A Project Report On

### ABSTRACT

The diabetes is one of lethal diseases in the world. It is additional a inventor of various varietiesof disorders for example: coronary failure, blindness, urinary organ diseases etc. In such case thepatient is required to visit a diagnostic center, to get their reports after consultation. Due to everytime they have to invest their time and currency. But with the growth of Machine Learningmethods we have got the flexibility to search out an answer to the current issue, we have gotadvanced system mistreatment information processing that has the ability to forecast whether thepatient has polygenic illness or not. Furthermore, forecasting the sickness initially ends up inproviding the patients before it begins vital. Information withdrawal has the flexibility to removeunseen data from a large quantity of diabetes associated information.

.

|  |  |  |  |
| --- | --- | --- | --- |
| **LIST OF FIGURES** | | | |
| **S.NO** | **Figure No.** | **Title of Figure** | **Page No** |
| **1** | 4.1 | Architecture design | 8 |
| **2** | 4.3.1 | Usecase diagram | 10 |
| **3** | 4.3.2 | Sequenece diagram | 11 |
| **4** | 5.3.1 | Histogram of each figure | 16 |
| **5** | 5.3.2 | Pairplot of all features | 17 |
| **6** | 5.3.3 | Heatmap of feature collection | 17 |
| **7** | 5.3.4 | Heatmap of confusion matrix | 19 |
| **8** | 8.1 | SVM | 22 |
| **9** | 8.2 | Symbols in usecase diagram | 28 |

|  |  |  |  |
| --- | --- | --- | --- |
| **LIST OF TABLES** | | | |
| **S.NO** | **Table No.** | **Title of Table** | **Page No** |
| **1** | 1.5 | Data preview | 3 |
| **2** | 5.3 | Statistical summary | 13 |

***INDEX***

|  |  |
| --- | --- |
|  | **Page No** |
| **1. INTRODUCTION** | **1** |
| 1.1 Project Introduction | **1** |
| 1.2 Scope | **1** |
| 1.3 Project Overview | **2** |
| 1.4 Objectives | **2** |
| 1.5 Data Set | **3** |
| **2. LITERATURE SURVEY** | **4** |
| 2.1 Existing System | **4** |
| 2.2 Proposed System | **4** |
| 2.3 Related Work | **4** |
| **3. SYSTEM ANALYSIS** | **6** |
| 3.1 Functional Requirements | **6** |
| 3.2 Performance Requirements | **6** |
| 3.3 Software Requirements | **7** |
| 3.4 Hardware Requirements | **7** |
| **4. SYSTEM DESIGN** | **8** |
| 4.1 System Architecture | **8** |
| 4.2 Data Flow Diagram | **9** |
| 4.3 UML Diagrams | **10** |
| 4.3.1 Use Case Diagram | **10** |
| 4.3.2 Sequence Diagram | **11** |
| **5. IMPLIMENTATION AND RESUTS** | **12** |
| 5.1 Language / Technology Used | **12** |
| 5.2 Methods / Algorithms Used | **12** |
| 5.3 Sample Code | **13** |
| 5.4 Results (Accuracy) / Output Screens | **20** |
| **6. TESTING** | **21** |
| 6.1 Types of Testing | **21** |
| 6.2 Test Cases | **23** |
| **7. CONCLUSION** | **24** |
| **8. FUTURE SCOPE** | **25** |
| **9. BIBLIOGRAPHY** | **26** |
| **APPENDIX :** | **27** |

# 1. INTRODUCTION

* 1. PROJECT INTRODUCTION

Diabetes is an increasingly growing health issue due to inactive lifestyle. If it is detected in time then through proper medical treatment, adverse effects can be prevented. To help in early detection,machine learning technology can be used very reliably and efficiently.

So in this project, the objective is to predict whether the person has Diabetes or not using a machine learning model. Here we will use the Pima Indians dataset from the UCI Machine learning repository for testing and training the model. This dataset contains attributes like Bloodpressure,preg, Insulin, Age, BMI etc.

The aim of this analysis isto develop a system which might predict the diabetic risk level of a patient with a betteraccuracy. Model development is based on categorization methods as SVM algorithms .Our objective is to predict whetherthe patient has diabetes or notbased on various features like Glucose level, Insulin, Age, BMI. We will perform all the steps

from Data gathering to Model deployment. During Model evaluation, we compare variousmachine learning algorithms on the basis of accuracy scoremetric and find the best one. Thenwe create a web app using Flask which is a python micro framework.

* 1. **SCOPE**

The main aim of this project was to design and implement Diabetes Prediction Using Machine Learning Methods and Performance Analysis of that methods and it has been achieved successfully.

