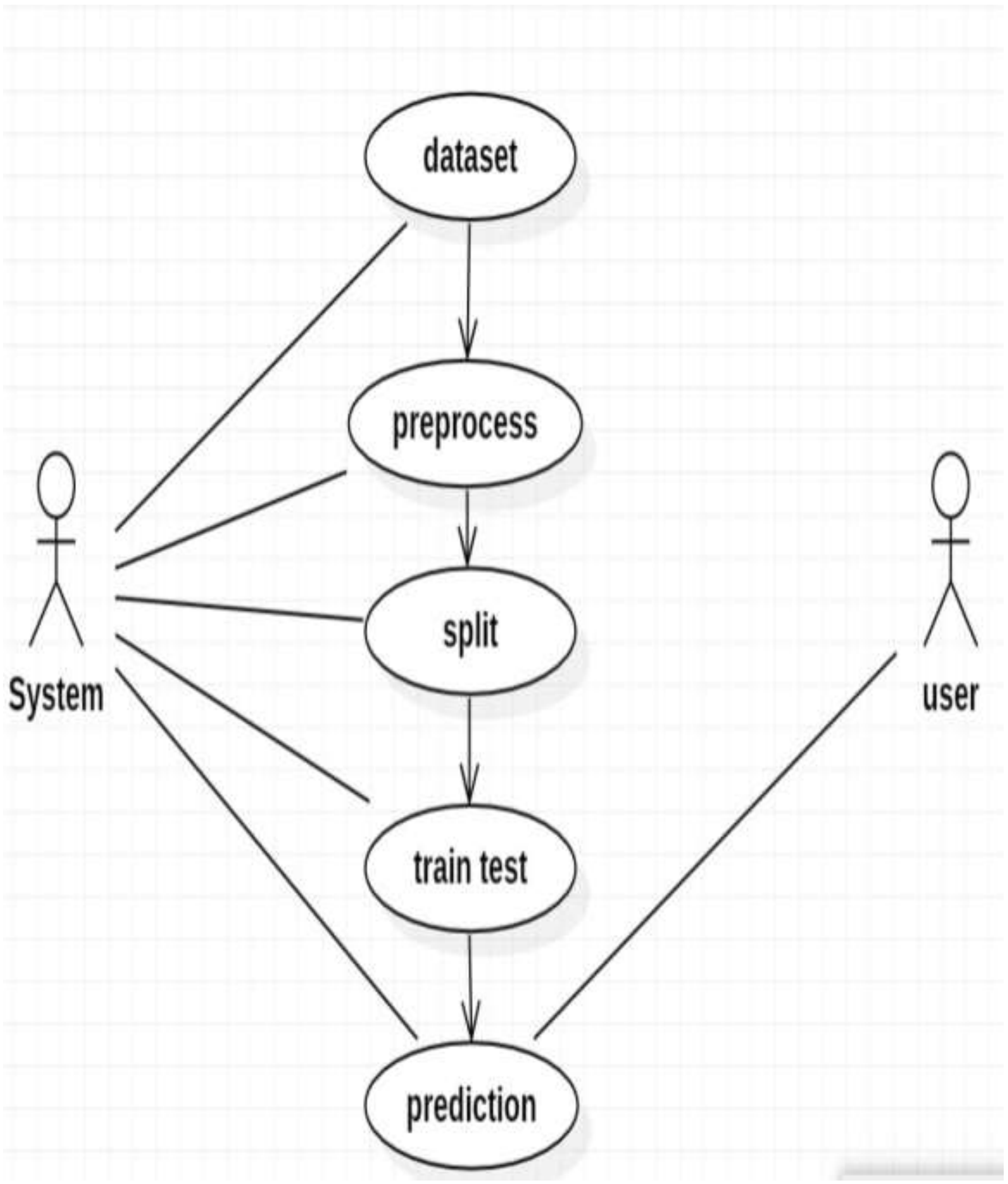
- In this the structural and behavioural aspect of the environment in which the system is to be implemented are represented.

UML is specifically constructed through two different domains they are:

- UML Analysis modeling, this focuses on the user model and structural model views of the system.
- UML design modeling, which focuses on the behavioural

