PREDICTION OF DIABETES USING MACHINE LEARNING

A project report submitted in partial fulfillment of the requirements for the award of the degree of

BACHELOR OF TECHNOLOGY

In

COMPUTER SCIENCE AND ENGINEERING

Submitted by

SEGU VENKATA SRI SANTHI

(17A31A0588)

LANKA SUNDAR PRASANNA KUMAR

(17A31A05A6)

SABBELLA VENKATESWARA AYYAPPA REDDY

(17A31A05B2)

K LAKSHMI VIVEKA, MTech

Assistant Professor

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

PRAGATI ENGINEERING COLLEGE

(AUTONOMOUS)

(Approved by AICTE & Permanently Affiliated to JNTUK, Kakinada & Accredited by NAAC)

1-378, ADB Road, Surampalem, E.G.Dist., A.P, Pin-533437.

2017-2021

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

PRAGATI ENGINEERING COLLEGE
(AUTONOMOUS)

(Approved by AICTE & Permanently Affiliated to JNTUK & Accredited by NAAC)

1-378, ADB Road, Surampalem, E.G.Dist., A.P, Pin-533437.

CERTIFICATE

This is to certify that the report entitled **"PREDICTION OF DIABETES USING MACHINE LEARNING"** *that is being submitted by* ***S.V.SRI SANTHI, L.SUNDAR, S.V.AYYAPPA REDDY of III Year II Semester bearing (17A31A0588,17A31A05A6, 17A31A05B2)****, in partial fulfillment for the award of the Degree of Bachelor of Technology in Computer Science and Engineering, Pragati Engineering College is a record of Bonafide work carried out by them.*

Supervisor

K LAKSHMI VIVEKA,MTech

Assistant Professor

Head of the Department

M .RadhikaMani, M.Tech, PhD

Assoc. Professor

ACKNOWLEDGEMENT

Presentation inspiration and motivation have always played a key role in the success of any venture.

It is our privilege to express our sincerest regards to our project supervisor, ***K LAKSHMI VIVEKA*** ,M.Tech for their valuable inputs, able guidance, encouragement, whole-hearted cooperation and constructive criticism throughout the duration of our project.

We deeply express our sincere thanks to our Head of Department of Computer Science and Engineering**. Dr .M. RADHIKA MANI** for encouraging and allowing us to present the project on the topic at "**PREDICTION OF DIABETES USING MACHINE LEARNING**" .Our department premises for the partial fulfillment of the requirements leading to the award of B-Tech degree.

I would like to express my special thanks of gratitude to my principle **Dr. S. SAMBHU PRASAD of Pragati Engineering College.**

S.V.SRI SANTHI

L.SUNDAR

S.V.AYYAPPA REDDY

ABSTRACT

Diabetes is a chronic disease with the potential to cause a worldwide health care crisis. Diabetes mellitus or simply diabetes is a disease caused due to the increase level of blood glucose. Various traditional methods, based on physical and chemical tests, are available for diagnosing diabetes. However, early prediction of diabetes is quite challenging task for medical practitioners due to complex interdependence on various factors as diabetes affects human organs such as kidney, eye, heart, nerves, foot etc. Machine learning is an emerging scientific field in data science dealing with the ways in which machines learn from experience. The aim of this project is to develop a system which can perform early prediction and recommendation of diabetes for a patient with a higher accuracy by combining the results of different machine learning techniques. This research has focused on developing a system based on three classification methods namely, Decision Tree, Naïve Bayes, and Support Vector Machine algorithms.

Classification strategies are broadly used in the medical field for classifying data into different classes according to some constrains comparatively an individual classifier. Diabetes is an illness which affects the ability of the body in producing the hormone insulin, which in turn makes the metabolism of carbohydrate abnormal and raise the levels of glucose in the blood. In Diabetes a person generally suffers from high blood sugar. Intensify thirst, intensify hunger and Frequent urination are some of the symptoms caused due to high blood sugar.

Many complications occur if diabetes remains untreated. Some of these severe complications include diabetic ketoacidosis and non ketotichyperosmolar coma.

