A

Project Report

on

A MODIFIED MACHINE LEARNING TECHNIQUES FOR DIABETES PREDICTION

By

Mahajan Bhagyashri Liladhar (21517220171124510144)
Chaudhari Mamta Santosh (21517220171124510095)
Shinde Ashwini Sunil (21517220171124510086)
Ahire Nikita Sunil (21517220171124510119)

The Shirpur Education Society's

Department of Computer Engineering

R. C. Patel Institute of Technology Shirpur - 425405.
Maharashtra State, India

[2020-21]

A
Project Report
on

A MODIFIED MACHINE LEARNING TECHNIQUES FOR DIABETES PREDICTION

Submitted By

Mahajan Bhagyashri Liladhar (21517220171124510144)
Chaudhari Mamta Santosh (21517220171124510095)
Shinde Ashwini Sunil (21517220171124510086)
Ahire Nikita Sunil (21517220171124510119)

Guided By

Prof. S. A. Pinjari

The Shirpur Education Society's

Department of Computer Engineering

R. C. Patel Institute of Technology Shirpur - 425405.

Maharashtra State, India

[2020-21]

The Shirpur Education Society's

R. C. Patel Institute of Technology
Shirpur, Dist. Dhule (M.S.)
Department of Computer Engineering

Maharashtra State, India

CERTIFICATE

This is to certify that the project entitled“ A MODIFIED MACHINE LEARNING TECHNIQUES FOR DIABETES PREDICTION”has been carried out by team: :

Mahajan Bhagyashri Liladhar(21517220171124510144)
Chaudhari Mamta Santosh(21517220171124510095)
Shinde Ashwini Sunil (21517220171124510086)
Ahire Nikita Sunil (21517220171124510119)

under the guidance of Prof. R. J. Jaiswal in partial fulfillment of the requirement for the de-gree of Bachelor of Engineering in Computer Engineering of Dr. Babasaheb Ambedkar Tech-nological University, Lonere during the academic year 2020-21.

Date:
Place: Shirpur

Guide
Prof. S. A. Pinjari

Project Coordinator
Prof. R.B. Wagh

H. O. D.
Prof. Dr. Nitin N. Patil

Principal
Prof. Dr. J. B. Patil

Acknowledgment

No volume of words is enough to express my gratitude towards my guide, (S.A.Pinjari), Associate Professor in Computer Engineering Department, who has been very concerned and have aided for all the material essential for the preparation of this work. He has helped me to explore this vast topic in an organized manner and provided me with all the ideas on how to work towards a research oriented venture.

we wish to express our sincere gratitude towards Project Coordinator Prof. Dr.R. B. Wagh for his timely suggestions and instructions.

we are also thankful to Prof. Dr. Nitin N. Patil, Head of Department, Computer Engineering, for the motivation and inspiration that triggered me for the project work. we are thankful to Prof. Dr. J. B. Patil, Principal, R. C. P. I. T., Shirpur for the support and encouragement.

Mahajan Bhagyashri Liladhar(21517220171124510144)
Chaudhari Mamta Santosh(21517220171124510095)
Shinde Ashwani Sunil (21517220171124510086)
Ahire Nikita Sunil (21517220171124510119)

Contents

List of Figures

ABSTRACT

Diabetes is among critical diseases and lots of people are suffering from this disease. Age, obesity, lack of exercise, hereditary diabetes, living style, bad diet, high blood pressure, etc. can cause Diabetes Mellitus. People having diabetes have high risk of diseases like heart disease, kidney disease, stroke, eye problem, nerve damage, etc. Current practice in hospital is to collect required information for diabetes diagnosis through various tests and appropriate treatment is provided based on diagnosis. Big Data Analytics plays an significant role in healthcare industries. Healthcare industries have large volume databases. Using big data analytics one can study huge datasets and find hidden information, hidden patterns to discover knowledge from the data and predict outcomes accordingly. In existing method, the classification and prediction accuracy is not so high. In this paper, we have proposed a diabetes prediction model for better classification of diabetes which includes few external factors responsible for diabetes along with regular factors like Glucose, BMI, Age, Insulin, etc. Classification accuracy is boosted with new dataset compared to existing dataset. Further with imposed a pipeline model for diabetes prediction intended towards improving the accuracy of classification.

Chapter 1

INTRODUCTION

Healthcare sectors have large volume databases. Such databases may contain structured, semi-structured or unstructured data. Big data analytics is the process which analyses huge data sets and reveals hidden information, hidden patterns to discover knowledge from the given data. Considering the current scenario, in developing countries like India, Diabetic Mellitus (DM) has become a very severe disease. Diabetic Mellitus (DM) is classified as Non-Communicable Disease (NCB) and many people are suffering from it. Around 425 million people suffer from diabetes according to 2017 statistics. Approximately 2-5 million patients every year lose their lives due to diabetes. It is said that by 2045 this will rise to 629 million.Machine learning is considered to be a dire need of today's situation in order to eliminate human efforts by supporting automation with minimum flaws. Existing method for diabetes detection is uses lab tests such as fasting blood glucose and oral glucose tolerance. However, this method is time consuming.

1.1 Literature Review

The analysis of related work gives results on various healthcare datasets, where analysis and predictions were carried out using various methods and techniques. Various prediction models have been developed and implemented by various researchers using variants of data mining techniques, machine learning algorithms or also combination of these techniques. Dr Saravana Kumar N M, Eswari, Sampath P and Lavanya S (2015) implemented a system using Hadoop and Map Reduce technique for analysis of Diabetic data. This system predicts type of diabetes and also risks associated with it. The system is Hadoop based and is economical for any healthcare organization.[4] Aiswarya Iyer (2015) used classification technique to study hidden patterns in diabetes dataset. Na¨ıve Bayes and Decision Trees were used in this model. Comparison was made for performance of both algorithms and effectiveness of both algorithms was shown as a result.[5] K. Rajesh and V. Sangeetha (2012) used classification technique. They used C4.5 decision tree algorithm to find hidden patterns from the dataset for classifying efficiently.[8] Humar Kahramanli and Novruz Allahverdi (2008) used Artificial neural network (ANN) in combination with fuzzy logic to predict diabetes.[9] B.M. Patil, R.C. Joshi and Durga Toshniwal (2010) proposed Hybrid Prediction Model which includes Simple K-means clustering algorithm followed by application of classification algorithm to the result obtained from clustering algorithm. In order to build classifiers C4.5 decision tree algorithm is use.[10] Mani Butwall and Shraddha Kumar (2015) proposed a model using Random Forest Classifier to forecast diabetes behaviour.[7] Nawaz Mohamudally1 and Dost Muhammad (2011) used C4.5 decision tree algorithm, Neural Network, K-means clustering algorithm and Visualization to predict diabetes.[11] Fig 1, represents taxonomy for Machine Learning Algorithms that can be used for diabetes prediction. The task of choosing a machine learning algorithm includes feature matching of the data to be learned based on existing approaches. Taxonomy of machine learning algorithms is discussed belowMachine learning has numerous algorithms which are classified into three categories: Supervised learning, Unsupervised learning, Semi-supervised learning

1.2 Data

We build a spoken expression corpus of 10 million data by crawling comments on political and economics articles. We create an initial abusive word list of 7,450 words by combining Google's bad-word list, a swear- and four-letter-word list, an offensive/profane word list, profanity in the English language, banned words in online game and community site, and other abusive words in the related literature. We then construct a non-abusive word list containing 1.5 million words to reduce positive errors when classifying abusive text.

1.2.1 History

1.2.2 Terminology

The Supervised Learning/Predictive Models

Supervised learning algorithms are used to construct predictive models. A predictive model predicts missing value using other values present in the dataset. Supervised learning algorithm has a set of input data and also a set of output, and builds a model to make realistic predictions for the response to new dataset. Supervised learning includes Decision Tree, Bayesian Method, Artificial Neural Network, Instance based learning, Ensemble Method. These are booming techniques in Machine learning.

Unsupervised Learning / Descriptive Models

Descriptive models are developed using unsupervised learning method. In this model we have known set of inputs but output is unknown. Unsupervised learning is mostly used on transactional data. This method includes clustering algorithms like k-Means clustering and k-Medians clustering.

Semi-supervised Learning

Semi Supervised learning method uses both labeled and unlabeled data on training dataset. Classification, Regression techniques come under Semi Supervised Learning. Logistic Regression, Linear Regression are examples of regression techniques

1.2.3 MOTIVATION

There has been drastic increase in rate of people suffering from diabetes since a decade. Current human lifestyle is the main reason behind growth in diabetes. In current medical diagnosis method, there can be three different types of errors1. The false-negative type in which a patient in reality is already a diabetic patient but test results tell that the person is not having diabetes. 2. The false-positive type. In this type, patient in reality is not a diabetic patient but test reports say that he/she is a diabetic patient. 3. The third type is unclassifiable type in which a system cannot diagnose a given case. This happens due to insufficient knowledge extraction from past data, a given patient may get predicted in an unclassified type. However, in reality, the patient must predict either to be in diabetic category or non-diabetic category. Such errors in diagnosis may lead to unnecessary treatments or no treatments at all when required. In order to avoid or reduce severity of such impact, there is a need to create a system using machine learning algorithm and data mining techniques which will provide accurate results and reduce human efforts.

1.3 Diabetes Prediction Techniques

There are different kinds of techniques are used for diabetes prediction, which are differentiated on the basis of types of document, types of domain *etc* [12]. The various types of watermarking according to different categories are shown in Figure 1.2 [7].

Watermarking techniques are broadly divided into four types:

1. According to working domain
2. According to types of document
3. According to human perception
4. According to application

These four categories are further classified as below

1. According to working domain
   - Spatial domain
   - Frequency domain
2. According to application
   - Source based detection
   - Destination based detection

1.4 Types of Diabetes

In this section we will discuss some of the types of the diabetes prediction.Additional specific testing advice based on risk factors*etc*.

1.4.1 Testing for Type 1 diabetes

Test in children and young adults who have a family history of diabetes. Less commonly, older adults may also develop Type 1 diabetes. Therefore, testing in adults who come to the hospital and are found to be in diabetic ketoacidosis is important. Ketoacidosis a dangerous complication that can occur in people with Type 1 diabetes.Type 1 diabetes is a condition in which your immune system destroys insulin-making cells in your pancreas. These are called beta cells. The condition is usually diagnosed in children and young people, so it used to be called juvenile diabetes.

A condition called secondary diabetes is like type 1, but your beta cells are wiped out by something else, like a disease or an injury to your pancreas, rather than by your immune system.

Both of these are different from type 2 diabetes, in which your body doesn't respond to insulin the way it should.

1.4.2 Testing for type 2 diabetes

Test adults age 45 or older, those between 19 and 44 who are overweight and have one or more risk factors, women who have had gestational diabetes, children between 10 and 18 who are overweight and have at least two risk factors for type 2 diabetes. MENU Diabetes Reference Type 2 Diabetes Medically Reviewed by Michael Dansinger, MD on December 06, 2020 ARTICLES ON TYPE 2 DIABETES OVERVIEW Type 2 Diabetes Type 2 Diabetes Symptoms Glucose Test for Diabetes Type 2 Diabetes Treatment What Is Type 2 Diabetes? Type 2 diabetes is a lifelong disease that keeps your body from using insulin the way it should. People with type 2 diabetes are said to have insulin resistance.

People who are middle-aged or older are most likely to get this kind of diabetes. It used to be called adult-onset diabetes. But type 2 diabetes also affects kids and teens, mainly because of childhood obesity.

Type 2 is the most common type of diabetes. There are about 29 million people in the U.S. with type 2. Another 84 million have prediabetes, meaning their blood sugar (or blood glucose) is high but not high enough to be diabetes yet.

1.4.3 Gestational Diabetes

Test all pregnant women who have had a diagnosis of diabetes. Test all pregnant women between weeks 24 and 28 of their pregnancy. If you have other risk factors for gestational diabetes, your obstetrician may test you earlier. Gestational diabetes is a condition in which your blood sugar levels become high during pregnancy. It affects up to 10% of women who are pregnant in the U.S. each year. It affects pregnant women who haven't ever been diagnosed with diabetes.

There are two classes of gestational diabetes. Women with class A1 can manage it through diet and exercise. Those who have class A2 need to take insulin or other medications.

Gestational diabetes goes away after you give birth. But it can affect your baby's health, and it raises your risk of getting type 2 diabetes later in life. You can take steps so you and your baby stay healthy.