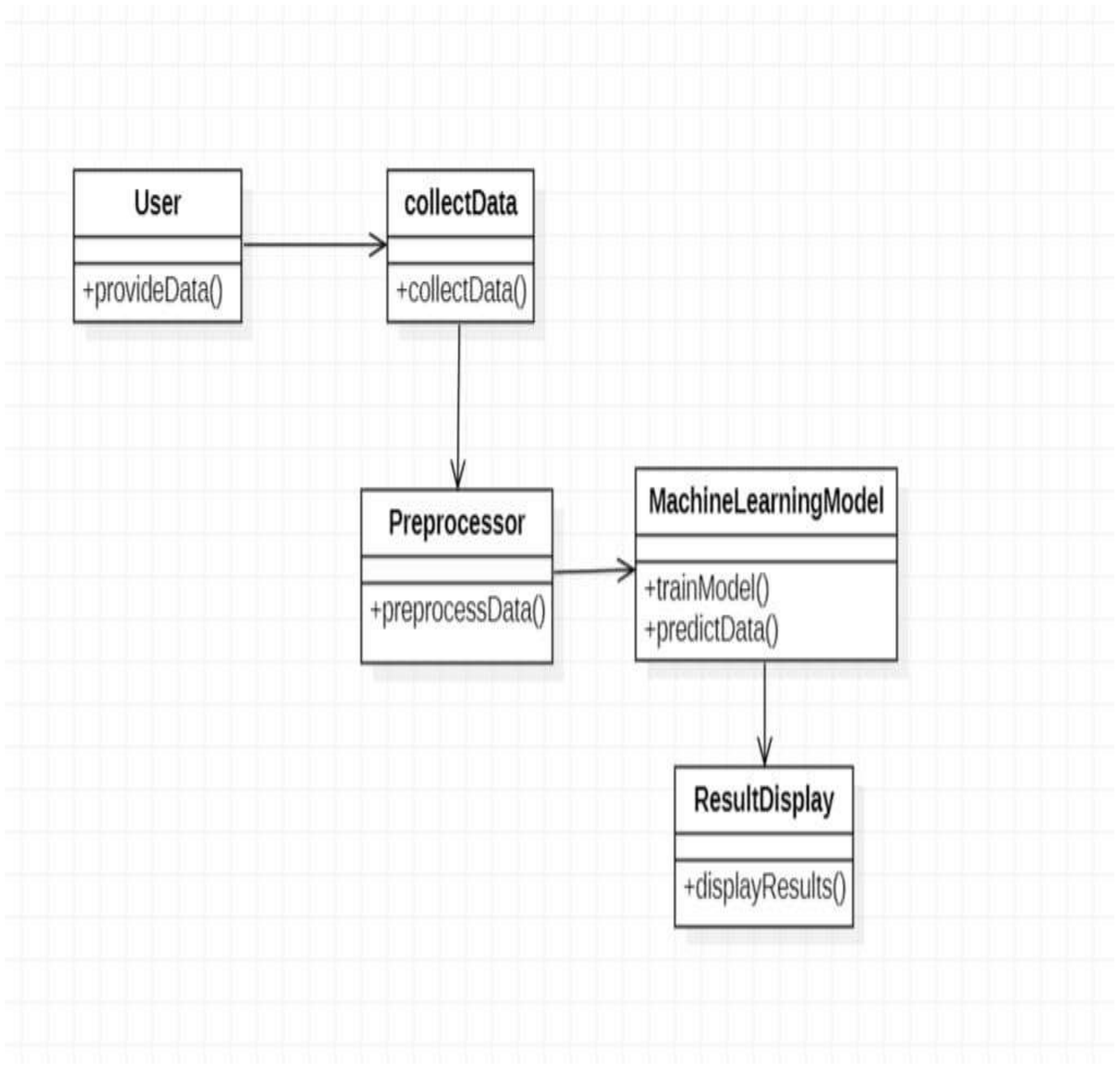
5.1.1 Usecase Diagram

A use case diagram is used to represent the dynamic behavior of a system. It encapsulates the system's functionality by incorporating use cases, actors, and their relationships. It models the tasks, services, and functions required by a system/subsystem of an application. It depicts the high-level functionality of a system and also tells how the user handles a system.