Diabetes is examined as a vital serious health matter during which the measure of sugar substance cannot be controlled. Diabetes is not only affected by various factors like height, weight, hereditary factor and insulin but the major reason considered is sugar concentration among all factors. The early identification is the only remedy to stay away from the complications.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

[Abbreviation Short Form] **[Abbreviation Full Form]**

AI	-	Artificial Intelligence
ML	-	Machine learning
HRS	-	Healthcare Recommender System
FPG	-	Fasting Plasma Glucose
CGTT	-	Casual Glucose Tolerance Test
HBA1C	-	Glycated Hemoglobin Tests
T1D	-	Type1 Diabetes
T2D	-	Type2 Diabetes
GDM	-	Gestational Diabetes
BG	-	Blood Glucose
MDI	-	Multiple Daily Injections
CSII	-	Continuous Subcutaneous Insulin Infusion
CNN	-	Convolutional Neural Network
KNN	-	K-Nearest Neighbor
SVM	-	Support Vector Machine
ROC	-	Receiver Operating Curve
B.P	-	Blood Pressure
BMI	-	Body Mass Index
CART	-	Classification and Regression Tree Algorithm
TPR	-	True Positive Rate
FPR	-	False Positive Rate
AUC	-	Area under the Curve
UCI	-	UC Irvine Machine Learning Repository

CHAPTER 1

INTRODUCTION

Diabetes is one of the deadliest diseases in the world. It is not only a disease but also a creator of different kinds of diseases like heart attack, blindness, kidney diseases, etc. The normal identifying process is that patients need to visit a diagnostic center and it is waste of time. So, by using machine learning we can easily predict the disease with suitable recommendations.

1.1 ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

Machine Learning is an extensively algorithm driven study which makes computer/device/software capable of learning on the basis of their own previous experience and improve the performance of a task. It also gives machines/software ability to analyze, predict and sort huge amounts of data.

Machine learning is a technology that helps to improve the services provided by systems, web, and smart phones. The terms Machine Learning and Artificial intelligence seem to be connected sometimes but they are quite distinct in the area of computing.

Artificial Intelligence is a branch of computer science that aims to create intelligent machines. It has become an essential part of the technology industry.

Artificial intelligence methods in combination with the latest technologies, including medical devices, mobile computing, and sensor technologies, have the potential to enable the creation and delivery of better management services to deal with chronic diseases. One of the most lethal and prevalent chronic diseases is diabetes mellitus, which is characterized by dysfunction of glucose homeostasis.

1.2 CLASSIFICATION:

In machine learning, classification is a supervised leaning approach in which the computer program learns from the data input given to it and then uses this learning to classify new observation. Classification strategies are broadly used in the medical field for classifying data into different classes according to some constraints comparatively an individual classifier.

1.3PREDICTIONS AND RECOMMENDATIONS:

Prediction:

With the rise of Machine Learning approaches, we have the ability to find a solution to this issue, we have developed a system using data mining 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. Healthcare Recommendation System (HRS) using machine learning can be developed to predict about the

health condition by analyzing patient's life style, physical health factors, mental health factors and their social network activities. For example, on training the model with the age of women and diabetes condition helps to predict the chances of getting diabetes for new women patients without detailed diagnosis.

Diabetes can be predicted by the glucose levels in the body. If the pancreas damage then there will be reduction in the insulin production. Then there will be increase in glucose levels in the body. Then diabetes can be predicted.

Prediction is expressed as a numerical value that represents the disease risk diagnosis for future cases based on active patients. The rules for detecting various ranges for diabetes based on Fasting Plasma Glucose test (FPG), Casual Glucose Tolerance Test (CGTT) and Glycated Hemoglobin tests (HBA1C) are given below:

Rule 1: No Diabetes Range

If FPG has a level between 70 and 100 mg/dl (3.9 and 5.6 mmol/L) and

If the blood glucose level below 125 mg/dl in CGTT and

If HBA1C value is below 97 mg/dl, then it indicates- no diabetes range.

Rule 2: Pre-diabetes Range

If FPG ranges from 100 mg/dl to 125 mg/dl and

CGTT ranges from 140 mg/dl to 199 mg/d and

HBA1C test values lie in range 97-154 mg/dl, it indicates pre-diabetes range.