1.4.4 Prediabetes

If you have prediabetes, the goal is to keep you from progressing to diabetes. Treatments are focused on treatable risk factors, such as losing weight by eating a healthy diet (like the Mediterranean diet) and exercising (at least five days a week for 30 minutes). Many of the strategies used to prevent diabetes are the same as those recommended to treat diabetes

1.5 Organization

This dissertation is organized as follows:

- The introduction to the dissertation is given in Chapter 1. This section describes definition, essentiality, usage and requirements of diabetes prediction concept.
- Chapter 2 focuses on research contribution of various authors.
- Chapter 3 focuses on introduction and implementation details of improved technique.

1.6 Diabetes Overview

Diabetes mellitus refers to a group of diseases that affect how your body uses blood sugar (glucose). Glucose is vital to your health because it's an important source of energy for the cells that make up your muscles and tissues. It's also your brain's main source of fuel.

The underlying cause of diabetes varies by type. But, no matter what type of diabetes you have, it can lead to excess sugar in your blood. Too much sugar in your blood can lead to serious health problems.

Chronic diabetes conditions include type 1 diabetes and type 2 diabetes. Potentially reversible diabetes conditions include prediabetes and gestational diabetes. Prediabetes occurs when your blood sugar levels are higher than normal, but not high enough to be classified as diabetes. And prediabetes is often the precursor of diabetes unless appropriate measures are taken to prevent progression. Gestational diabetes occurs during pregnancy but may resolve after the baby is delivered.

1.7 Relavance Of The Project

Diabetes mellitus refers to a group of diseases that affect how your body uses blood sugar (glucose). Glucose is vital to your health because it's an important source of energy for the cells that make up your muscles and tissues. It's also your brain's main source of fuel.

The underlying cause of diabetes varies by type. But, no matter what type of diabetes you have, it can lead to excess sugar in your blood. Too much sugar in your blood can lead to serious health problems.

Chronic diabetes conditions include type 1 diabetes and type 2 diabetes. Potentially reversible diabetes conditions include prediabetes and gestational diabetes. Prediabetes occurs when your blood sugar levels are higher than normal, but not high enough to be classified as diabetes. And prediabetes is often the precursor of diabetes unless appropriate measures are taken to prevent progression. Gestational diabetes occurs during pregnancy but may resolve after the baby is delivered.

1.8 Purpose

Large amount of data has been continuously generated in field of engineering and science. Recent advances in technology have resulted in big electronic data that allow data to be captured, processed, analyzed and stored rather inexpensively. This change leads to new trends in market as well as industry such as Internet banking and e-commerce, insurance, financial transactions, supermarket, healthcare, communications, location of data that generate huge amount of electronic data. The need to understand huge, complex, information rich data sets is important to virtually all fields in business, science, engineering and medical. The data used in data warehouses and data marts has been extracted from knowledge hidden in that data. This knowledge is becoming vital in today's increasingly competitive world. The greatest problem of today is how to teach people to ignore the irrelevant data. With the rise of Machine Learning approaches we have the ability to find a solution diabetes prediction, we have developed a system using machine learning which has the ability to predict whether the patient has diabetes or not. Furthermore, predicting the disease early leads to treating the patients before it becomes critical. Machine Learning has the ability to extract hidden knowledge from a huge amount of diabetes-related data. Because of that, it has a significant role in diabetes research, now more than ever. The aim of this research is to develop a system which can predict the diabetic risk level of a patient with a higher accuracy. This research has focused on developing a system based on logistic regression.

1.9 Scope Of The Project

The early intervention of diabetes can reduce the prevalence of diabetes and hence the economic burden due to it. Machine Learning techniques play an important role in treatment plan workout, rehabilitation, chronic diseases management plan etc. Long term follow up plan may be easily guided and keen supervision is possible. The systems may definitely helpful in reduction of cost of patient management by avoiding unnecessary investigations and patients follow up. These prediction systems will add accuracy and time management. Computer-based patient support systems benefit patients by providing informational support that increases their participation in health care.

1.10 Problem Statement And Definition

To identify whether a given person in dataset will be diabetic, non diabetic or pre- diabetic will be done on basis of attribute values. Values exceeding a specific value may contribute to identify whether a person is diabetic, non diabetic or pre-diabetic. The aim of prediction of diabetes is to make aware people about diabetes and what it takes to treat it and gives the power to control. The model can be used by the endocrinologists, dietitians, ophthalmologists and podiatrists to predict if or if not the patient is likely to suffer from diabetes, if yes, how intense it could be. The dataset consists of features comprising the medical details of the

patients that are useful in determining the health condition of the patient.
Goals of the System:-

1. Convert manual to computerize. Before this, majority of the process is done manually. After converting it to computerize it will be easy to predict.
2. Bvy computerizing, it is easier to understand the doctor's report which is hardly understood with different and complicated handwriting.
3. Easy to maintain record.
4. Enable to predict various type of diabetes.
5. Ensure the system useful to doctor and patient.

Chapter 2

REQUIREMENT ANALYSIS

Regarding our project work the current problem is related to Doctors and Patients of Diabetes. If any of the grievances is there, then solving all those problems of Doctors and Patients is not an easy job because at that time we should consider the maximum workload and we needs another platform for keeping that track of record so that Doctor can identify the type of Diabetes in minimum time span and the problem will be solved as early as possible. Requirement analysis, also called as requirement engineering, is the process of determining user expectations for a new or modified product and provide better service experience. Requirement analysis is critical to the success of system or software project. The requirement should be docu- mented,actionable, measurable, testable, traceable, related to the identified business or opportunities and depend to a level of detail sufficient for system design.

2.1 Support Vector Machine

Support Vector Machine also known as svm is a supervised machine learning algorithm. Svm is most popular classification technique. Svm creates a hyperplane that separate two classes. It can create a hyperplane or set of hyperplane in high dimensional space. This hyper plane can be used for classification or regression also. Svm differentiates instances in specific classes and can also classify the entities which are not supported by data. Separation is done by through hyperplane performs the separation to the closest training point of any class.

2.1.1 Algorithm

- Select the hyper plane which divides the class better.
- To find the better hyper plane you have to calculate the distance between the planes and the data which is called Margin.
- If the distance between the classes is low then the chance of miss conception is high and vice versa. So we need to
- Select the class which has the high margin. Margin = distance to positive point + Distance to negative point

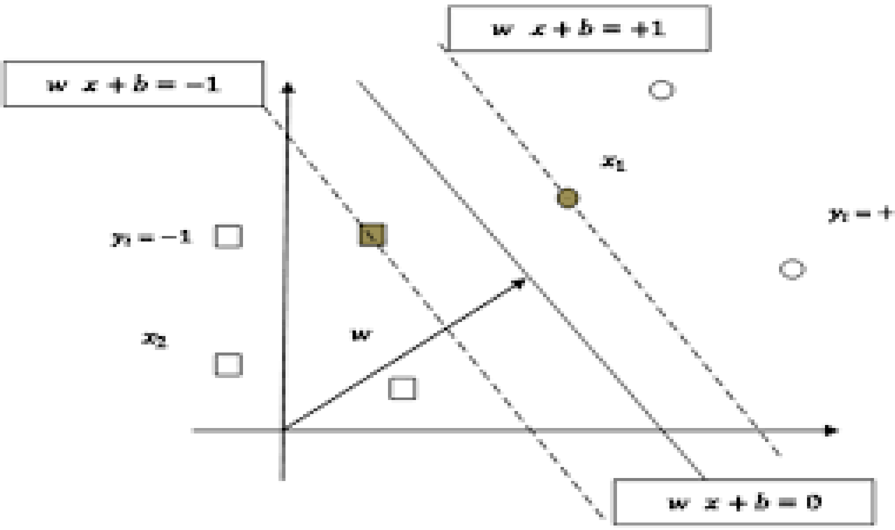


Figure 2.1: Support Vector Machine

2.2 K-Nearest Neighbor

KNN is also a supervised machine learning algorithm. KNN helps to solve both the classification and regression problems. KNN is lazy prediction technique.KNN assumes that similar things are near to each other. Many times data points which are similar are very near to each other.KNN helps to group new work based on similarity measure.KNN algorithm record all the records and classify them according to their similarity measure. For finding the distance between the points uses tree like structure. To make a prediction for a new data point, the algorithm finds the closest data points in the training data set — it's nearest neighbors. Here K= Number of nearby neighbors, it's always a positive integer. Neighbor's value is chosen from set of class. Closeness is mainly defined in terms of Euclidean distance. The Euclidean distance between two points P and Q i.e. P (p1,p2, ... Pn)

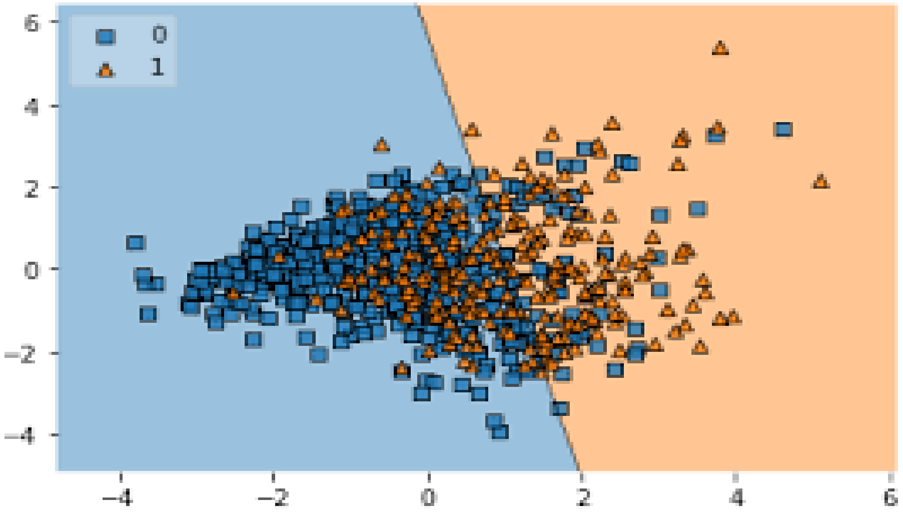


Figure 2.2: KNN

2.2.1 Algorithm

- Take a sample dataset of columns and rows named as Pima Indian Diabetes data set.
- Take a test dataset of attributes and rows.
- Find the Euclidean distance by the help of formula-
- Then, Decide a random value of K. is the no. of nearest neighbors
- Then with the help of these minimum distance and Euclidean distance find out the nth column of each.
- Find out the same output values

2.3 Decision Tree

Decision tree is a basic classification method. It is supervised learning method. Decision tree used when response variable is categorical. Decision tree has tree like structure based model which describes classification process based on input feature. Input variables are any types like graph, text, discrete, continuous etc. Steps for Decision Tree

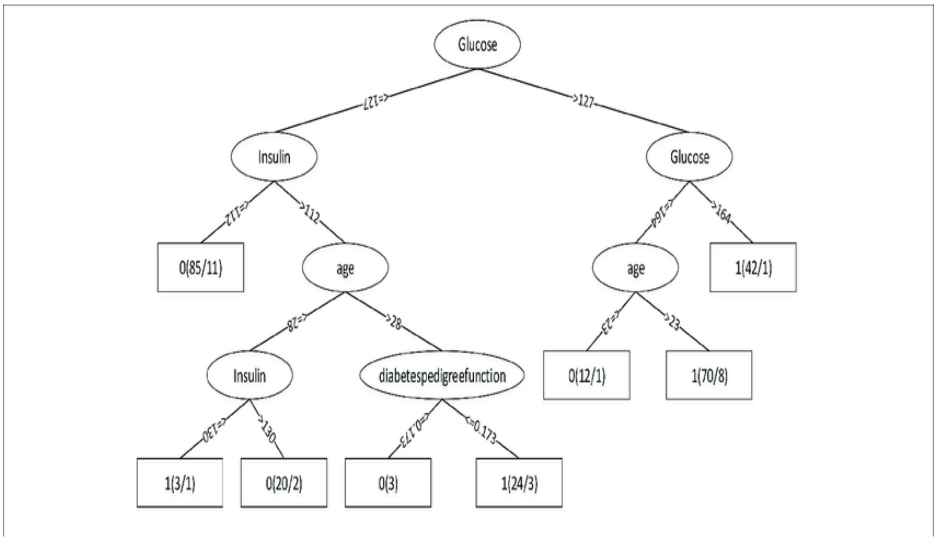


Figure 2.3: Decision tree

2.3.1 Algorithm

- Construct tree with nodes as input feature.
- Select feature to predict the output from input feature whose information gain is highest.
- The highest information gain is calculated for each attribute in each node of tree.
- Repeat step 2 to form a subtree using the feature which is not used in above node.