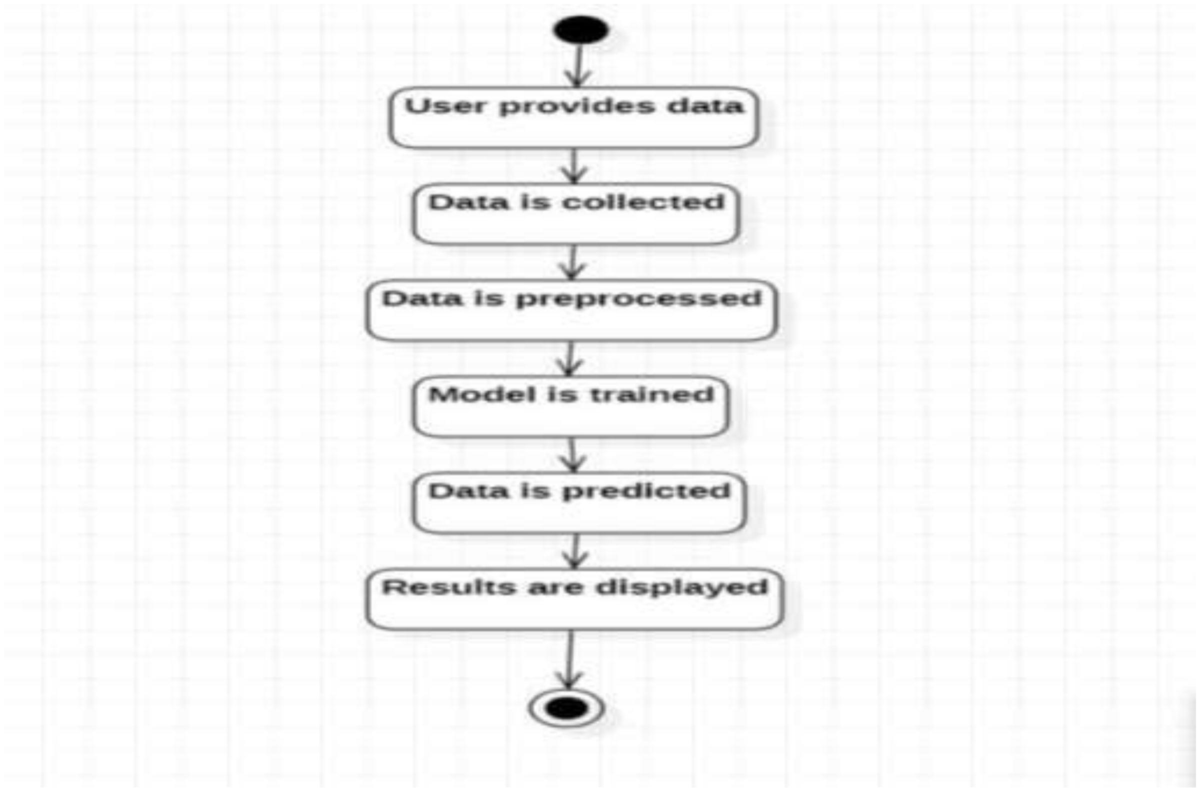
5.1.2 Class Diagram

In software engineering, a class diagram in the unified modelling language is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations, and the relationships among the classes it explains which class explain which information .



5.1.2 Activity Diagram

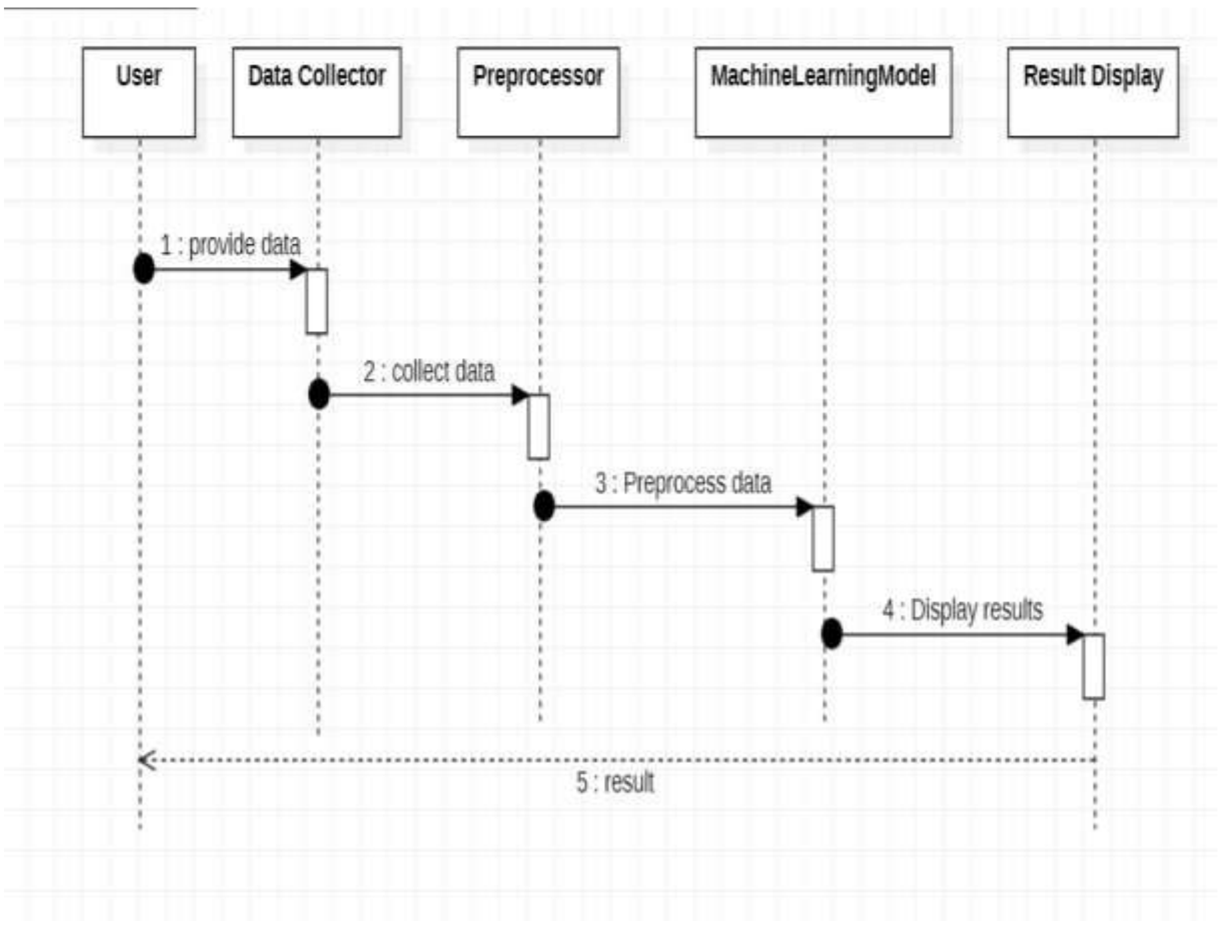
The basic purposes of activity diagrams is similar to other four diagrams. It captures the dynamic behaviour of the system. Activity is a particular operation of the system. Activity diagrams are not only used for visualizing the dynamic nature of a system, but they are also used to construct the executable system by using forward and reverse engineering techniques.