Rule 3: Diabetes Range

If FPG is 126 mg/dl or more and

CGTT is 200 mg/dl or more and

HBA1C is greater than 180 mg/dl, it indicates diabetes

Classification of diabetes:

Diabetes can be categorized into 3 subgroups: type 1 diabetes (T1D), type 2 diabetes (T2D), and gestational

diabetes (GDM)

Type 1 Diabetes (T1D):

T1D patients have severe impairments in insulin production, and must use external insulin exclusively to manage their blood glucose (BG). Treatment of T1D requires consistent doses of insulin through multiple daily injections (MDIs) or continuous subcutaneous insulin infusion (CSII) using a pump.

Type 2 Diabetes (T2D):

Over the long term, T2D patients become resistant to the normal effects of insulin and gradually lose their capacity to produce enough of this hormone. A wide range of therapeutic options are available for patients with T2D. At the early stages of disease, they commonly receive medications that improve insulin secretion or insulin absorption, but eventually they must receive external doses of insulin.

Type 3 Diabetes (GDM):

On the other hand, GDM is treated similarly to T2D, but only occurs during pregnancy due to the interaction between insulin and hormones released by the placenta.

Recommendations:

The next phase is recommendation which is expressed as the suggestion required by the patients. One way is to inject insulin into the body in the form of injections to balance the glucose levels in the body. For example, non-healthcare professional might be requiring alternative remedies for treating diabetes. The recommendation rules will be created by taking opinion from many doctors for different possible scenarios. Deep learning using CNN with auto-encoders or similar will be exploited for this task.

- Attain and maintain optimal metabolic outcomes including
- Blood glucose levels in the normal range or as close to normal as is safely possible to prevent or reduce the risk for complications of diabetes.
- A lipid and lipoprotein profile that reduces the risk for macro vascular disease.
- Blood pressure levels that reduce the risk for vascular disease.
- Improve health through healthy food choices and physical activity.

Diabetes is difficult to control and it is important to manage the diabetic’s blood sugar level and prevent the associated complications by appropriate diabetic treatment. This paper proposes a system that can provide appropriate management for diabetes patients, according to their blood sugar level.

The system is designed to send the information about the blood sugar levels, blood pressure, food consumption, exercise, *etc.*, of diabetes patients, and manage the treatment by recommending and monitoring food consumption, physical activity, insulin dosage, *etc.*, so that the patient can better manage their condition. The system is based on rules and the K Nearest Neighbor (KNN) classifier algorithm, to obtain the optimum treatment recommendation.

Diabetes mellitus refers collectively to a group of diseases resulting from dysfunction of the glucoregulatory system.

Hyperglycaemia, the hallmark of diabetes, is the primary consequence of this dysregulation. Chronic hyperglycaemia in diabetes is associated with long-term complications involving tissue damage and organ failure, which can decrease life expectancy and even cause death. The International Diabetes Federation estimates that, by 2017, diabetes affected 425 million people worldwide, of whom, 4 million died in the same year. These figures are expected to increase dramatically in the coming decades, placing a rising burden on health care systems

Most diabetes can be categorized into 3 subgroups: type 1 diabetes (T1D), type 2 diabetes (T2D), and gestational diabetes (GDM). Over the long term, T2D patients become resistant to the normal effects of insulin and gradually lose their capacity to produce enough of this hormone. A wide range of therapeutic options are available for patients with T2D. At the early stages of disease, they commonly receive medications that improve insulin secretion or insulin absorption, but eventually they must receive external doses of insulin. On the other hand, T1D patients have severe impairments in insulin production, and must use external insulin exclusively to manage their blood glucose (BG). Treatment of T1D requires consistent doses of insulin through multiple daily injections (MDIs) or continuous subcutaneous insulin infusion (CSII) using a pump. GDM is treated similarly to T2D, but only occurs during pregnancy due to the interaction between insulin and hormones released by the placenta.