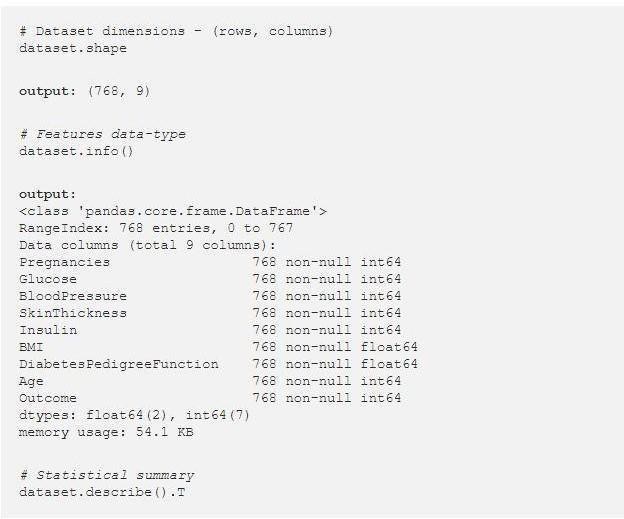
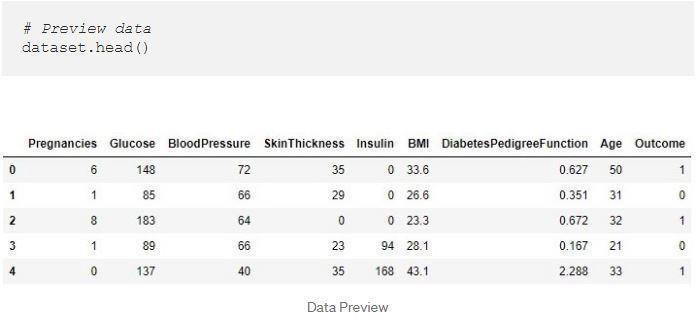
Diabetes is an increasingly growing health issue due to our inactive lifestyle. If it is detected in time then through proper medical treatment, adverse effects can be prevented. To help in early detection, technology can be used very reliably and efficiently. Using machine learning we have built a predictive model that can predict whether the patient is diabetes positive or not.

### PROJECT OVERVIEW

Our analysis finds five main predictors of diabetes: **glucose, pregnancy, body mass index, age, and diabetes pedigree function**. These risk factors of diabetes identified by the logistic

regression were validated by the decision tree and could help classify high-risk individuals and prevent, diagnose and manage diabetes.

1. There are a total of 768 records and 9 features in thedataset.
2. Each feature can be either of integer or float data type.
3. Some features like Glucose, Blood pressure, Insulin, BMI have zero values which represent missing data.
4. There are zero NaN values in the dataset.
5. In the outcome column, 1 represents diabetes positive and 0 represents diabetes negative.
   1. OBJECTIVES
      1. Predict if person is diabetes patient or not
      2. Find most indicative features of diabetes
      3. To find highest accuracy using support vector machine(SVM) classification algorithm
   2. DATASET



# LITERATURE SURVEY

### EXISTING SYSTEM

Existing method for diabetes detection is uses lab tests such as fasting blood glucose and oral glucose tolerance. However, this method is time consuming. And there are also machine learning techniques such as decision tree, with an accuracy of 68%,Random forest accuracy 71.2%.All of this techniques has some serious disadvantages such as decrease in accuracy levels , lack of efficency . This project focuses on building predictive model using machine learning algorithms and data mining techniques for diabetes prediction.

* 1. PROPOSED SYSTEM

The proposed system predicts the disease of diabetes in patients with maximum accuracy. We shall talk about various machine learning, the algorithm which can help in decision making and prediction. We shall use more than one algorithm to get better accuracy of prediction.

The disease dataset is given to the system which is then pre-processed so that the data is in a useable format for analysis. If the dataset is not structured or if the dataset is or if the dataset is huge or it has irrelevant features, we shall use feature extraction to extract the data. After this the data is trained and we apply a relevant machine learning algorithm to the dataset.

* 1. RELATED WORK

This diabetes prediction using machine learning enables patient to enter their details and check if the patient is diabetic or not.

* Patients need to visit the website then enter their details such as glucose level, body mass index(BMI),age and check the patient is diabetic or not.

This is most important phase which includes model build- ing for prediction of diabetes. In this we have implemented various machine learning algorithms which are discussed above for diabetes prediction.

Procedure of Proposed Methodology-

Step1: Import required libraries, Import diabetes dataset. Step2: Pre-process data to remove missing data.

Step3: Perform percentage split of 80% to divide dataset as Training set and 20% to Test set.

Step4: Select the machine learning algorithm i.e. K- Nearest Neighbor, Support Vector Machine, Decision Tree, Logistic regression, Random Forest and Gradient boosting algorithm.

Step5: Build the classifier model for the mentioned ma- chine learning algorithm based on training set.

Step6: Test the Classifier model for the mentioned ma- chine learning algorithm based on test set.

Step7: Perform Comparison Evaluation of the experi- mental performance results obtained for each classifier.

Step8: After analyzing based on various measures con- clude the best performing algorithm.

# SYSTEM ANALYSIS

* 1. FUNCTIONAL REQUIREMENTS

Functional Requirements are also called functinal specifications. In software and system engineering, a functional requirement can range from the high level abstract statement of the sender necessity to detailed mathematical functional requirement specification . Collecting the system functional and technical requirements in one place will give you clarity and create a requirement document to start the software selection process . Additionally,this document will serve as a constant point of reference for you and your team during software selection and implementation.

All the standard libraries like numpy, pandas, matplotlib and seaborn are imported in this step. We use numpy for linear algebra operations, pandas for using data frames, matplotlib and seaborn for plotting graphs.