5.1.4 Sequence Diagram

A sequence diagram is a graphical view of a scenario that shows object interaction in a time-based sequence what happens first, what happens next. Sequence diagrams establish the roles of objects and help provide essential information to determine class responsibilities and interfaces.

There are two main differences between sequence and collaboration diagrams: sequence diagrams show time-based object interaction while collaboration diagrams show how objects associate with each other. A sequence diagram has two dimensions: typically, vertical placement represents time and horizontal placement represents different objects.



CHAPTER-6

SOFTWARE ENVIRONMENT

6.1 Machine Learning

In the last few years, the Machine Learning (ML) has attracted attention from both academic and industrial worlds. Machine learning is a branch of artificial intelligence (AI) and computer science which focuses on the use of data and algorithms to imitate the way that humans learn, gradually improving its accuracy. Machine learning is an important component of the growing field of data science. Through the use of statistical methods, algorithms are trained to make classifications or predictions, uncovering key insights within data mining projects. These insights subsequently drive decision making within applications and businesses, ideally impacting key growth metrics. As big data continues to expand and grow, the market demand for data scientists will increase, requiring them to assist in the identification of the most relevant business questions and subsequently the data to answer them.

6.2 Technologies Used in The Application

6.2.1 Python

The Python integrated development environment (IDE) is a cross-platform application for Microsoft Windows, mac OS, and Linux) that is written in the C programming language. It originated from the IDE for the languages Processing.

The Python Programming Language is basically a framework built on top of C. You can argue that it's not a real programming language in the traditional term, but I think this helps avoiding confusion for beginners.

6.2.2 Reasons for Using Python

From development to deployment and maintenance, Python helps developers be productive and confident about the software they're building. Benefits that make Python the best fit for machine learning and AI-based projects include simplicity and consistency, access to great libraries and frameworks for AI and machine learning (ML), flexibility, platform independence, and a wide community. These add to the overall popularity of the language.

6.3 Reasons for Using Machine Learning

Today, we have many machine learning algorithms that can help us classify abnormal transactions. The only requirement is the past data and the suitable algorithm that can fit our data in a better form.Traditional rules-based expert systems are not enough to catch fraud. They can do an excellent job of uncovering known patterns; but alone aren't very effective at uncovering unknown schemes, adapting to new fraud patterns, or handling fraudsters' increasingly sophisticated techniques. And this is where machine learning becomes necessary for fraud detection.

For the above reason, we have used Machine Learning technology to verify the frauds in unbalanced datasets by using python libraries.

6.4 Machine Learning in Diabetes Prediction

Machine Learning algorithm helps us to train a ML model and predicts the outputs of the future unknown data values which is a critical task even for the doctors. Machine Learning model reduce the errors in predictions and increase the accuracy, which is most important in the medical field.

6.4.1 Supervised and Unsupervised Machine Learning

- **Supervised Machine Learning:** A model that is trained on a set of properly "labeled" transactions. Each transaction is tagged as either fraud or non-fraud. Supervised machine learning model accuracy is directly correlated with the amount

  of clean, relevant training data. Common supervised machine learning methods include linear regression, logistic regression,

- **Unsupervised Machine Learning:** A model that is trained in cases where tagged transaction data is relatively thin or non-existent. Unsupervised models are designed to discover outliers that represent previously unseen forms of fraud. In the real world of fraud detection, well labeled data is very rare. Therefore, supervised machine learning methods alone cannot do a good job and unsupervised learning will play an important role in the war.

The main distinction between the two approaches is the use of labelled datasets. To put it simply, supervised learning uses labelled input and output data, while an unsupervised learning algorithm does not.

In supervised learning, the algorithm "learns" from the training dataset by iteratively making predictions on the data and adjusting for the correct answer. While supervised learning models tend to be more accurate than unsupervised learning models, they require upfront human intervention to label the data appropriately. For example, a supervised learning model can predict how long your commute will be based on the time of day, weather conditions and so on. But first, you'll have to train it to know that rainy weather extends the driving time.

Unsupervised learning models, in contrast, work on their own to discover the inherent structure of unlabelled data. Note that they still require some human intervention for validating output variables. For example, an unsupervised learning model can identify that online shoppers often purchase groups of products at the same time. However, a data analyst would need to validate that it makes sense for a recommendation engine to group baby clothes with an order of diapers, applesauce and sippy cups.