2.4 Logistic Regression

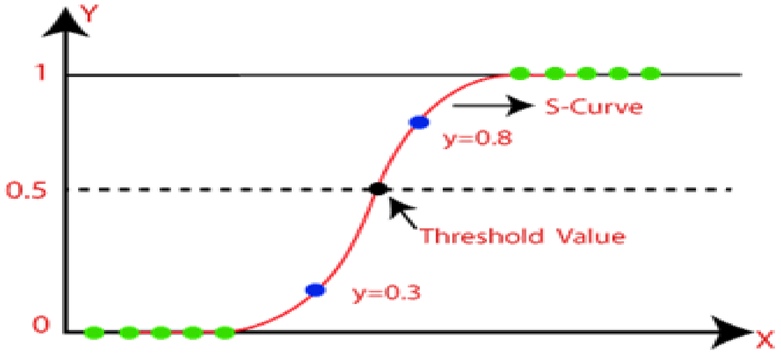


Fig 1:- Logistic Regression

Source: Javapoint

Figure 2.4: Logistic Regresssion

Logistic regression is also a supervised learning classification algorithm. It is used to estimate the probability of a binary response based on one or more predictors. They can be continuous or discrete. Logistic regression used when we want to classify or distinguish some data items into categories. It classify the data in binary form means only in 0 and 1 which refer case to classify patient that is positive or negative for diabetes. Main aim of logistic regression is to best fit which is responsible for describing the relationship between target and predictor variable. Logistic regression is a based on Linear regression model. Logistic regression model uses sigmoid function to predict probability of positive and negative class.Sigmoid function P = 1/1+e - (a+bx) Here P = probability, a and b = parameter of Model

2.5 Random Forest

– It is type of ensemble learning method and also used for classification and regression tasks. The accuracy it gives is grater then compared to other models. This method can easily handle large datasets. Random Forest is developed by Leo Bremen. It is popular ensemble Learning Method. Random Forest Improve Performance of Decision Tree by reducing variance. It operates by constructing a multitude of decision trees at training time and outputs the class that is the mode of the classes or classification or mean prediction (regression) of the individual trees.

2.5.1 Algorithm

- The first step is to select the “R” features from the total features “m” where R¡¡M.
- Among the “R” features, the node using the best split point.
- Split the node into sub nodes using the best split.

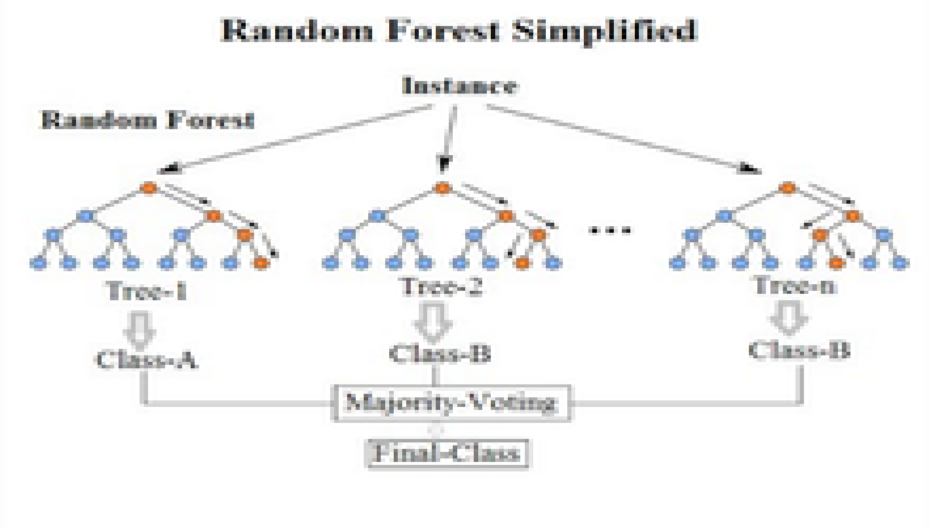


Figure 2.5: Random Forest

- Repeat a to c steps until ”l” number of nodes has been reached.
- Built forest by repeating steps a to d for “a” number of times to create “n” number of trees. The random forest finds the best split using the Gin-Index Cost Function

Chapter 3

PLANNING AND SCHEDULING

The Planning and Scheduling for the Diabetes Management System presented below in Figure 1 is the conceptual model that defines the structure, behavioural interactions, and multiple system views that underpins the system development. It presents the formal descriptions of the systems captured graphically that supports reasoning, and the submodules developed as well as the dataflows between the developed modules.

3.1 Introduction to Proposed System

In this work different steps were taken. The proposed approach uses different classification and ensemble methods and implemented using python. These methods are standard Machine Learning methods used to obtain the best accuracy from data. In this work we see that random forest classifier achieves better compared to others. Overall we have used best Machine Learning techniques for prediction and to achieve high performance accuracy.

3.2 Objectives

The aim of this project is to develop a system which can perform early prediction of diabetes for a patient with a higher accuracy by combining the results of different machine learning techniques

- To implement data processing algorithm which will predict the output.
- To implement the process of execution.
- To implement the html page which will display final output.

3.2.1 Proposed Methodology

Diabetes prediction is an internet-primarily based device gaining knowledge of utility, skilled through a Pima Indian dataset. The person inputs its particular clinical information to get the prediction of diabetes. The set of rules will calculate the opportunity of presence of diabetes. Thus, minimizing the price and time required to are expecting the disorder. Format of statistics plays essential element on this software. At the time of uploading the user information utility will take a look at its right record format and if it no longer as consistent with want then ERROR dialog box may be induced. Our device might be implementing the algorithm: Logistic Regression. The algorithms may be educated the use of the statistics set obtained from University of California, Irvine. 75Goal of the paper is to investigate for model to predict dia- betes with better accuracy. We experimented with different classification and ensemble algorithms to predict diabetes. In the following, we briefly discuss the phase.

Advantages Of Systems

1. Powerful, flexible, and easy to use.
2. Increased efficiency of doctor.
3. Improved patient satisfaction.
4. Reduce the use of papers.
5. Simple and Quick.
6. More accurate result.

Table 1: Dataset Description:

1. Pregnancy
2. Glucose
3. Blood Pressure
4. Skin thickness
5. Insulin
6. BMI(Body Mass Index)
7. Diabetes Pedigree Function
8. Age

   The 9th attribute is class variable of each data points. This class variable shows the outcome 0 and 1 for diabetics which indicates positive or negative for diabetics.

   Distribution of Diabetic patient- We made a model to predict diabetes however the dataset was slightly imbal- anced having around 500 classes labeled as 0 means nega- tive means no diabetes and 268 labeled as 1 means positive means diabetic.

1. **Data Preprocessing:**
   Data preprocessing is a process of preparing the raw data and making it suitable for a machine learning model. It is the first and crucial step while creating a machine learning model.

   When creating a machine learning project, it is not always a case that we come across the clean and formatted data. And while doing any operation with data, it is mandatory to clean it and put in a formatted way. So for this, we use data preprocessing task.

3.2.2 Steps Involved In Data Preprocesing

1. Getting the dataset
2. Importing libraries
3. Importing datasets
4. Finding Missing Data
5. Encoding Categorical Data
6. Splitting dataset into training and test set
7. Feature scaling

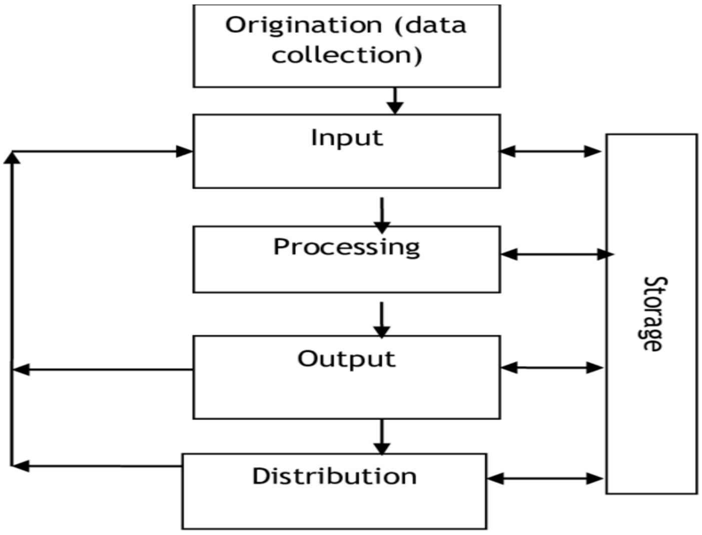


Figure 3.1: Data Preprocessing

1. **Get the Dataset:** To create a machine learning model, the first thing we required is a dataset as a machine learning model completely works on data. The collected data for a particular problem in a proper format is known as the dataset.

   Dataset may be of different formats for different purposes, such as, if we want to create a machine learning model for business purpose, then dataset will be different with the dataset required for a liver patient. So each dataset is different from another dataset. To use the dataset in our code, we usually put it into a CSV file. However, sometimes, we may also need to use an HTML or xlsx file.

2. **Importing Libraries:**In order to perform data preprocessing using Python, we need to import some predefined Python libraries. These libraries are used to perform some specific jobs. There are three specific libraries that we will use for data preprocessing, which are:

Numpy: Numpy Python library is used for including any type of mathematical operation in the code. It is the fundamental package for scientific calculation in Python. It also supports to add large, multidimensional arrays and matrices. So, in Python, we can import it as:

import numpy as nm :

Matplotlib: The second library is matplotlib, which is a Python 2D plotting library, and with this library, we need to import a sub-library pyplot. This library is used to plot any type of charts in Python for the code. It will be imported as below:

import matplotlib.pyplot as mpt

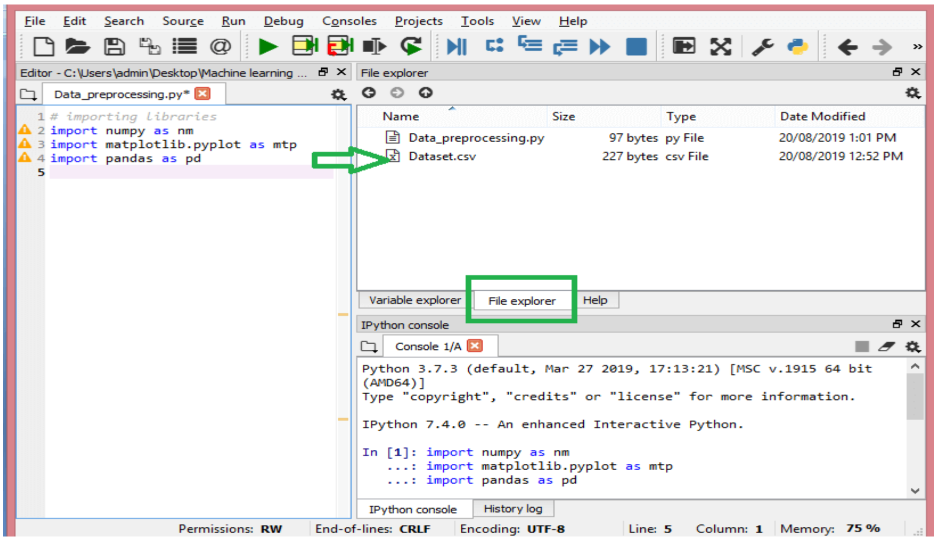
Here we have used mpt as a short name for this library.