* 1. PERFORMANCE REQUIREMENTS

Performance is measured in terms of the output provided by the application. Requirement specification plays an important part in the analysis of a system. Only when the requirement specifications are properly given, it is possible to design a system, which will fit into required environment. It rests largely with the users of the existing system to give the requirement specifications because they are the people who finally use the system. This is because the requirements have to be known during the initial stages so that the system can be designed according to those requirements. It is very difficult to change the system once it has been designed and on the other hand designing a system, which does not cater to the requirements of the user, is of no use. The requirement specification for any system can be broadly stated as given below:

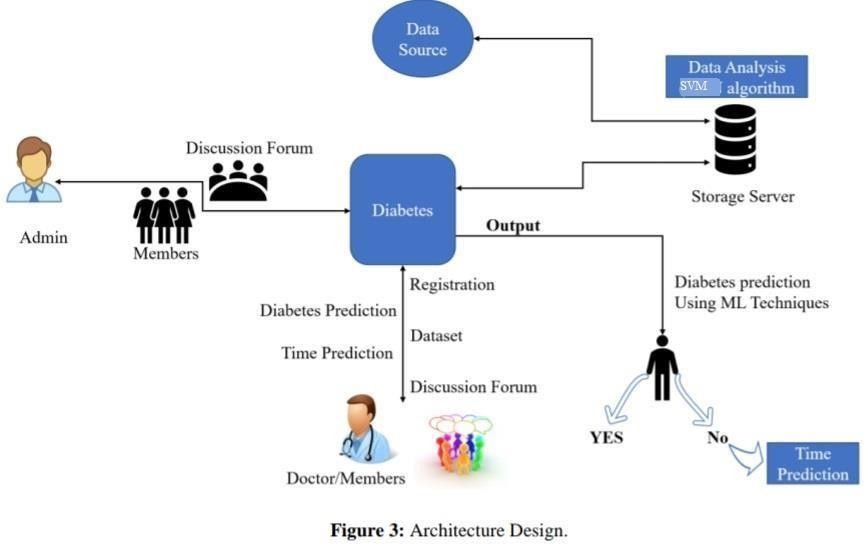
* The system should be able to interface with the existing system
* The system should be accurate
* The system should be better than the existing system The existing system is completely dependent on the user to perform all the duties.
  1. SOFTWARE REQUIREMENTS
* WINDOWS: Windows 7 and above
* LANGUAGES: python 3.7 and above
* LIBRARIES: pandas,numpy,matplotlib,seaborn,SVC,flask
  1. HARDWARE REQUIREMENTS
* 4 GB RAM
* 10GB HDD
* Intel 1.66 GHz Processor Pentium 4

## SYSTEM DESIGN

Systems design is the process of defining the architecture, components, modules, interfaces, and data for a system to satisfy specified requirements. One could see it as the application of systems theory to product development. Object-oriented analysis and design methods are becoming the most widely used methods for computer systems design.

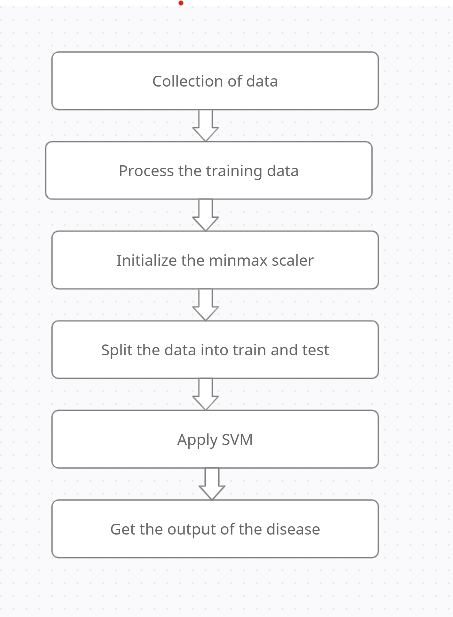
* 1. SYSTEM ARCHITECTURE

A system architecture is the conceptual model that defines the structure, behavior, and more views of a system. An architecture description is a formal description and representation of a system, organized in a way that supports reasoning about the structures and behaviors of the system.



* 1. Data Flow Diagram

A data-flow diagram is a way of representing a flow of data through a process or a system (usually an information system). The DFD also provides information about the outputs and inputs of each entity and the process itself. A data-flow diagram has no control flow — there are no decision rules and no loops.



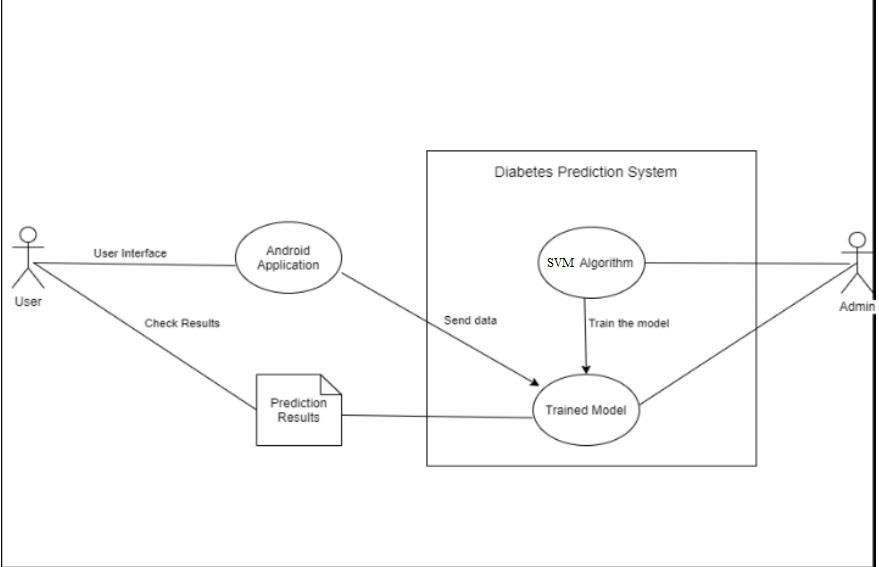
* 1. UML DIAGRAMS
     1. USE CASE DIAGRAM