CHAPTER-7

IMPLEMENTATION AND RESULTS

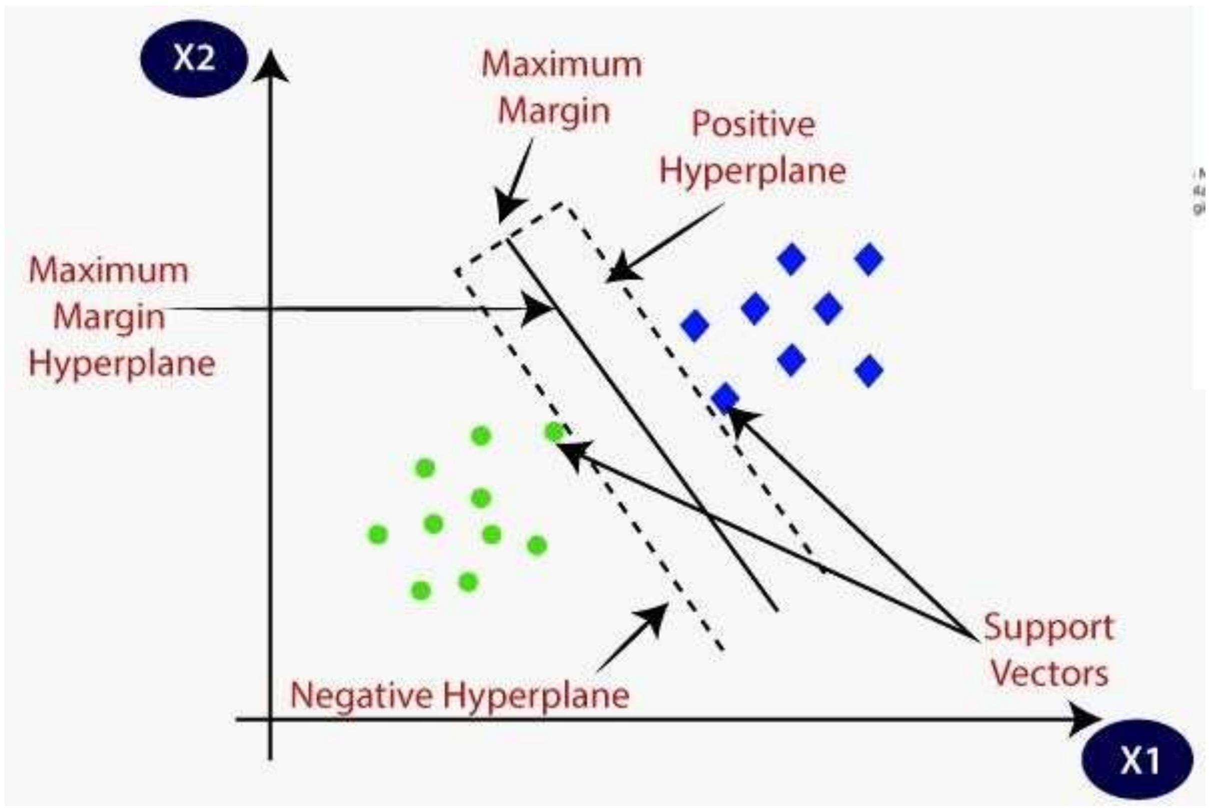
After Requirements Gathering, Analysis, and Design, there is the Implementation phase. In this step, the deliverable output was implemented. For the software part of our system, during this step, we translated the solution domain into source code, which included implementing the attributes and methods of each object and integrating all the objects such that they function as a single system. For the hardware part of the system, during this step we have written the code in a python tool (IDE) and executed the program for the accuracy of imbalanced dataset of the diabetes, from which it will receive commands. The first step involves the importing required modules and the datasets. Secondly, we are applying support vector machine algorithm on the European dataset that is fully unbalanced.

7.1 Algorithms Used

We are using support vector machine algorithm

7.1.1 support vector machine

Support Vector Machine (SVM) is a supervised machine learning algorithm used for both classification and regression. Though we say regression problems as well it’s best suited for classification. The main objective of the SVM algorithm is to find the optimal hyperplane in an Ndimensional space that can separate the data points in different classes in the feature space. The hyperplane tries that the margin between the closest points of different classes should be as maximum as possible. The dimension of the hyperplane depends upon the number of features. If the number of input features is two, then the hyperplane is just a line. If the number of input features is three, then the hyperplane becomes a 2-D plane. It becomes difficult to imagine when the number of features exceeds three.



7.2 CODE

Importing the dependencies

```
import numpy as np
 import pandas as pd
 from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
from sklearn import svm
from sklearn.metrics import accuracy_score
```

Data Collecton And Analysis

```
#loading the diabetes dataset to a pandas DataFrame
diabetes_dataset = pd.read_csv('/content/diabetes
(1).csv')
```

```
pd.read_csv?



# printing the first 5 rows of the dataset diabetes_dataset.head()



# number of rows and Columns in this dataset diabetes_dataset.shape


# getting the statistical measures of the data diabetes_dataset.describe()
diabetes_dataset['Outcome'].value_counts()


diabetes_dataset.groupby('Outcome').mean()


# separating the data and labels
X = diabetes_dataset.drop(columns = 'Outcome', axis=1)
Y = diabetes_dataset['Outcome']


print(X)


print(Y)
```