Pandas: The last library is the Pandas library, which is one of the most famous Python libraries and used for importing and managing the datasets. It is an open-source data manipulation and analysis library. It will be imported as below:

Here, we have used pd as a short name for this library. Consider the below image:

3. **Importing the Datasets:**Now we need to import the datasets which we have collected for our machine learning project. But before importing a dataset, we need to set the current directory as a working directory. To set a working directory in Spyder IDE, we need to follow the below steps:

   - Save your Python file in the directory which contains dataset.
   - Go to File explorer option in Spyder IDE, and select the required directory.
   - Click on F5 button or run option to execute the file.



```
# importing libraries
import numpy as nm
import matplotlib.pyplot as mtp
import pandas as pd
```

```
Name                     Size       Type      Date Modified
Data_preprocessing.py    97 bytes   py File   20/08/2019 1:01 PM
Dataset.csv              227 bytes  csv File  20/08/2019 12:52 PM
```

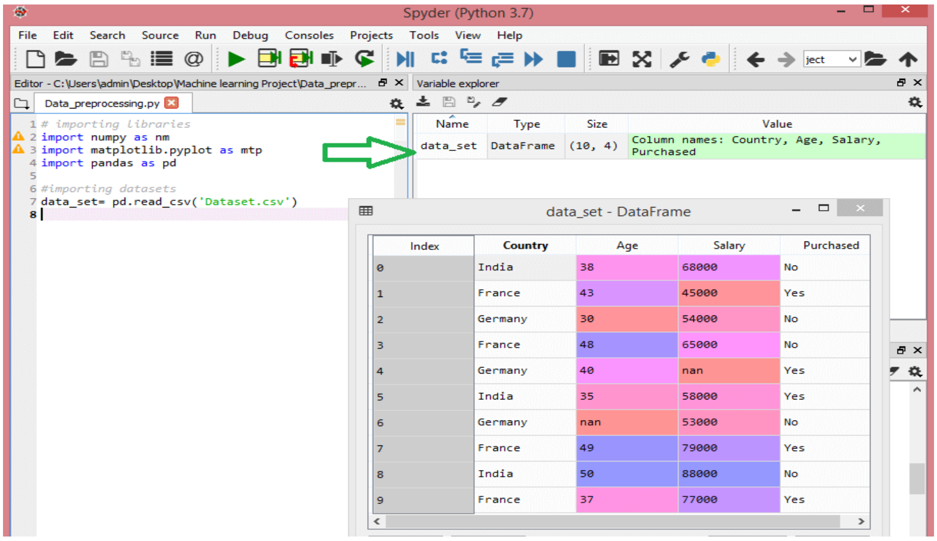
```
Python 3.7.3 (default, Mar 27 2019, 17:13:21) [MSC v.1915 64 bit
(AMD64)]
Type "copyright", "credits" or "license" for more information.

IPython 7.4.0 -- An enhanced Interactive Python.

In [1]: import numpy as nm
   ...: import matplotlib.pyplot as mtp
   ...: import pandas as pd
```

```
Permissions: RW   End-of-lines: CRLF   Encoding: UTF-8   Line: 5   Column: 1   Memory: 75 %
```

Figure 3.2: Import Dataset

Here, in the below image, we can see the Python file along with required dataset. Now, the current folder is set as a working directory.

readcsv() function: Now to import the dataset, we will use readcsv() function of pandas library, which is used to read a csv file and performs various operations on it. Using this function, we can read a csv file locally as well as through an URL. We can use readcsv function as below:Here, dataset is a name of the variable to store our dataset, and inside the function, we have passed the name of our dataset. Once we execute the above line of code, it will successfully import the dataset in our code. We can also check the imported dataset by clicking on the section variable explorer, and then double click on dataset. Consider the below image:



```
# importing libraries
import numpy as nm
import matplotlib.pyplot as mtp
import pandas as pd

#importing datasets
data_set= pd.read_csv('Dataset.csv')
```

Name	Type	Size	Value
data_set	DataFrame	(10, 4)	Column names: Country, Age, Salary, Purchased

Index	Country	Age	Salary	Purchased
0	India	38	68000	No
1	France	43	45000	Yes
2	Germany	30	54000	No
3	France	48	65000	No
4	Germany	40	nan	Yes
5	India	35	58000	Yes
6	Germany	nan	53000	No
7	France	49	79000	Yes
8	India	50	88000	No
9	France	37	77000	Yes

Figure 3.3: Extracting Dependent And Independent Vasriables

Extracting dependent and independent variables: In machine learning, it is important to distinguish the matrix of features (independent variables) and dependent variables from dataset. In our dataset, there are three independent variables that are Country, Age, and Salary, and one is a dependent variable which is Purchased.

x= data$_set.iloc[:, : -1].values$
$Intheabovecode, thefirstcolon(:)$
$isusedtotakealltherows, andthesecondcolon(:)isforallthecolumns.$
$Herewehaveused : -1, becausewedon'twanttotakethelastcolumn$
$asitcontainsthedependentvariable.Sobydoingthis, wewillgetthematrixoffeatures.$

By executing the above code, we will get output as:

[['India' 38.0 68000.0] ['France' 43.0 45000.0] ['Germany' 30.0 54000.0] ['France' 48.0 65000.0] ['Germany' 40.0 nan] ['India' 35.0 58000.0] ['Germany' nan 53000.0] ['France' 49.0 79000.0] ['India' 50.0 88000.0] ['France' 37.0 77000.0]] As we can see in the above output, there are only three variables.

Extracting dependent variable:

To extract dependent variables, again, we will use Pandas .iloc[] method.

y= data$_set.iloc[:, 3].values$

Here we have taken all the rows with the last column only. It will give the array of dependent variables.

By executing the above code, we will get output as:

Output:

array(['No', 'Yes', 'No', 'No', 'Yes', 'Yes', 'No', 'Yes', 'No', 'Yes'], dtype=object) Note: If you are using Python language for machine learning, then extraction is mandatory, but for R language it is not required.

4) Handling Missing data: The next step of data preprocessing is to handle missing data in the datasets. If our dataset contains some missing data, then it may create a huge problem for our machine learning model. Hence it is necessary to handle missing values present in the dataset.

Ways to handle missing data:

There are mainly two ways to handle missing data, which are:

By deleting the particular row: The first way is used to commonly deal with null values. In this way, we just delete the specific row or column which consists of null values. But this way is not so efficient and removing data may lead to loss of information which will not give the accurate output.

By calculating the mean: In this way, we will calculate the mean of that column or row which contains any missing value and will put it on the place of missing value. This strategy is useful for the features which have numeric data such as age, salary, year, etc. Here, we will use this approach.

To handle missing values, we will use Scikit-learn library in our code, which contains various libraries for building machine learning models. Here we will use Imputer class of sklearn.preprocessing library. Below is the code for it:

handling missing data (Replacing missing data with the mean value)

from sklearn.preprocessing import Imputer imputer$\bar{I}$mputer(missing$_v alues =' NaN', strategy\bar{'}mean', axis =$

5) Encoding Categorical data: Categorical data is data which has some categories such as, in our dataset; there are two categorical variable, Country, and Purchased.

Since machine learning model completely works on mathematics and numbers, but if our dataset would have a categorical variable, then it may create trouble while building the model. So it is necessary to encode these categorical variables into numbers.

For Country variable:

Firstly, we will convert the country variables into categorical data. So to do this, we will use LabelEncoder() class from preprocessing library.

Catgorical data for Country Variable from sklearn.preprocessing import
LabelEncoder label$_encoder_x = LabelEncoder()x[:,0] = label_encoder\_x.fit_transform(x[:,0])$
$Output:$

Out[15]: array([[2, 38.0, 68000.0], [0, 43.0, 45000.0], [1, 30.0, 54000.0], [0, 48.0, 65000.0], [1, 40.0, 65222.22222222222], [2, 35.0, 58000.0],
41.111111111111114, 53000.0
, [0, 49.0, 79000.0], [2, 50.0, 88000.0], [0, 37.0, 77000.0]], dtype=object) Explanation:

In above code, we have imported LabelEncoder class of sklearn library. This class has successfully encoded the variables into digits.

But in our case, there are three country variables, and as we can see in the above output, these variables are encoded into 0, 1, and 2. By these values, the machine learning model may assume that there is some correlation between these variables which will produce the wrong output. So to remove this issue, we will use dummy encoding.

Dummy Variables:

Dummy variables are those variables which have values 0 or 1. The 1 value gives the presence of that variable in a particular column, and rest variables become 0. With dummy encoding, we will have a number of columns equal to the number of categories.

In our dataset, we have 3 categories so it will produce three columns having 0 and 1 values. For Dummy Encoding, we will use OneHotEncoder class of preprocessing library.

It can be seen more clearly in the variables explorer section, by clicking on x option as:

Data Preprocessing in Machine learning

6) Splitting the Dataset into the Training set and Test set: In machine learning data preprocessing, we divide our dataset into a training set and test set. This is one of the crucial steps of data preprocessing as by doing this, we can enhance the performance of our machine learning model.

Suppose, if we have given training to our machine learning model by a dataset and we test it by a completely different dataset. Then, it will create difficulties for our model to understand the correlations between the models.

If we train our model very well and its training accuracy is also very high, but we provide a new dataset to it, then it will decrease the performance. So we always try to make a machine learning model which performs well with the training set and also with the test dataset. Here, we can define these datasets as:

Data Preprocessing in Machine learning Training Set: A subset of dataset to train the machine learning model, and we already know the output.

Test set: A subset of dataset to test the machine learning model, and by using the test set, model predicts the output.

For splitting the dataset, we will use the below lines of code:

from sklearn.model$_{s}electionimporttrain_{t}est_{s}plit$
$x\_train, x\_test, y\_train, y\_test = train_{t}est_{s}plit(x, y, test_{s}ize = 0.2, random_{s}tate = 0) Explanation :$

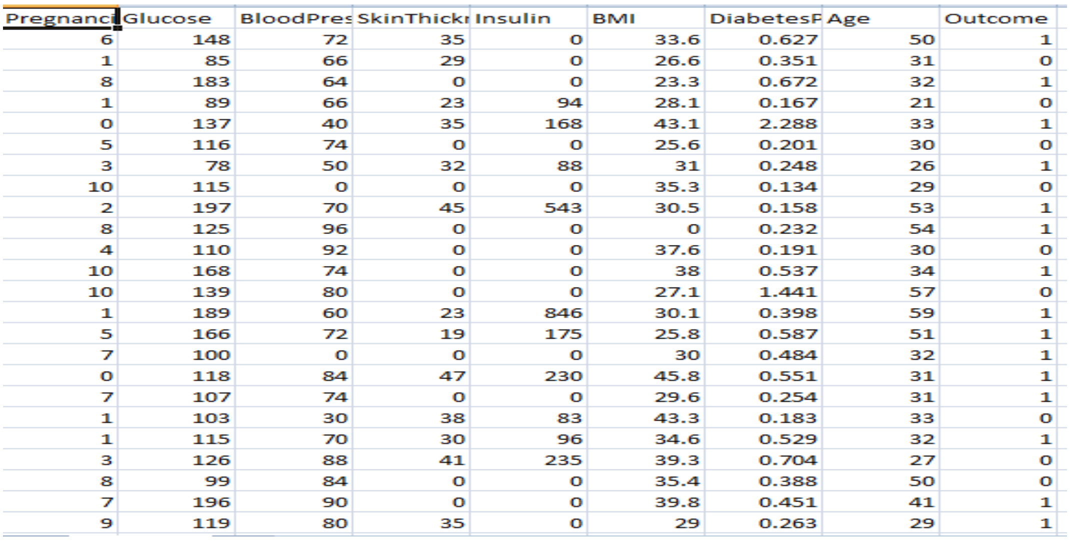
In the above code, the first line is used for splitting arrays of the dataset into random train and test subsets. In the second line, we have used four variables for our output that are x_train: features for the training data
x_test: features for testing data
y_train: Dependent variables for training data
y_test: Independent variable for testing data

By executing the above code, we will get 4 different variables, which can be seen under the variable explorer section.

Data Preprocessing in Machine learning As we can see in the above image, the x and y variables are divided into 4 different variables with corresponding values.

7) Feature Scaling: Feature scaling is the final step of data preprocessing in machine learning. It is a technique to standardize the independent variables of the dataset in a specific range. In feature scaling, we put our variables in the same range and in the same scale so that no any variable dominate the other variable.

Consider the below dataset: Data Preprocessing in Machine learning As we can see, the age and salary column values are not on the same scale. A machine learning model is based on Euclidean distance, and if we do not scale the variable, then it will cause some issue in our machine learning model.