The Use Case diagram of the project disease prediction using machine learning consist ofall thevarious aspects a normal use case diagram requires. This use case diagram shows how from starting the model flows from one step to another, like he enter into the system then enters all the information’s and all other general information along with the symptoms that goes into the system,compares with the prediction model and if true is predicts the appropriate results otherwise it shows the details where the user if gone wrong while entering the information’s and it also shows the appropriate precautionary measure for the user to follow. Here the use case diagram of all the entities are linked to each other where the user gets started with the system.

**PURPOSE OF USE CASE DIAGRAMS**

Used to gather the requirements of a system. Used to get an outside view of a system.

Identify the external and internal factors influencing the system. Show the interaction among the requirements are actors.



* + 1. SEQUENCE DIAGRAM

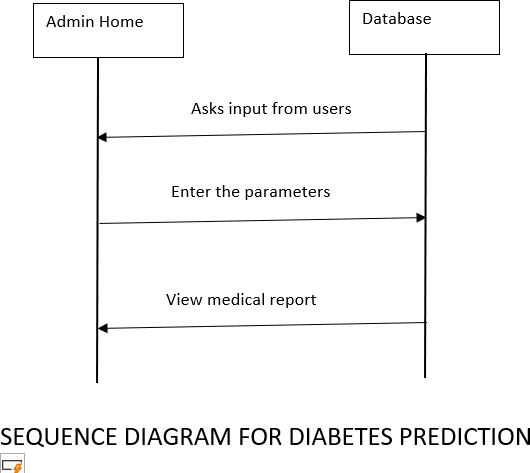
The Sequence diagram of the project disease prediction using machine learning consist of all thevarious aspects a normal sequence diagram requires. This sequence diagram shows how from starting the model flows from one step to another, like he enter into the system then enters all theinformation’s and all other general information along with the symptoms that goes into the system,compares with the prediction model and if true is predicts the appropriate results otherwise it shows the details where the user if gone wrong while entering the information’s and it also shows the appropriate precautionary measure for the user to follow. Here the sequence of all the entities are linked to each other where the user gets started with the system.

PURPOSE OF SEQUENCE DIAGRAM

To model the flow of control by time sequence.

To model the flow of control by structural organizations. For forward engineering.

For reverse engineering.



* 1. IMPLEMENTATION AND RESULTS

The implementation stage of any project is a true display of the defining moments that make a project a success or a failure. The implementation stage is defined as the system or system modifications being installed and made operational in a production environment. The phase is initiated after the system has been tested and accepted by the user. This phase continues until the system is operating in production in accordance with the defined user requirements.

* 1. LANGUAGE / TECHNOLOGY USED

Python is the language used for developing the predictive system using Support Vector Machine (SVM) algorithm.

* 1. LIBRARIES / ALGORITHMS USED
* **NUMPY:** which stands for Numerical Python, is a library consisting of multidimensional array objects and a collection of routines for processing those arrays.
* **PANDAS:** is a Python library. Pandas is used to analyze data, Learning by Reading.
* **SEABORN:** The Python Seaborn library lets you visualize data using pair plots that produce a matrix of relationships between each variable in the dataset. In the below plot, all the plots are histograms that represent the distribution of each feature.
* **Support Vector Machine (SVM):** It is a supervised machine learning algorithm capable of performing classification, regression and even outlier detection.
  1. SAMPLE CODE PROCEDURE/ALGORITHM:

Start

CREATING A MACHINE LEARNING MODEL

{

IMPORTING DEPENDENCIES;

DATA COLLECTION AND ANALYSIS

{

loading the data from csv file to a Pandas DataFrame;

Data Normalization(removing inconsistent values);

}

DATA PREPROCESSING{

Separating the features & Target;

Splitting the data to training data & Test data;

}

Data Standardization; Model Training; Model Evaluation

{

}

calculating accuracy of training dataset; calculating accuracy of test dataset;

}

*BUILDING A PREDICTIVE SYSTEM

*{

user\_input\_data(parameters/attributes); user\_input\_data to numpy.array; standardization of data:

std\_data = scaler.transform(input\_data); #predicting

prediction = predict(std\_data) print(prediction);

if prediction is ‘0’ then do

print("The Person does not have diabetes Disease");

else do

print("The Person has diabetes Disease");

}

End;

CODE:

*# Importing libraries* import pandas as pd import numpy as np

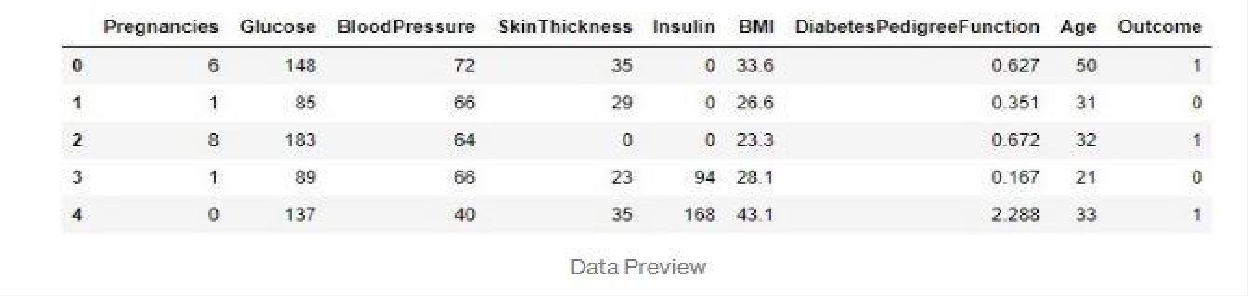
import matplotlib.pyplot as plt import seaborn as sns

*# Importing dataset*

dataset = pd.read\_csv('diabetes.csv')

*# Preview data*

dataset.head()



# Dataset dimensions - (rows, columns) dataset.shape**output:** (768, 9)*# Features data-type* dataset.info()**output:**

<class 'pandas.core.frame.DataFrame'> RangeIndex: 768 entries, 0 to 767 Data columns (total 9 columns):

Pregnancies 768 non-null int64

Glucose 768 non-null int64

BloodPressure 768 non-null int64

SkinThickness 768 non-null int64

Insulin 768 non-null int64

BMI 768 non-null float64

DiabetesPedigreeFunction 768 non-null float64 Age 768 non-null int64

Outcome 768 non-null int64 dtypes: float64(2), int64(7)

memory usage: 54.1 KB*# Statistical summary*

dataset.describe().T



*# Count of null values* dataset.isnull().sum()**output:** Pregnancies 0

Glucose 0

BloodPressure 0

SkinThickness 0

Insulin 0

BMI 0

DiabetesPedigreeFunction 0

Age 0

Outcome 0

dtype: int64

*# Outcome countplot*

sns.countplot(x = 'Outcome',data = dataset)

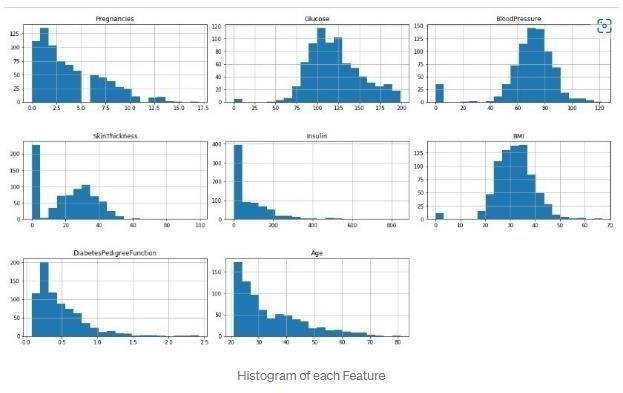
*# Histogram of each feature*

import itertoolscol = dataset.columns[:8] plt.subplots(figsize = (20, 15))

length = len(col)for i, j in itertools.zip\_longest(col, range(length)):

plt.subplot((length/2), 3, j + 1) plt.subplots\_adjust(wspace = 0.1,hspace = 0.5) dataset[i].hist(bins = 20)

plt.title(i) plt.show()



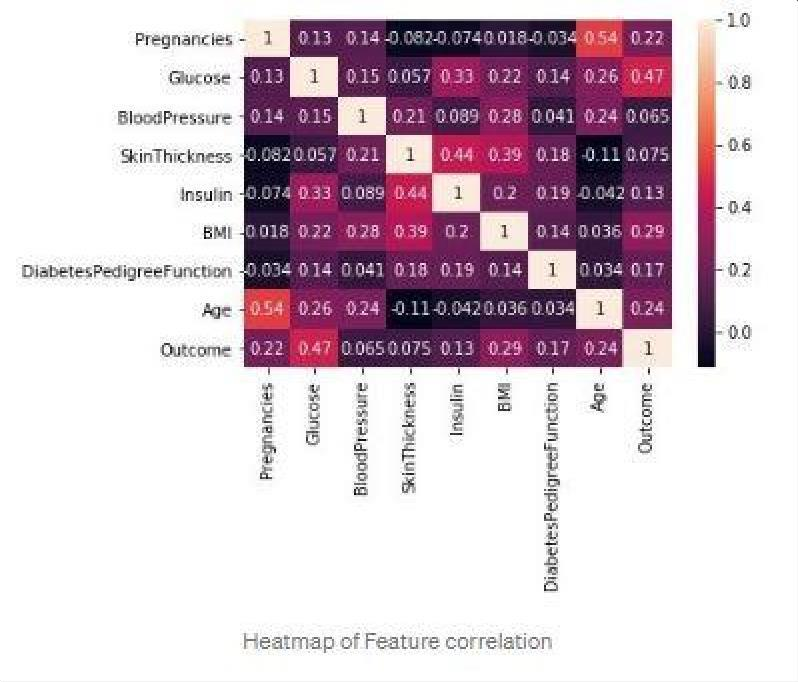
*# Pairplot*

sns.pairplot(data = dataset, hue = 'Outcome') plt.show()



*# Heatmap*

sns.heatmap(dataset.corr(), annot = True) plt.show()





dataset[["Glucose", "BloodPressure", "SkinThickness", "Insulin", "BMI"]] = dataset[["Glucose", "BloodPressure", "SkinThickness", "Insulin", "BMI"]].replace(0, np.NaN)*# Count of NaN* dataset.isnull().sum()**Output:**