Data Standardization

```
scaler = StandardScaler()


standardized_data = scaler.transform(X)


print(standardized_data)


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

```
print(X)


print(Y)
```

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)
```

Training the model

```
classifier = svm.SVC(kernel='linear')


#training the support vector Machine Classifier classifier.fit(X_train,
Y_train)
```

Model Evaluation

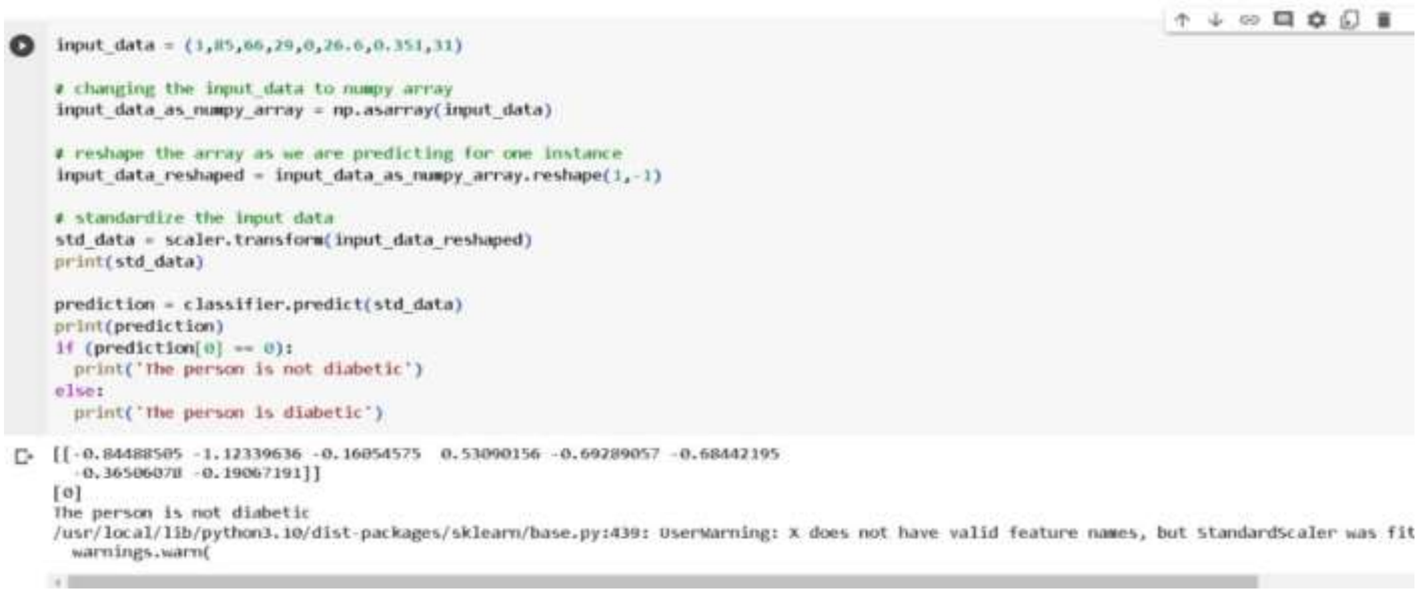
Accuracy score

```
# accuracy score on the training data
X_train_prediction = classifier.predict(X_train)
 training_data_accuracy = accuracy_score(X_train_prediction, Y_train)



print('Accuracy score of the training data : ', training_data_accuracy)



# accuracy score on the test data
X_test_prediction = classifier.predict(X_test)
test_data_accuracy = accuracy_score(X_test_prediction, Y_test)
print('Accuracy score of the test data : ', test_data_accuracy)
```