Pregnanci	Glucose	BloodPres	SkinThickr	Insulin	BMI	DiabetesF	Age	Outcome
6	148	72	35	0	33.6	0.627	50	1
1	85	66	29	0	26.6	0.351	31	0
8	183	64	0	0	23.3	0.672	32	1
1	89	66	23	94	28.1	0.167	21	0
0	137	40	35	168	43.1	2.288	33	1
5	116	74	0	0	25.6	0.201	30	0
3	78	50	32	88	31	0.248	26	1
10	115	0	0	0	35.3	0.134	29	0
2	197	70	45	543	30.5	0.158	53	1
8	125	96	0	0	0	0.232	54	1
4	110	92	0	0	37.6	0.191	30	0
10	168	74	0	0	38	0.537	34	1
10	139	80	0	0	27.1	1.441	57	0
1	189	60	23	846	30.1	0.398	59	1
5	166	72	19	175	25.8	0.587	51	1
7	100	0	0	0	30	0.484	32	1
0	118	84	47	230	45.8	0.551	31	1
7	107	74	0	0	29.6	0.254	31	1
1	103	30	38	83	43.3	0.183	33	0
1	115	70	30	96	34.6	0.529	32	1
3	126	88	41	235	39.3	0.704	27	0
8	99	84	0	0	35.4	0.388	50	0
7	196	90	0	0	39.8	0.451	41	1
9	119	80	35	0	29	0.263	29	1

Figure 3.4: Dataset

Euclidean distance is given as:

Data Preprocessing in Machine learning If we compute any two values from age and salary, then salary values will dominate the age values, and it will produce an incorrect result. So to remove this issue, we need to perform feature scaling for machine learning.

textbfThere are two ways to perform feature scaling in machine learning:

Standardization

Data Preprocessing in Machine learning Normalization

Data Preprocessing in Machine learning Here, we will use the standardization method for our dataset.

For feature scaling, we will import StandardScaler class of sklearn.preprocessing library as:

from sklearn.preprocessing import StandardScaler Now, we will create the object of StandardScaler class for independent variables or features. And then we will fit and transform the training dataset.

x_test= st_x.transform(x_test) Output:

By executing the above lines of code, we will get the scaled values for x$_t rainandx_t estas$:

x_train:

Data Preprocessing in Machine learning x_test:

Data Preprocessing in Machine learning As we can see in the above output, all the variables are scaled between values -1 to 1.

Note: Here, we have not scaled the dependent variable because there are only two values 0 and 1. But if these variables will have more range of values, then we will also need to scale those variables. Combining all the steps:

Now, in the end, we can combine all the steps together to make our complete code more understandable.

importing libraries import numpy as nm import matplotlib.pyplot as mtp import pandas as pd

importing datasets data$_set = pd.read_csv('Dataset.csv')$

Extracting Independent Variable x= data$_set.iloc[:, : -1].values$

Extracting Dependent variable y= data$_set.iloc[:, 3].values$

handling missing data(Replacing missing data with the mean value) from sklearn.preprocessing import Imputer imputer= Imputer(missing$_values =' NaN', strategy =' mean', axis = 0)$

Fitting imputer object to the independent varibles x. imputerimputer= imputer.fit(x[:, 1:3])

Replacing missing data with the calculated mean value x[:, 1:3]= imputer.transform(x[:, 1:3])

for Country Variable from sklearn.preprocessing import LabelEncoder, OneHotEncoder label$_encoder_x = LabelEncoder()x[:, 0] = label_encoder_x.fit_transform(x[:, 0])$

Encoding for dummy variables onehot$_encoder = OneHotEncoder(categorical_features = [0])x = onehot_encoder.fit_transform(x).toarray()$

encoding for purchased variable labelencoder$_y = LabelEncoder()y = labelencoder_y.fit_transform(y)$

In the above code, we have included all the data preprocessing steps together. But there are some steps or lines of code which are not necessary for all machine learning models. So we can exclude them from our code to make it reusable for all models.

3.2.3 Logistic Regression in Machine Learning

(a) Logistic regression is one of the most popular Machine Learning algorithms, which comes under the Supervised Learning technique. It is used for predicting the categorical dependent variable using a given set of independent variables.

(b) Logistic regression predicts the output of a categorical dependent variable. Therefore the outcome must be a categorical or discrete value. It can be either Yes or No, 0 or 1, true or False, etc. but instead of giving the exact value as 0 and 1, it gives the probabilistic values which lie between 0 and 1.

(c) Logistic Regression is much similar to the Linear Regression except that how they are used. Linear Regression is used for solving Regression problems, whereas Logistic regression is used for solving the classification problems.

(d) In Logistic regression, instead of fitting a regression line, we fit an ”S” shaped logistic function, which predicts two maximum values (0 or 1).

(e) The curve from the logistic function indicates the likelihood of something such as whether the cells are cancerous or not, a mouse is obese or not based on its weight, etc.

(f) Logistic Regression is a significant machine learning algorithm because it has the ability to provide probabilities and classify new data using continuous and discrete datasets.

(g) Logistic Regression can be used to classify the observations using different types of data and can easily determine the most effective variables used for the classification. The below image is showing the logistic function:

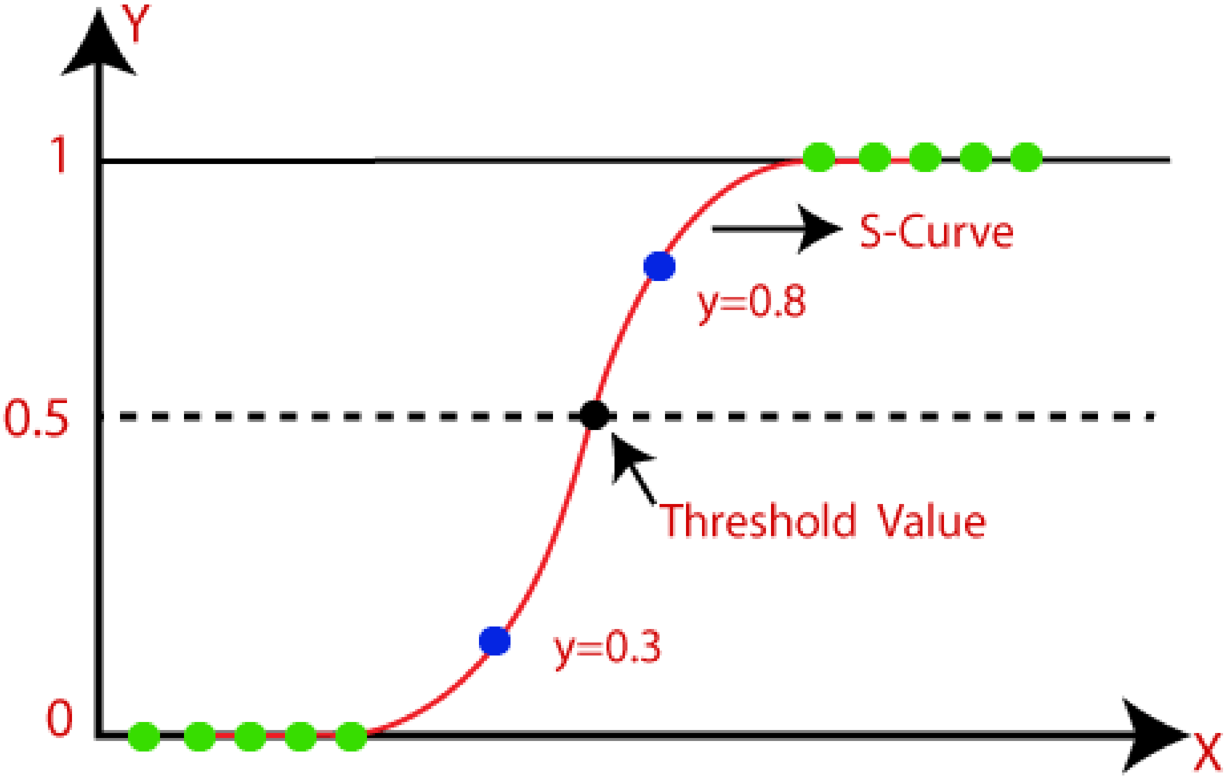
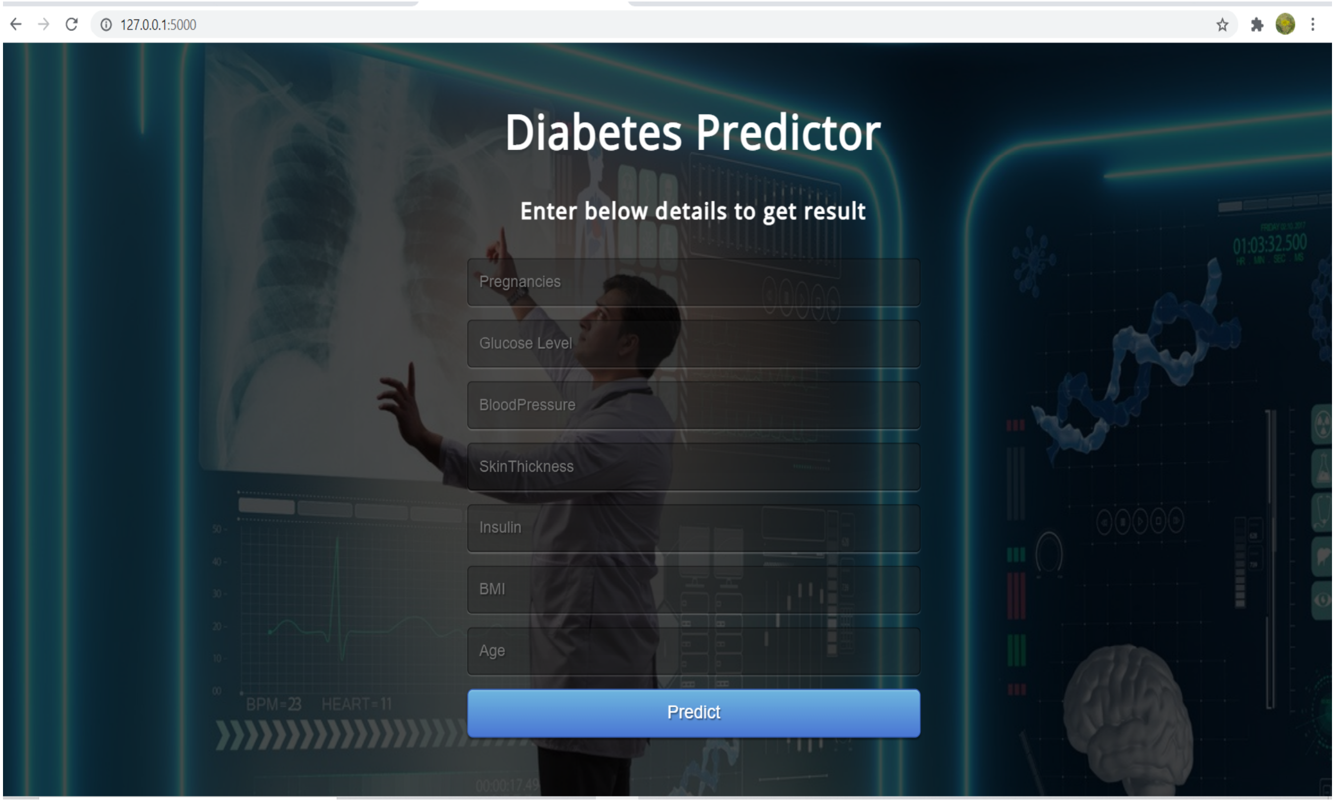