Pregnancies

dataset["Glucose"].fillna(dataset["Glucose"].mean(), inplace = True)

dataset["Insulin"].fillna(dataset["Insulin"].mean(), inplace = True) dataset["BMI"].fillna(dataset["BMI"].mean(), inplace = True)

*# Feature scaling using MinMaxScaler*

from sklearn.preprocessing import MinMaxScaler sc = MinMaxScaler(feature\_range = (0, 1))

dataset\_scaled = sc.fit\_transform(dataset\_new)dataset\_scaled = pd.DataFrame(dataset\_scaled)

*# Selecting features - [Glucose, Insulin, BMI, Age]*

X = dataset\_scaled.iloc[:, [1, 4, 5, 7]].values

Y = dataset\_scaled.iloc[:, 8].values*# Splitting X and Y*

from sklearn.model\_selection import train\_test\_split

X\_train, X\_test, Y\_train, Y\_test = train\_test\_split(X, Y, test\_size

= 0.20, random\_state = 42, stratify = dataset\_new['Outcome'] )*#*

*Checking dimensions*

print("X\_train shape:", X\_train.shape) print("X\_test shape:", X\_test.shape) print("Y\_train shape:", Y\_train.shape) print("Y\_test shape:", Y\_test.shape)**Output:** X\_train shape: (614, 4)

X\_test shape: (154, 4)

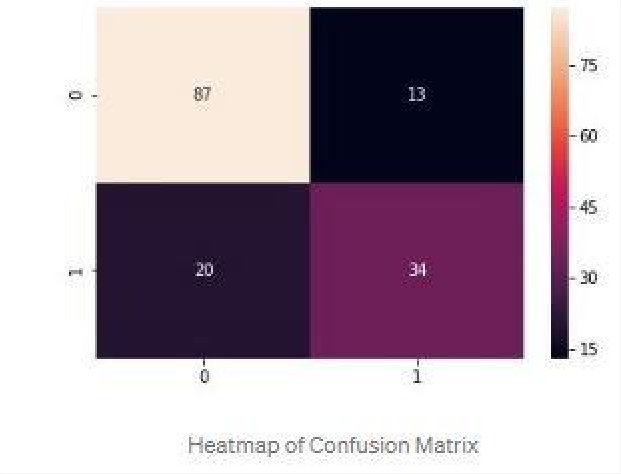
Y\_train shape: (614,)

Y\_test shape: (154,)

*# Support Vector Classifier Algorithm*

from sklearn.svm import SVC

svc = SVC(kernel = 'linear', random\_state = 42) svc.fit(X\_train, Y\_train)*# Making predictions on test dataset* Y\_pred = svc.predict(X\_test)



*# Evaluating using accuracy\_score metric*

from sklearn.metrics import accuracy\_score

accuracy = accuracy\_score(Y\_test, Y\_pred)print("Accuracy: " + str(accuracy \* 100))**Output:**

Accuracy: 73.37662337662337*# Confusion matrix* from sklearn.metrics import confusion\_matrix cm = confusion\_matrix(Y\_test, Y\_pred) cm**Output:**

array([[87, 13],

[20, 34]], dtype=int64)*# Heatmap of Confusion matrix*

sns.heatmap(pd.DataFrame(cm), annot=True)

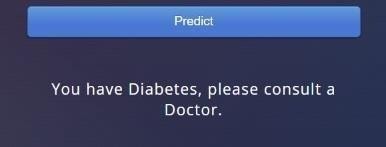
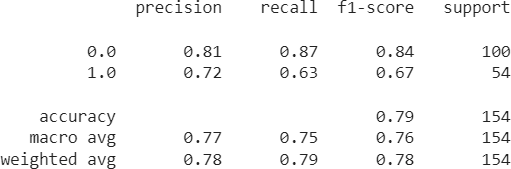
*# Classification report*