Making a Predicting System

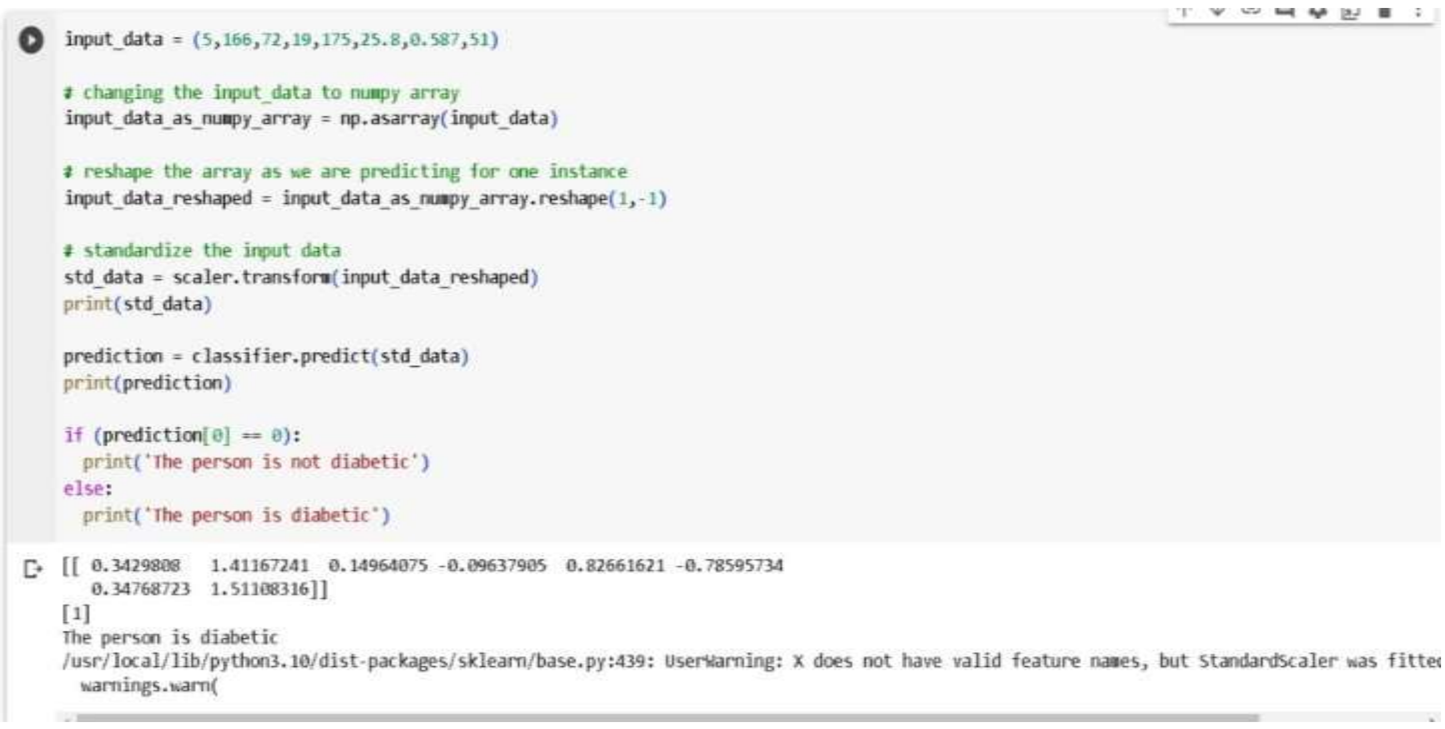
```
input_data = (1,85,66,29,0,26.6,0.351,31)

# 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')
```

7.3 Screenshots



```
input_data = (1,85,66,29,0,26.6,0.351,31)

# 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')
```

```
[[-0.84488505 -1.12339636 -0.16054575  0.53090156 -0.69289057 -0.68442195
  -0.36506078 -0.19067191]]
[0]
The person is not diabetic
/usr/local/lib/python3.10/dist-packages/sklearn/base.py:439: UserWarning: X does not have valid feature names, but StandardScaler was fit
  warnings.warn(
```



```
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')
```

```
[[ 0.3429808   1.41167241  0.14964075 -0.09637905  0.82661621 -0.78595734
   0.34768723  1.51108316]]
[1]
The person is diabetic
/usr/local/lib/python3.10/dist-packages/sklearn/base.py:439: UserWarning: X does not have valid feature names, but StandardScaler was fitted
  warnings.warn(
```

CHAPTER-8

TESTING AND VALIDATION

Testing a machine learning model for diabetes prediction involves several steps to ensure that the model performs accurately and reliably. Here's a high-level overview of the process:

Data Collection and Preprocessing:

Gather a dataset containing relevant features (e.g., age, BMI, family history, glucose levels) and the target variable (diabetes status - yes/no).

Preprocess the data by handling missing values, encoding categorical variables, and normalizing or standardizing numeric features.

Data Splitting:

Split the dataset into training, validation, and test sets. Common splits are 70% for training, 15% for validation, and 15% for testing.

Model Selection:

Choose an appropriate machine learning algorithm for the task. Common choices for classification problems like diabetes prediction include logistic regression, decision trees, random forests, support vector machines, or neural networks.

Model Training:

Train the selected model using the training dataset. Tune hyperparameters to optimize model performance. Use the validation set to fine-tune the model.

Model Evaluation:

Evaluate the model's performance using various metrics such as accuracy, precision, recall, F1score, and ROC AUC on the validation set.

Model Testing:

Use the test dataset to assess the model's performance on unseen data. This provides a more accurate estimate of how the model will perform in real-world scenarios.

Performance Metrics:

Calculate various performance metrics, including accuracy, precision, recall, F1-score, and ROC AUC, on the test dataset to evaluate the model's generalization performance.

Cross-Validation :

If needed, perform k-fold cross-validation to get a more robust estimate of the model's performance.

Model Interpretation :

Interpretability is crucial, especially in healthcare applications. Use techniques like SHAP values or LIME to explain the model's predictions.