Figure 3.5: curve.jpg

3.3 GUI Snapshots



Diabetes Predictor

Enter below details to get result

Pregnancies

Glucose Level

BloodPressure

SkinThickness

Insulin

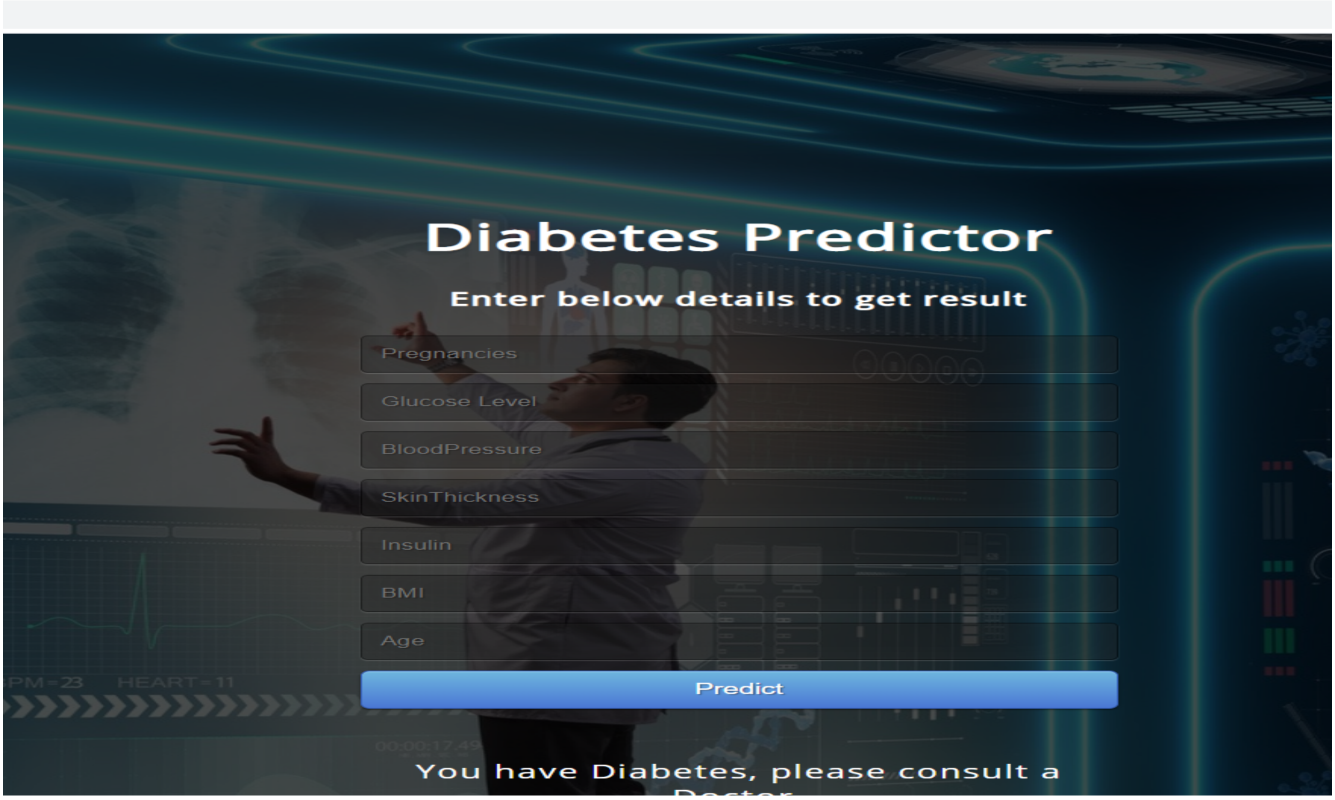
BMI

Age

Predict

Figure 3.6: Attributes

Figure 3.5 shows GUI for the project. The GUI is designed to enter the image and get the watermarked image. The GUI provides facility to calculate the tamper detection rate and PSNR as quality parameter.We are having different buttons for inserting image I, calculating the watermarked image, generating the tamper in the image. We have written separate code for tamper generation of the image for getting the exact number of tampered pixels,button for generating the recovered image. We have added the buttons to calculate the Pregnancy,Glucose,Blood Pressure,Skin thickness,Insulin,BMI(Body Mass Index) ,Diabetes Pedigree Function,Age



You have Diabetes, please consult a Doctor

Figure 3.7: Result

3.4 Overview of Python and Requirements

3.4.1 Tools and Technologies Used

This section will focus on the tools and technology used in the system

3.4.2 Hardware and Software Requirements

Implementation required for this software and hardware on the development side system. The discussed system has been implemented using Jupiter notebook

(a) Hardware Recommended
  i. 2.0 GHz Processor required (Pentium IV or above)
  ii. Minimum 512 MB RAM
  iii. Minimum 25 GB HDD

(b) Software Required
  i. Visual Studio Code
  ii. Jupyter Notebook
  iii. Flask

Chapter 4

LANGAUGE SPECIFICATION

4.1 Python

Python is an easy to learn, powerful programming language. It has efficient high- level data structures and a simple but effective approach to object-oriented programming. Python's elegant syntax and dynamic typing, together with its interpreted nature, make it an ideal language for scripting and rapid application development in many areas on most platforms. The Python interpreter and the extensive standard library are freely available in source or binary form for all major platforms from the Python Web site, https://www.python.org/, and may be freely distributed. The same site also contains distributions of and pointers to many free third party Python modules, programs and tools, and additional documentation. The Python interpreter is easily extended with new functions and data types implemented in C or C++ (or other languages callable from C). Python is also suitable as an extension language for customizable applications. Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently where as other languages use punctuation, and it has fewer syntactical constructions than other languages.

- Python is Interpreted Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.
- Python is Interactive you can actually sit at a Python prompt and interact with the interpreter directly to write your programs.
- Python is Object-Oriented Python supports Object-Oriented style or technique of programming that encapsulates code within objects. Python is a Beginner's Language Python is a great language for the beginner-level programmers and supports the development of a wide range of applications from simple text processing to WWW browsers to games.

4.1.1 Machine Learning Features

Machine Learning is a method of statistical learning where each instance in a dataset is described by a set of features or attributes. In contrast, the term "Deep Learning" is a method of statistical learning that extracts features or attributes from raw data. Deep Learning does this by utilizing neural networks with many hidden layers, big data, and powerful computational resources. The terms seem somewhat interchangeable, however, with Deep Learning method, The algorithm constructs representations of the data automatically. In contrast, data representations are hard-coded as a set of features in machine learning algorithms, requiring further processes such as feature selection and extraction, (such as PCA). Both of these terms are in dramatic contrast with another class of classical artificial intelligence algorithms known as Rule-Based Systems where each decision is manually programmed in such a way that it resembles a statistical model. In Machine Learning and Deep Learning, there are many different models that fall into two different categories, supervised and unsupervised. In unsupervised learning, algorithms such as k-Means, hierarchical clustering, and Gaussian mixture models attempt to learn meaningful structures in the data. Supervised learning involves an output label associated with each instance in the dataset. This output can be discrete/categorical or real-valued. Regression models estimate real-valued outputs, whereas classification models estimate discrete-valued outputs. Simple binary classification models have just two output labels, 1 (positive) and 0 (negative). Some popular supervised learning algorithms that are considered Machine Learning: are linear regression, logistic regression, decision trees, support vector machines, and neural networks, as well as non-parametric models such as k-Nearest Neighbors.

Trauma Injury Severity Score Severity Score Trauma Injury Severity Score, which is widely used to predict mortality in injured patients, was originally developed by Boyd et al. using logistic regression. Many other medical scales used to assess severity of a patient have been developed using logistic regression.

Binary logistic regression is estimated using Maximum Likelihood Estimation (MLE), unlike linear regression which uses the Ordinary Least Squares (OLS) approach. MLE is an iterative procedure, meaning that it starts with a guess as to the best weight for each predictor variable (that is, each coefficient in the model) and then adjusts these coefficients repeatedly until there is no additional improvement in the ability to predict the value of the outcome variable (either 0 or 1) for each case. Life insurance actuaries use logistic regression to predict, based on given data on a policy holder (e.g. age, gender, results from a physical examination) the chances that the policy holder will die before the term of the policy expires. Political campaigns try to predict the chances that a voter will vote for their candidate (or do something else desirable, such as donate to the campaign).

Chapter 5

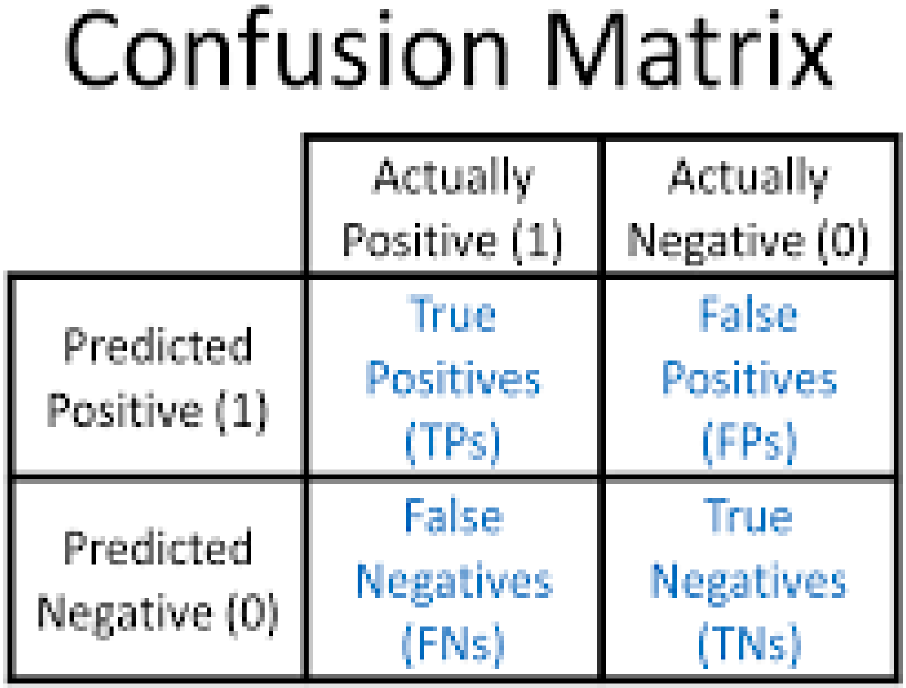
IMPLEMENTAION

In this work, a business intelligent model has been developed, to classify different diabetes type, based on attribute values using a suitable machine learning technique. The model was evaluated by a scientific approach to measure accuracy. We are using Logistic Regression to build our model.
Analysis:

In this final phase, we will test our classification model on our prepared image dataset and also measure the performance on our dataset. To evaluate the performance of our created classification and make it comparable to current approaches, we use accuracy to measure the effectiveness of classifiers. After model building, knowing the power of model prediction on a new instance, is very important issue. Once a predictive model is developed using the historical data, one would be curious as to how the model will perform on the data that it has not seen during the model building process. One might even try multiple model types for the same prediction problem, and then, would like to know which model is the one to use for the real-world decision making situation, simply by comparing them on their prediction performance (e.g., accuracy). To measure the performance of a predictor, there are commonly used performance metrics, such as accuracy, recall etc. First, the most commonly used performance metrics will be described, and then some famous estimation methodologies are explained and compared to each other. "Performance Metrics for Predictive Modeling In classification problems, the primary source of performance measurements is a coincidence matrix (classification matrix or a contingency table)". Above figure shows a coincidence matrix for a two-class classification problem. The equations of the most commonly used metrics that can be calculated from the coincidence matrix.

Confusion matrix:

The belo0w figure 5.1 shows the Confusion matrix for the proposed system. The numbers along the diagonal from upper-left to lower-right represent the correct decisions made, and the numbers outside this diagonal represent the errors. "The true positive rate (also



Confusion Matrix

	Actually Positive (1)	Actually Negative (0)
Predicted Positive (1)	True Positives (TPs)	False Positives (FPs)
Predicted Negative (0)	False Negatives (FNs)	True Negatives (TNs)

Figure 5.1: Confusion Matrix

called hit rate or recall) of a classifier is estimated by dividing the correctly classified positives (the true positive count) by the total positive count. The false positive rate (also called a false alarm rate) of the classifier is estimated by dividing the incorrectly classified negatives (the false negative count) by the total negatives. The overall accuracy of a classifier is estimated by dividing the total correctly classified positives and negatives by the total number of samples.

Numpy: Numpy, which stands for Numerical Python, is a library consisting of multidimensional array objects and a collection of routines for processing those arrays. Using NumPy, mathematical and logical operations on arrays can be performed. This tutorial explains the basics of NumPy such as its architecture and environment. It also discusses the various array functions, types of indexing, etc. An introduction to Matplotlib is also provided. All this is explained with the help of examples for better understanding.Numpy is a Python package. It stands for 'Numerical Python'.
Numeric: the ancestor of NumPy, was developed by Jim Hugunin. Another package Numarray was also developed, having some additional functionality. In 2005, Travis Oliphant created NumPy package by incorporating the features of Numarray into Numeric package. There are many contributors to this open source project.Operations Using NumPy, a developer can perform the following operations

- Mathematical and logical operations on arrays.
- Fourier transforms and routines for shape manipulation.

- Operations related to linear algebra. NumPy has in-built functions for linear algebra and random number generation

5.1 Steps Involved In Predicting The Diabetes disease:

(a) **import standard libraries**
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np

(b) **import libraries for prepearing model:**
from sklearn.model_selection
import train_test_split from sklearn import metrics

(c) **import library for Logistic regression:**
fromsklearn.linear$_{m}odelimportLogisticRegression$

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. The datasets consist of several medical predictor (independent) variables and one target (dependent) variable, Outcome. Independent variables include the number of pregnancies the patient has had, their BMI, insulin level, age, and so on.