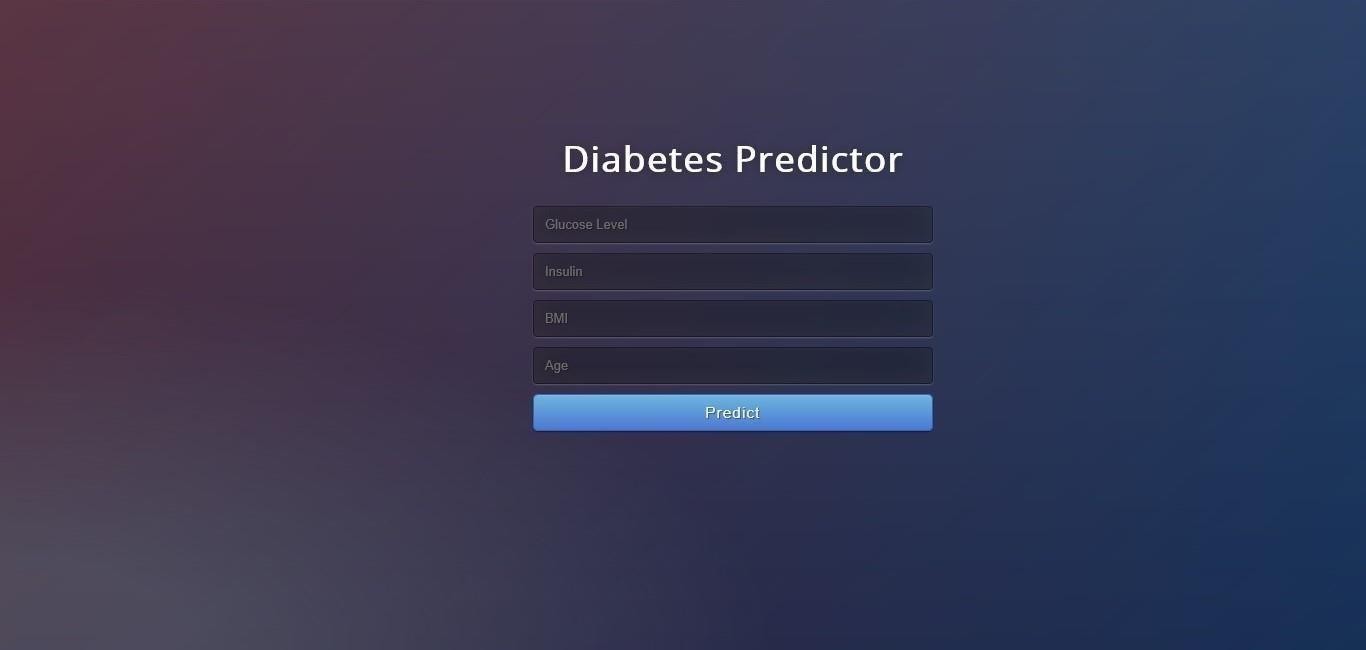
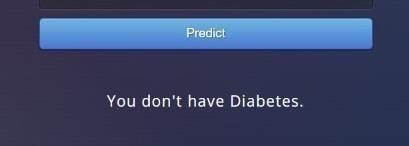
from sklearn.metrics import classification\_report print(classification\_report(Y\_test, Y\_pred))**Output:**

precision recall f1-score support

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 0.0 | 0.81 | 0.87 | 0.84 | 100 |
| 1.0 | 0.72 | 0.63 | 0.67 | 54 |
| micro avg | 0.79 | 0.79 | 0.79 | 154 |
| macro avg | 0.77 | 0.75 | 0.76 | 154 |
| weighted avg | 0.78 | 0.79 | 0.78 | 154 |

* 1. RESULTS(ACCURACY)/OUTPUT SCREEN





# TESTING

In general, diabetes disease can't be cured, but medications can help control the symptoms, often dramatically. In some more advanced cases, surgery may be advised.Your health care provider may also recommend lifestyle changes, especially ongoing aerobic exercise. In some cases, physical therapy that focuses on balance and stretching is important. A speech- language pathologist may help improve speech problems.

Medications may help you manage problems with walking, movement and tremor. These medications increase or substitute for dopamine.

People with diabetes disease have low brain dopamine concentrations. However, dopamine can'tbe given directly as it can't enter the brain.

You may have significant improvement of your symptoms after beginning diabetes disease treatment. Over time, however, the benefits of drugs frequently diminish or become less consistent. You can usually still control your symptoms well.

**What Actully we do** is initially we take a dataset of 200 members including diabetes and non diabetic diseased persons.Here in dataset ,the attributes are minimum voice frequency(Hz),maximum voice frequency(Hz),Jitter,Shimmer,PPQ,RAP and other attributes.

After importing this dataset into our program,we process this data so that it normalizes the values(removing null values if present which avoids redundancy).This can be achieved by using StandardScaler() method.Dividing the data into testing and training data by using train\_test\_split() function from sklearn library.Generally we split the data as Train data -80% andtest data – 20% so that we csn train with different data.

Importing SVM algorithm by using sklearn library.Train the module with train data.The svm algorithm generally classifies the data into n- dimensions.After training id divides data into diseases exist and not exist.To get complete knowledge we cross check by using test data .

Accuracy() is used to find the accuracy of this model.

After the model is trained ,user is ready to give input (taken through numpy) and by using predictby using predict() .By using control statements ,if predicted value is 0 then the person is not having disease ,if 1 diabetes exist.

TEST CASES

Testcase 1:

User input\_data: (197.07600,206.89600,192.05500,0.00289,0.00001,0.00166,0.00168,0.00498,0.01098,0.09700,0

.00563,0.00680,0.00802,0.01689,0.00339,26.77500,0.422229,0.741367,-

7.348300,0.177551,1.743867,0.085569)

Output :

Person have diabetes .

Testcase 2:

User input\_data : (197.07600,206.89600,192.05500,0.00289,0.00001,0.00166,0.00168,0.00498,0.01098,0.09700,0

.00563,0.00680,0.00802,0.01689,0.00339,26.77500,0.422229,0.741367,-

7.348300,0.177551,1.743867,0.085569)

Output :

Person don’t have diabetes.

# CONCLUSION

Diabetes is a slow killer with no known curable treatments. However, its complications can be reduced through proper awareness and timely treatment. Three major complications are related to blindness, kidney damage and heart attack. It is important to keep the blood glucose levels of patients under strict control for avoiding the complications. One of the difficulties with tight control of glucose levels in the blood is that such attempts may lead to hypoglycemia that creates much severe complications than an increased level of blood glucose. Researchers now look for alternative methods for diabetes treatment. The goal of this paper is to give a general idea of the current status of diabetes research. The author believes that diabetes is one of the highly demanding research topics of the new century and wants to encourage new researchers to take up the challenges.