In each class of diabetes, timely diagnosis, education of patients in self-management, and continuous medical care are required to prevent acute complications (eg, ketoacidosis) and minimize the risk of long-term complications (e.g., nephropathy, retinopathy, diabetic foot, cardiovascular disease, or stroke). In addition to medication, management of diabetes requires adherence to an array of self-care behaviours that are often very burdensome for patients: carefully scheduling meals, counting carbohydrates, exercising, monitoring BG levels, and adjusting endeavours on a daily basis. The effects of non adherence to recommended treatment are not immediately evident and long-term complications may take years to develop.

CHAPTER 2
LITERATURE SURVEY

Evidence-based medicine is a powerful tool to help minimize treatment variation and unexpected costs. Large amount of healthcare data such as physician notes, medical history, medical prescription, lab and scan reports generated is useless until there is a proper method to process this data interactively in real - time. In this world filled with the latest technology, healthcare professionals feel more comfortable to utilize the social network to treat their patients effectively.To achieve this, an effective framework is needed which is capable of handling large amount of structured, unstructured and live streaming data about the patients from their social network activities. Healthcare Recommendation System (HRS) using machine learning can be developed to predict about the health condition by analysing patient's life style, physical health factors, mental health factors and their social network activities. For example, on training the model with the age of women and diabetes condition helps to predict the chances of getting diabetes for new women patients without detailed diagnosis.

2.1 BACK GROUND

In today's digital world people are prone to many health issues due to the sedentary lifestyle. The cost of medical treatments also keeps on increasing. An effective health care system is the one providing better personalized treatments with minimized cost. Medical expert systems are a branch of artificial intelligence that applies reasoning methods and domain specific knowledge to suggest recommendations like human experts. To enable reliable and fast decision-making process, medical expert knowledge needs to be stored as a knowledge-based system (KBS). KBS alone is not sufficient to suggest reliable recommendations due to the limitations in updating expert rules based on the population studies and limited personalization. Data driven approaches like data mining and machine learning can be applied to extract insights from the heterogeneous data of the patients. It provides individual recommendations based on the past learning experience and the patterns extracted from clinical data. Combination of information retrieval and machine learning can be used for medical database classification.

How do people know if they have diabetes?

People with diabetes frequently experience certain symptoms. These include:

- being very thirsty
- frequent urination
- weight loss

- increased hunger
- blurry vision
- irritability
- tingling or numbness in the hands or feet
- frequent skin, bladder or gum infections
- wounds that don't heal
- extreme unexplained fatigue

Who gets diabetes?

Diabetes can occur in anyone. However, people who have close relatives with the disease are somewhat more likely to develop it. Other risk factors include obesity, high cholesterol, high blood pressure, and physical inactivity. The risk of developing diabetes also increases as people grow older.

How is diabetes treated?

There are certain things that everyone who has diabetes, whether type 1 or type 2, needs to do to be healthy. They need to have a meal (eating) plan**.** They need to pay attention to how much physical activity they engage in, because physical activity can help the body use insulin better so it can convert glucose into energy for cells. Everyone with type 1 diabetes, and some people with type 2 diabetes, also need to take insulin injections**.** Some people with type 2 diabetes take pills called "oral agents" which help their bodies produce more insulin and/or use the insulin it is producing better. Some people with type 2 diabetes can manage their disease without medication by appropriate meal planning and adequate physical activity.

Also, people with diabetes need to learn how to monitor their blood glucose. Daily testing will help determine how well their meal plan, activity plan, and medication are working to keep blood glucose levels in a normal range.

2.2 EXISTING SYSTEM

Diabetes is one of the most chronic and deadliest diseases which causes an increase in blood sugar. Many complications occur, if diabetes is untreated and unidentified. In today’s world, people are visiting various hospitals and taking various treatments without knowing the correct treatment. One should first predict the disease at an early stage to avoid complex situations. Now a day’s people lack knowledge in predicting the disease and taking the correct treatment which affect their lives. They are wasting a lot of money on hospitals. The aim of our project is to overcome this problem.

There are many cases in which symptoms caused by high glucose levels in the blood do not appear immediately, and if systematic management is not provided, lead to complications such as cataracts, hardening of the arteries, kidney problems, abnormal nervous system conditions, loss of immunity, *etc.* Therefore, diabetes patients must monitor and manage their blood sugar constantly with diet control, exercise therapy, medication, *etc*. However, diabetes patients have difficulty managing this by themselves and they need constant management assistance and the help of their friends and family to maintain a lifestyle in which the diet is coordinated with exercise and activities. Therefore, there is a need for a system that is able to effectively allow control of the blood sugar of diabetes patients.