Chapter 6

SYSTEM DESIGN AND ANALYSIS

System Design is a process of planning a new business system or replacing an existing system by defining its components or modules to satisfy the specific requirements. Before planning, you need to understand the old system thoroughly and determine how computers can best be used in order to operate efficiently. System analysis is conducted for the purpose of studying a system or its parts in order to identify its objectives. It is a problem solving technique that improves the system and ensures that all the components of the system work efficiently to accomplish their purpose.

6.0.1 Preliminary Design

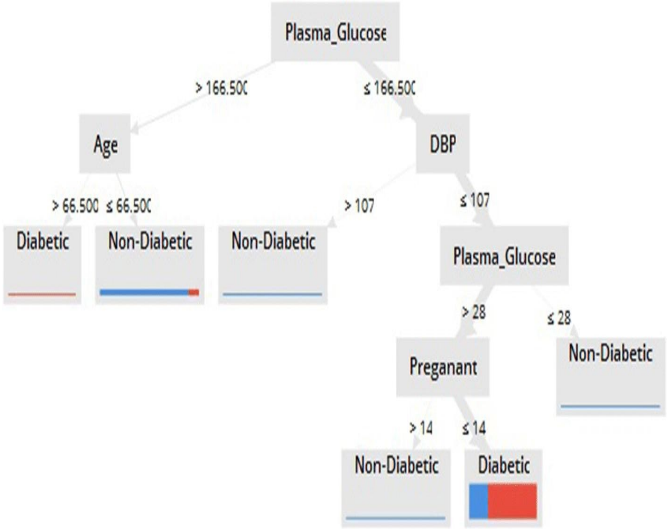


Figure 6.1: Preliminary design of Diabetes Prediction system

The above figure 6.1 depicts the preliminary design of Diabetes prediction system. The system initially considers the plasma Glucose level of an individual with the age of the person. Based on the value of Glucose level it categorizes the person whether the person is diabetic or non diabetic. Here the design shows that the person whose glucose level is ¿66.500 is considered to be diabetic and person whose glucose value is 66.500 will be considered as non diabetic. If DBP is greater than 107 the patient will be non diabetic. If the DBP is less than or equal to 107 and plasma glucose level is greater than 28 and pregnant is considered to be diabetic otherwise non diabetic.

6.0.2 System Architecture

The below figure 6.2 shows the architecture of Diabetes Prediction System. The training data set is fed to the system as input which will be initially pre processed. Data pre processing is the phase where the raw data will be transformed into meaningful and understandable format.

The pre processed data will be later classified using the best classification mechanism. Then the classified will be compared with the test data in order to classify it accurately using some distance measures. The final classified data will be converted to data patterns using intelligent methods. The obtained patterns will be evaluated for accuracy and correctness. The identified patterns will be represented as knowledge in the required form as output.

6.0.3 Data Flow Diagram

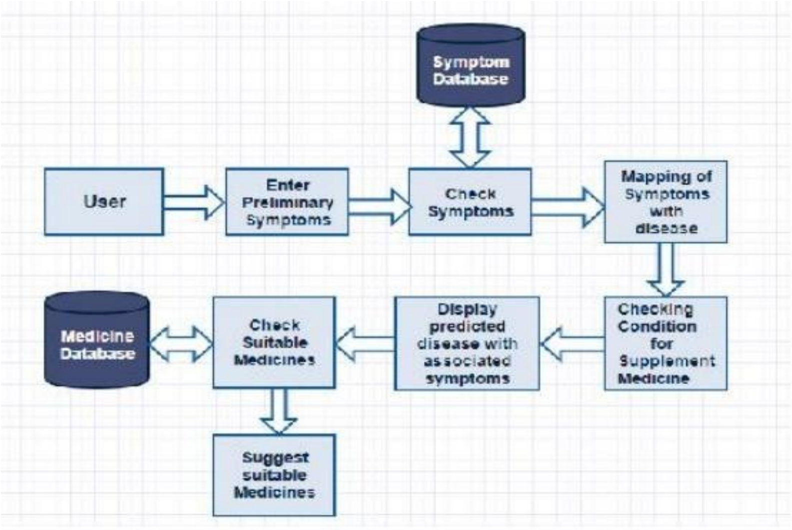

Symptom Database
User
Enter Preliminary Symptoms
Check Symptoms
Mapping of Symptoms with disease
Medicine Database
Check Suitable Medicines
Display predicted disease with associated symptoms
Checking Condition for Supplement Medicine
Suggest suitable Medicines

Figure 6.2: Data Flow Diagram

The above figure 6.2 shows the Data Flow Diagram of Diabetes Prediction. The user will enter preliminary symptoms those symptoms will be compared with symptoms database. Symptoms will be mapped with disease. Display predicted disease with associated symptoms after that check suitable medicine and suggest suitable medicines from the medicine database.

6.0.4 DFD for Data Extraction

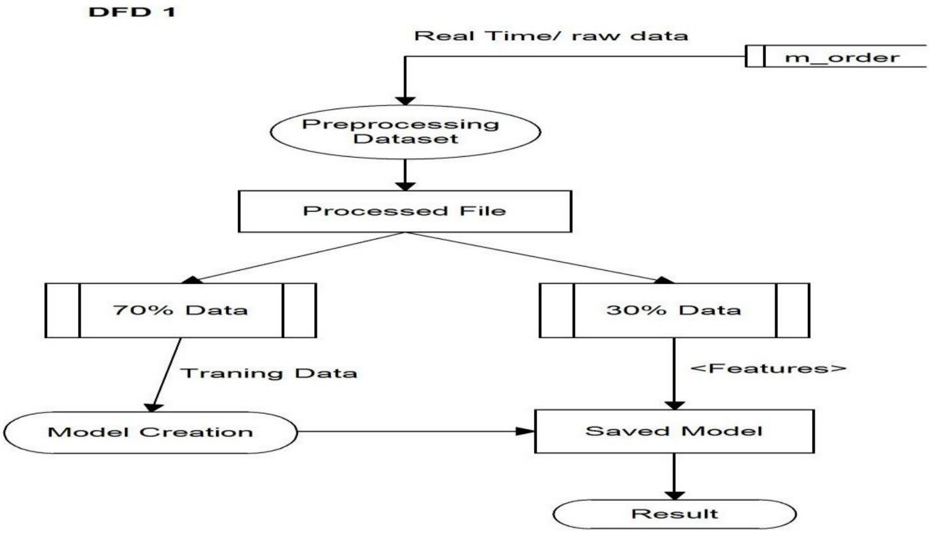


Figure 6.3: DFD for Data Extraction

The above figure 6.3 shows the DFD for Data Extraction for the proposed system. Raw Data will be pre processed. Processed file will be divided into 70% of training data and 30% of test data. Model is created for the training data and save the model. Result will be obtained for the test data using the saved model.

6.0.5 DFD for Classification of Data

The above figure 4.5 shows the DFD for Classification of Data of the proposed system. For the problem definition data will be collected and pre processed. Model will be created for the gathered data and data access, data sampling, data transformation will be done. After creating the model and evaluate and interpret it. Created model is applied for external applications. 33

6.0.6 Use Case Diagram

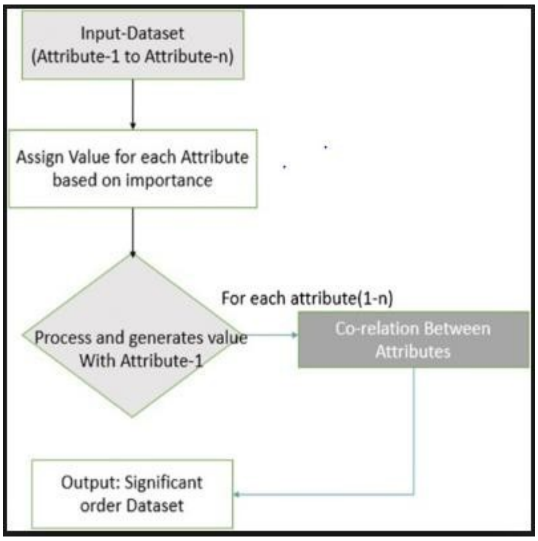

Input-Dataset
(Attribute-1 to Attribute-n)
Assign Value for each Attribute based on importance
For each attribute(1-n)
Process and generates value With Attribute-1
Co-relation Between Attributes
Output: Significant order Dataset

Figure 6.4: Use Case Diagram

The figure 6.4 above shows the use case diagram of the proposed system. Values for each attribute assigned based on importance for the input dataset. Process and generate value. If the value is equal to one then the patient is diabetic otherwise the patient is non diabetic.

Chapter 7

EXPERIMENTAL RESULTS

7.1 Outcomes of the Proposed System

Diabetes prediction is an internet-primarily based device gaining knowledge of utility, skilled through a Pima Indian dataset. The person inputs its particular clinical information to get the prediction of diabetes. The set of rules will calculate the opportunity of presence diabetes.

Data distribution:

The datasets consist of several medical predictor (independent) variables and one target (dependent) variable, Outcome. Independent variables include the number of pregnancies the patient has had, their BMI, insulin level, age, and so on.

The figure 4.6 above shows the use case diagram of the proposed system. Values for each attribute assigned based on importance for the input dataset. Process and generate value. If the value is equal to one then the patient is diabetic otherwise the patient is non diabetic.

Chapter 8

TESTING

Software testing is defined as an activity to check whether the actual results match the expected results and to ensure that the software system is Defect free. It involves execution of a software component or system component to evaluate one or more properties of interest.

8.0.1 Testing and Validations

Validation is a complex process with many possible variations and options, so specifics vary from database to database, but the general outline is:

(a) **Requirement Gathering**

   i. The Sponsor decides what the database is required to do based on regulations, company needs, and any other important factors.

   ii. The requirements are documented and approved.

(b) System Testing

   i. Procedures to test the requirements are created and documented.

   ii. The version of the database that will be used for validation is set up.

   iii. The Sponsor approves the test procedures.

   iv. The tests are performed and documented.

   v. Any needed changes are made. This may require another, shorter round of testing and documentation.

(c) **System Release**

   i. The validation documentation is finalized.

   ii. The database is put into production.

8.1 Testing Levels

8.1.1 Functional Testing:

This type of testing is done against the functional requirements of the project. Types: Unit testing: Each unit /module of the project is individually tested to check for bugs. If any bugs found by the testing team, it is reported to the developer for fixing. Integration testing: All the units are now integrated as one single unit and checked for bugs. This also checks if all the modules are working properly with each other. System testing: This testing checks for operating system compatibility. It includes both functional and non functional requirements. Sanity testing: It ensures change in the code doesn't affect the working of the project. Smoke testing: this type of testing is a set of small tests designed for each build. Interface testing: Testing of the interface and its proper functioning. Regression testing: Testing the software repetitively when a new requirement is added, when bug fixed etc. Beta/Acceptance testing: User level testing to obtain user feedback on the product.

8.1.2 Non-Functional Testing

This type of testing is mainly concerned with the non-functional requirements such as performance of the system under various scenarios. Performance testing: Checks for speed, stability and reliability of the software, hardware or even the network of the system under test. Compatibility testing: This type of testing checks for compatibility of the system with different operating systems, different networks etc. Localization testing: This checks for the localized version of the product mainly concerned with UI. Security testing: Checks if the software has vulnerabilities and if any, fix them. Reliability testing: Checks for the reliability of the software Stress testing: This testing checks the performance of the system when it is exposed to different stress levels. Usability testing: Type of testing checks the easily the software is being used by the customers. Compliance testing: Type of testing to determine the compliance of a system with internal or external standards.

- **Reliability:** The structure must be reliable and strong in giving the functionalities. The movements must be made unmistakable by the structure when a customer has revealed a couple of enhancements. The progressions made by the Programmer must be Project pioneer and in addition the Test designer.
- **Maintainability:**The system watching and upkeep should be fundamental and focus in its approach. There should not be an excess of occupations running on diverse machines such that it gets hard to screen whether the employments are running without lapses.
- **Performance:**The framework will be utilized by numerous representatives all the while. Since the system will be encouraged on a single web server with a lone database server outside of anyone's ability to see, execution transforms into a significant concern. The structure should not capitulate when various customers would use everything the while. It should allow brisk accessibility to each and every piece of its customers. For instance, if two test specialists are all

the while attempting to report the vicinity of a bug, then there ought not to be any irregularity at the same time.