# FUTURE SCOPE

A rapid review of the current and future scope of stem cell research Author links open overlay panel SheriffSheik Abdulazeez Open Access funded by King Saud University Diabetes mellitus is a major health concern of the developing and developed nations across the globe. This devastating disease accounts for the 5% deaths around the world annually. The current treatment methods do not address the underlying causes of the disease and have severe limitations. Stem cells are unique cells with the potential to differentiate into any type of specialized cells. This feature of both adult and embryonic stem cells was explored in great detail by the scientists around the world and are successful in producing insulin secreting cells. The different type of stem cells (induced pluripotent stem cells (iPSCs), embryonic stem cells (ESCs) and adult stem cells) proves to be potent in treating diabetes with certain limitations

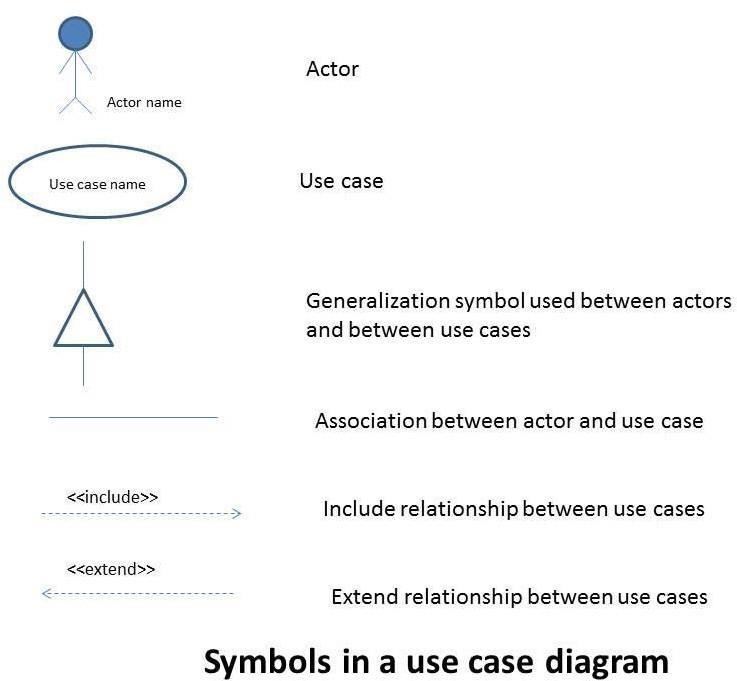
# BIBILOGRAPHY

* + - https://[www.frontiersin.org/articles/10.3389/fgene.2018.00515/full](http://www.frontiersin.org/articles/10.3389/fgene.2018.00515/full)
    - https://[www.hindawi.com/journals/jhe/2021/9930985/](http://www.hindawi.com/journals/jhe/2021/9930985/)
    - I Support Vector Machine:December 20, 201 by Funny Notebooks (Author)
    - Support Vector Machines Machine Learning: Notebook Planner - 6x9 inch Daily Planner Journal, To Do List Notebook, Daily Organizer, 114 Pages Paperback – May 18, 2022 by TYWANDA LIMON (Author)
    - https://doi.org/10.3389/fgene.2018.00515

**APPENDIX : UNIFIED MODELING LANGUAGE**

The Unified Modeling Language (UML) is a general-purpose visual modeling language that is used to specify, visualize, construct, and document the artifacts of a software system. It captures decisions and understanding about systems that must be constructed. It is used to understand, design, browse, configure, maintain, and control information about such systems. It is intended for use with all development methods, lifecycle stages, application domains, and media. The modeling language is intended to unify past experience about modeling techniques and to incorporate current software best practices into a standard approach. UML includes semantic concepts, notation, and guidelines. It has static, dynamic, environmental, and organizational parts. It is intended to be supported by interactive visual modeling tools that have code generators and report writers. The UML specification does not define a standard process but is intended to be useful with an iterative development process. It is intended to support most existing object oriented development processes. The UML captures information about the static structure and dynamic behavior of a system. A system is modeled as a collection of discrete objects that interact to perform work that ultimately benefits an outside user. The static structure defines the kinds of objects important to a system and to its implementation, as well as the relationships among the objects. The dynamic behavior defines the history of objects over time and the communications among objects to accomplish goals. Modeling a system from several separate but related viewpoints permits it to be understood for different purposes. The UML also contains organizational constructs for arranging models into packages that permit software teams to partition large systems into workable pieces, to understand and control dependencies among the packages, and to manage the versioning of model units in a complex development environment. It contains constructs for representing implementation decisions and for organizing run-time elements into components. 53 | P a g e UML is not a programming language. Tools can provide code generators from UML into a variety of programming languages, as well as construct reverseengineered models from existing programs. The UML is not a highly formal language intended for theorem proving. There are a number of such languages, but they are not easy to understand or to use for most purposes. The UML is a generalpurpose modeling language. For specialized domains, such as GUI layout, VLSI circuit design, or rule-based artificial intelligence, a more specialized tool with a special language might be appropriate. UML is a discrete modeling language. It is not intended to model continuous systems such as those

found in engineering and physics. UML is intended to be a universal general-purpose modeling language for discrete systems such as those made of software, firmware, or digital logic



.