2.3 PROPOSED SYSTEM

With the help of machine learning approaches, we have the ability to find a solution to this issue. We should develop a system using data mining which has the ability to predict whether the patient has diabetes or not and find the correct treatment. Furthermore, predicting the disease early leads to treating the patients before it becomes critical. Data mining has the ability to extract hidden knowledge from a huge amount of diabetes-related data.

Evidence-based medicine is a powerful tool to help minimize treatment variation and unexpected costs. In this world filled with the latest technology, healthcare professionals feel more comfortable to utilize the social network to treat their patients effectively. To achieve this, an effective framework is needed which is capable of handling large amount of structured, unstructured and live streaming data about the patients from their social network activities. Healthcare Recommendation System (HRS) using machine learning can be developed to predict about the health condition by analyzing patient's life style, physical health factors, mental health factors and their social network activities. Therefore, the proposed system integrates the KNN classification. Therefore, the system is designed to select the method of treatment using the KNN classifier to evaluate time, blood sugar, blood pressure etc…

Graph representing growth of diabetes and AI:

Artificial intelligence (AI) is a quickly growing field, and its applications to diabetes research are growing even more rapidly as shown in below figure which is a gross estimate of the number of related articles in the Google Scholar database.

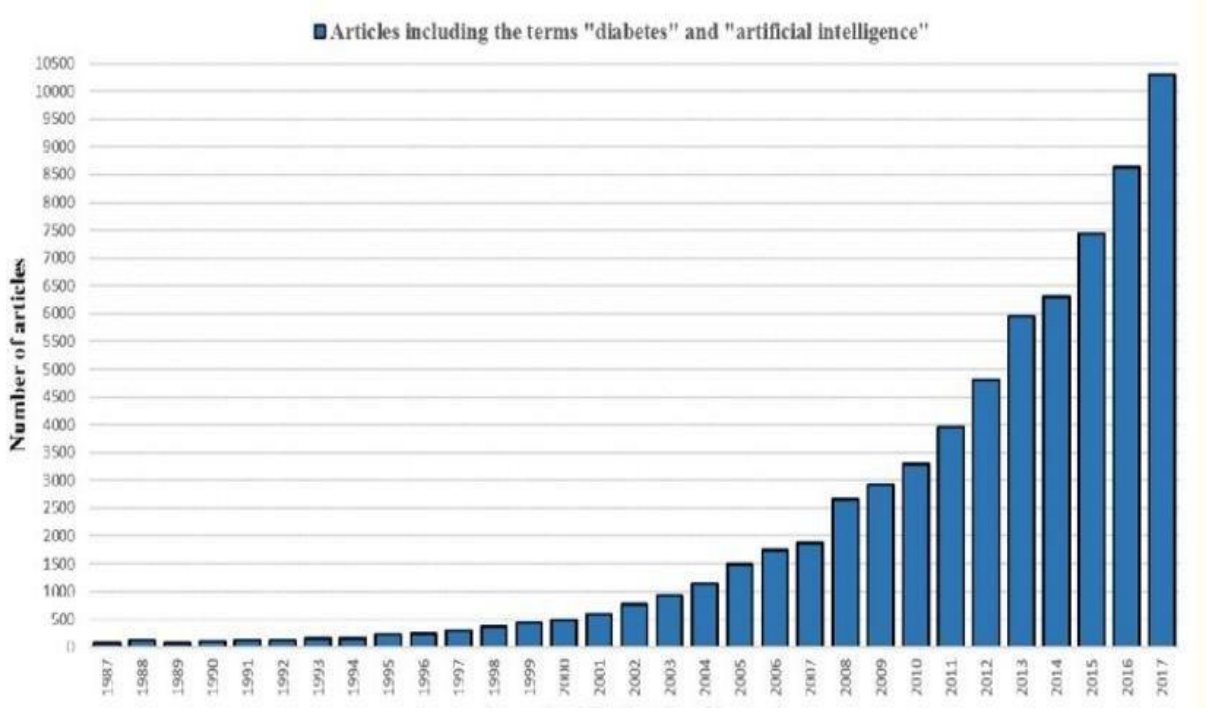

Fig 02.01: Graph representing growth of diabetes and AI

AI, as a science, can be defined as the ability to make computers do things that would require intelligence if done by humans. Increasingly, diabetes-related journals have been incorporating publications focused on AI tools applied to diabetes. Diabetes management scenarios have suffered a deep transformation that forces diabetologists to incorporate skills from new areas. This recently needed knowledge includes AI tools, which have become part of the diabetes health care.

CHAPTER 3

METHODOLOGY

Medical case history of diabetes patients will be stored which will contain information like blood sugar, blood pressure, weight etc. Diagnosis data comprises of physician notes, lab results, and medications. The data records may have many attributes, values and doctor's diagnosis for each case. Diagnosis scale ranges from 1 to 5 based on the severity of the disease, 5-represents critical condition, 4-represents severe requires immediate treatment, and 3-represents moderate requires further investigation, 2-represents normal, 1-represents within control. Along with this, demographic data of active patient like name, age, location, education level, wearable device, lifestyle, food habits and type of connectivity will also be collected.