- **Portability:**The framework should to be effectively versatile to another framework. This is obliged when the web server, which s facilitating the framework gets adhered because of a few issues, which requires the framework to be taken to another framework.
- **Scalability:**The framework should be sufficiently adaptable to include new functionalities at a later stage. There should be a run of the mill channel, which can oblige the new functionalities.
- **Flexibility:**Flexibility is the capacity of a framework to adjust to changing situations and circumstances, and to adapt to changes to business approaches and rules. An adaptable framework is one that is anything but difficult to reconfigure.

8.2 Different Stages of Testing

8.2.1 Unit Testing

Unit Testing is a level of software testing where individual units/ components of a software are tested. The purpose is to validate that each unit of the software performs as designed. A unit is the smallest testable part of any software. It usually has one or a few inputs and usually a single output. In procedural programming, a unit may be an individual program, function, procedure, etc. In object-oriented programming, the smallest unit is a method, which may belong to a base/ super class, abstract class or derived/ child class. (Some treat a module of an application as a unit. This is to be discouraged as there will probably be many individual units within that module.) Unit testing frameworks, drivers, stubs, and mock/ fake objects are used to assist in unit testing.

White Box Testing

White Box Testing is defined as the testing of a software solution's internal structure, design, and coding. In this type of testing, the code is visible to the tester. It focuses primarily on verifying the flow of inputs and outputs through the application,

improving design and usability, strengthening security. White box testing is also known as Clear Box testing, Open Box testing, Structural testing, Transparent Box testing, Code-Based testing, and Glass Box testing. It is usually performed by developers.

It is one of two parts of the "Box Testing" approach to software testing. Its counterpart, Blackbox testing, involves testing from an external or end-user type perspective. On the other hand, Whitebox testing is based on the inner workings of an application and revolves around internal testing.

The term "WhiteBox" was used because of the see-through box concept. The clear box or WhiteBox name symbolizes the ability to see through the software's outer shell (or "box") into its inner workings. Likewise, the "black box" in "Black Box Testing" symbolizes not being able to see the inner workings of the software so that only the end-user experience can be tested.

i. Internal security holes
ii. Broken or poorly structured paths in the coding processes
iii. flow of specific inputs through the code output functionality of conditional loops
of each statement, object, and function on an individual basis

The testing can be done at system, integration and unit levels of software development. One of the basic goals of whitebox testing is to verify a working flow for an application. It involves testing a series of predefined inputs against expected or desired outputs so that when a specific input does not result in the expected output, you have encountered a bug.

Different Stages of Testing
Unit Testing

Unit Testing is a level of software testing where individual units/ components of a software are tested. The purpose is to validate that each unit of the software performs as designed. A unit is the smallest testable part of any software. It usually has one or a few inputs and usually a single output. In procedural programming, a unit may be an individual program, function, procedure, etc. In object-oriented programming, the smallest unit is a method, which may belong to a base/ super class, abstract class or derived/ child class. (Some treat a module of an application as a unit. This is to be discouraged as there will probably be many individual units within that module.) Unit testing frameworks, drivers, stubs, and mock/ fake objects are used to assist in unit testing. Unit Test Plan:

i. Prepare
ii. Review
iii. Rework
iv. Baseline
v. Unit Test Cases/Scripts
vi. Prepare
vii. Review
viii. Rework
ix. Baseline Unit Test
x. Perform

Benefits

Unit testing increases confidence in changing/ maintaining code. If good unit tests are written and if they are run every time any code is changed, we will be able to promptly catch any defects introduced due to the change. Also, if codes are already made less interdependent to make unit testing possible, the unintended impact of changes to any code is less.

Codes are more reusable. In order to make unit testing possible, codes need to be modular. This means that codes are easier to reuse. Development is faster. How? If you do not have unit testing in place, you write your code and perform that fuzzy ‘developer test’ (You set some breakpoints, fire up the GUI, provide a few inputs that hopefully hit your code and hope that you are all set.) But, if you have unit testing in place, you write the test, write the code and run the test. Writing tests takes time but the time is compensated by the less amount of time it takes to run the tests; You need not fire up the GUI and provide all those inputs. And, of course, unit tests are more reliable than ‘developer tests’. Development is faster in the long run too. How? The effort required to find and fix defects found during unit testing is very less in comparison to the effort required to fix defects found during system testing or acceptance testing. The cost of fixing a defect detected during unit testing is lesser in comparison to that of defects detected at higher levels. Compare the cost (time, effort, destruction, humiliation) of a defect detected during acceptance testing or when the software is live. Debugging is easy. When a test fails, only the latest changes need to be debugged. With testing at higher levels, changes made over the span of several days/weeks/months need to be scanned.

Integration Testing

Integration Testing is a level of software testing where individual units are combined and tested as a group. The purpose of this level of testing is to expose faults in the interaction between integrated units. Test drivers and test stubs are used to assist in Integration Testing.

integration testing: Testing performed to expose defects in the interfaces and in the interactions between integrated components or systems. See also component integration testing, system integration testing. component integration testing: Testing performed to expose defects in the interfaces and interaction between integrated components.

system integration testing: Testing the integration of systems and packages; testing interfaces to external organizations (e.g. Electronic Data Interchange, Internet).

Tasks

i. Prepare
ii. Review
iii. Rework
iv. Baseline
v. Prepare
vi. Review
vii. Rework
viii. Baseline Integration Test

System Testing

System Testing is a level of software testing where a complete and integrated software is tested. The purpose of this test is to evaluate the system’s compliance with the specified requirements. The process of testing an integrated system to verify that it meets specified requirements.

Acceptance Testing

Acceptance Testing is a level of software testing where a system is tested for acceptability. The purpose of this test is to evaluate the system's compliance with the business requirements and assess whether it is acceptable for delivery. Formal testing with respect to user needs, requirements, and business processes conducted to determine whether or not a system satisfies the acceptance criteria and to enable the user, customers or other authorized entity to determine whether or not to accept the system.

Chapter 9

EXPERIMANTAL RESULTS

The dataset consists around 800 user's record which are has been considered for the analysis and that dataset is taken from kaggle. Totally 9 attributes are take place in the dataset. The attribute of the dataset are pregnancies, glucose, blood pressure, skin thickness, insulin, BMI, age, outcome. These attributes are used to classify and predict the diabetes. And the SVM data mining algorithm is used for both classification and prediction. The language chosen to design this system is Python the forms and the database connectivity everything is developed using Flask framework. This Application is developed using flask Framework so it is purely a windows desktop application and it is not platform independent software.

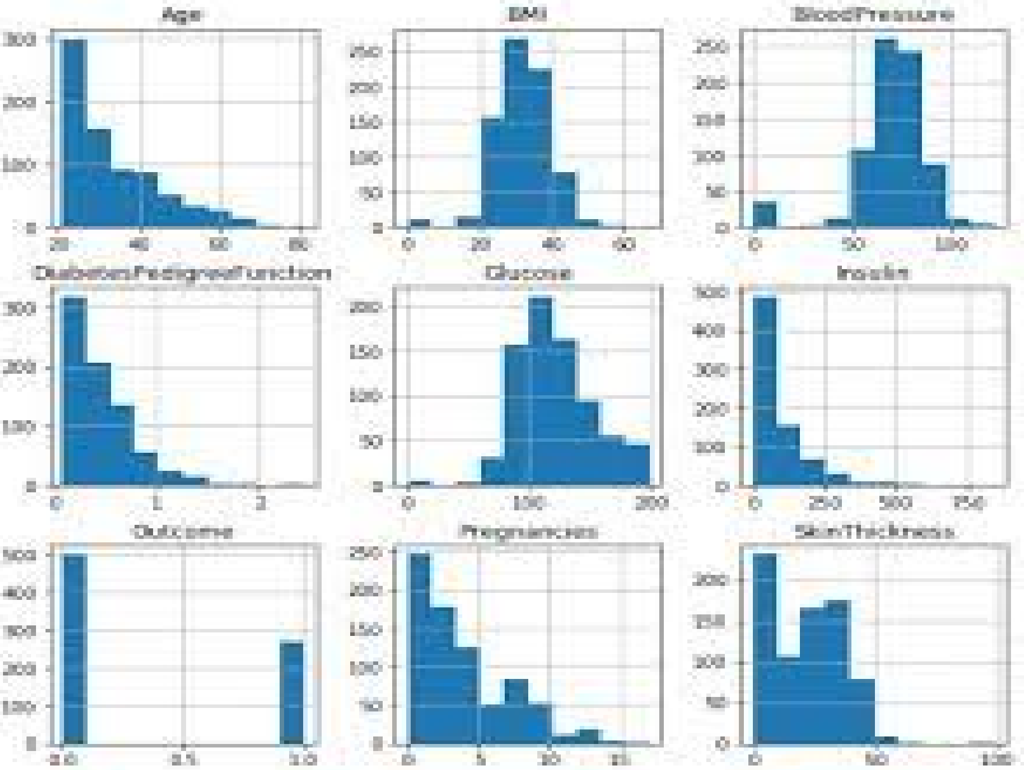


Figure 9.1: Data Distribution

Chapter 10

CONCLUSION

Machine learning has the great feature to revolutionize the Diabetes Mellitus risk prediction with the assistance of advanced computational ways and availability of enormous amount of medical specialty and genetic diabetes risk dataset. Detection of Diabetes Mellitus in its early stages is the key for treatment. This work has detailed machine learning approach to predicting diabetes levels.Machine learning has the great ability to revolutionize the diabetes risk prediction with the help of advanced computational methods and availability of large amount of epidemiological and genetic diabetes risk dataset. Detection of diabetes in its early stages is the key for treatment. This work has described a machine learning approach to predicting diabetes levels. The technique may also help researchers to develop an accurate and effective tool that will reach at the table of clinicians to help them make better decision about the disease status. ‘

Chapter 11

Future Enhancement

This technique may also help researchers to develop an accurate and effective tool that will reach at the table of clinicians to help them make better decision about the disease status. In this study, a systematic effort was made to identify and review machine learning and data mining approaches applied on DM research. DM is rapidly emerging as one of the greatest global health challenges of the 21st century. To date, there is a significant work carried out in almost all aspects of DM research and especially biomarker identification and prediction-diagnosis. The advent of biotechnology, with the vast amount of data produced, along with the increasing amount of EHRs is expected to give rise to further in-depth exploration toward diagnosis, etiopathophysiology and treatment of DM through employment of machine learning and data mining techniques in enriched datasets that include clinical and biological information.

11.0.1 Limitations

- The size of the database increases day-by-day, increasing the load on the database back update data maintenance activity.
- Training for simple computer operations is necessary for the users working on the system.

Bibliography

[1] D. Biber, S. Conrad, R. Reppen, Corpus Linguistics: Investigating Language Structure and Use, Cambridge University Press, 1998.

[2] Zunera Jalil, “Copyright Protection of Plain Text Using Digital Watermarking”, *submitted at FAST National University of Computer and Emerging Sciences*, Islamabad, Pakistan, 2010.

[3] Sonika Rathi, “Medical Image Authentication through Watermarking Preserving ROI”, *submitted at College of Engineering*, Pune, India, 2012.

[4] I. Cox and M. L. Miller, “A review of watermarking and the importance of perceptual modeling”, *in Proceedings of SPIE, Human Vision and Electronic Imaging II*, vol. 3016, no. 2, pp. 92-99, 1997.

[5] Ingemar J. Cox, Matt L. Miller and Jeffrey A. Bloom, “Watermarking applications and their properties”, *Published in the Int. Conf. on Information Technology*, 2000.

[6] A. Khan, Intelligent Perceptual Shaping of a digital Watermark, PhD Thesis, Faculty of Computer Science and Engineering, GIK Institute, Pakistan, 2006.

[7] W. Chung, H. Chen, W. Chang, S. Chou, Fighting cybercrime: a review and the Taiwan experience, Decision Support Systems 41 (2006) 669–682.

[8] N. Diakopoulos, M. Naaman, Towards quality discourse in online news comments, in: Proceedings of the 2011 ACM Conference on Computer Supported Cooperative Work, pp. 133–148.

[9] J.L. Fleiss, J. Cohen, The equivalence of weighted kappa and the intraclass correlation coefficient as measures of reliability, Educational and Psychological Measurement 33 (1973) 613–619.

13 R. Hamming, Error detecting and error correcting codes, Bell System Techincal Journal 29 (1950) 147–160.

[10] V. Freschi, A. Seraghiti, A. Bogliolo, Filtering obfuscated email spam by means of phonetic string matching, in: Advances in Information Retrieval, pp. 505-509.