Deployment :

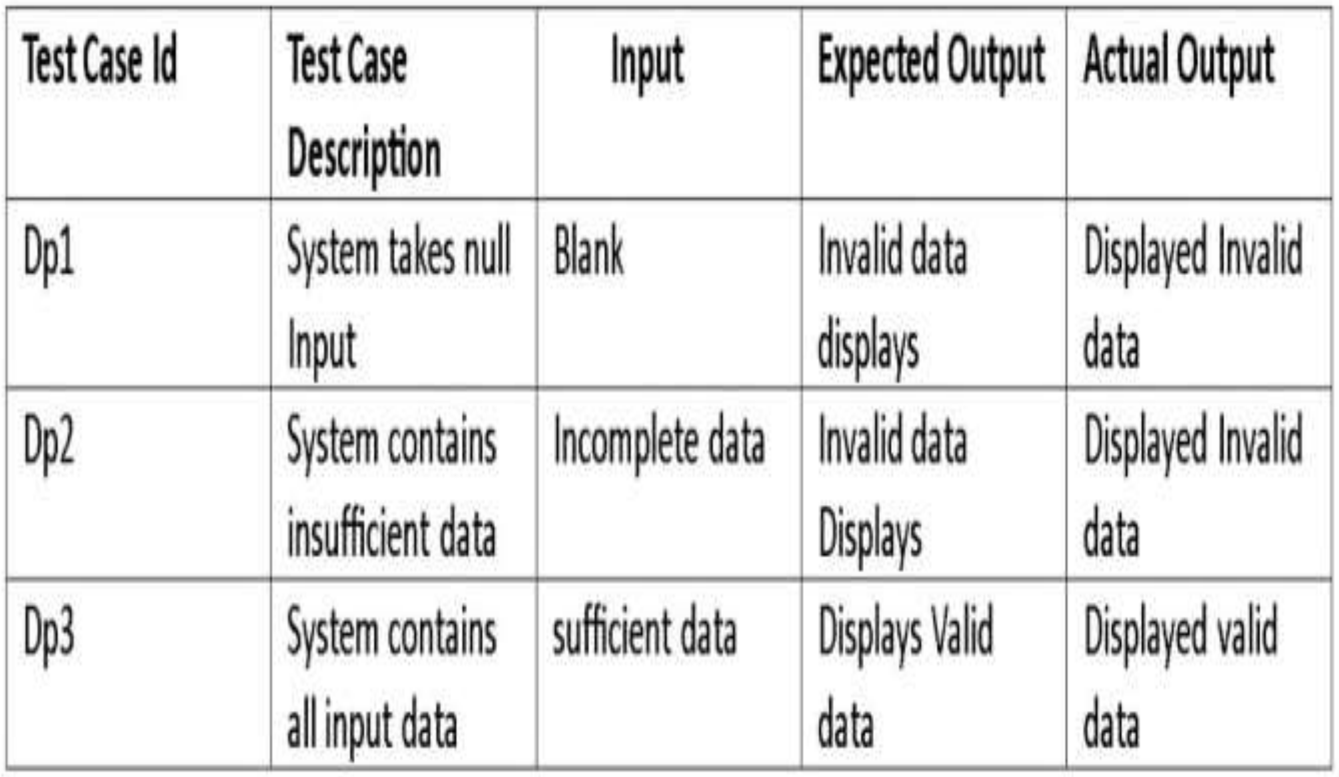
If the model performs satisfactorily, consider deploying it in a real-world healthcare setting, ensuring it complies with relevant regulations and ethical considerations.

Monitoring and Maintenance:

Continuously monitor the model's performance in the production environment and retrain it periodically with new data to ensure it stays up-to-date and accurate.

It's important to note that diabetes prediction in a real-world healthcare setting requires careful consideration of patient privacy, regulatory compliance (e.g., HIPAA in the United States), and ethical guidelines. Additionally, involving medical experts in the model development and validation process is crucial to ensure the model's safety and effectiveness in clinical practice.

Test Case:



Test Case Id	Test Case Description	Input	Expected Output	Actual Output
Dp1	System takes null Input	Blank	Invalid data displays	Displayed Invalid data
Dp2	System contains insufficient data	Incomplete data	Invalid data Displays	Displayed Invalid data
Dp3	System contains all input data	sufficient data	Displays Valid data	Displayed valid data

CHAPTER 9

CONCLUSION AND FUTURE ENHANCEMENT :

The main objective of project is to classify and predict diabetes using machine learning algorithms is being discussed throughout the project. We build the model using machine learning algorithm that is support vector machine, it is supervised learning algorithm in machine learning.

There is no interactive tool for users to predict diabetes. Due to small size of dataset, an accuracy of 77% for the model is achived. This issue can be resolved by continuously gathering more data in training model.

CHAPTER-10

REFERENCES

Text Books:

1.Python Machine Learning" by Sebastian Raschka

2.Fundamentals of Machine Learning for Predictive Data Analytics" by John D. Kelleher, Brian Mac Namee, and Aoife D'Arcy.

3.Python 3: The Comprehensive Guide to Hands-On Python Programming (1st Edition)

Web Links:

4.https://www.dropbox.com/s/uh7o7uyeghqkhoy/diabetes.csv?dl=0 (dataset)

5.https://www.youtube.com/watch?v=xUE7SjVx9bQ&t=38s(youtube)

6.https://colab.research.google.com/drive/126NuoDtraOm4o-H9gDN40mGHmGcUocAa