3.1 MODEL DIAGRAM

Proposed procedure is summarized in figure-1 below in the form of model diagram. The figure shows the flow of the research conducted in constructing the model.

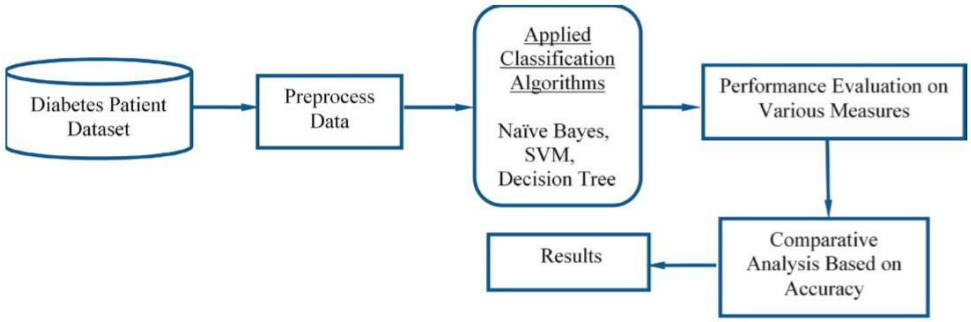

Fig 03.02: Model Diagram

3.2 ALGORITHMS USED:

SupportVectorMachine(SVM): SVM is one of the standard sets of supervised machine learning model employed in classification. Given a two-class training sample the aim of a support vector machine is to find the best highest-margin separating hyper plane between the two classes. For better generalization hyper plane should not lies closer to the data points belong to the other class. Hyper plane should be selected which is far from the data points from each category. The points that lie nearest to the margin of the classifier are the support vectors.

Naive Bayes Classifier: Naive Bayes is a classification technique with a notion which defines all features are independent and unrelated to each other. It defines that status of a specific feature in a class does not affect the status of another feature. Since it is based on conditional probability it is considered as a powerful algorithm employed for classification purpose. It works well for the data with unbalancing problems missing

values. Naive Bayes is a machine learning classifier which employs the Bayes Theorem.

Decision Tree Classifier: Decision Tree is a supervised machine learning algorithm used to solve classification problems. The main objective of using Decision Tree in this research work is the prediction of target class using decision rule taken from prior data. It uses nodes and internodes for the prediction and classification. Root nodes classify the instances with different features. Root nodes can have two or more branches while the leaf nodes represent classification. In every stage, Decision tree chooses each node by evaluating the highest information gain among all the attributes.

3.3 ACCURACY MEASURES& DATA:

KNN and Decision Tree algorithms are used in this research work. Experiments are performed using internal cross-validation 10-folds. Accuracy, F-Measure, Recall, Precision and ROC (Receiver Operating Curve) measures are used for the classification of this work. Table.03.01 defines accuracy measures below:

Measures	Definitions	Formula
1. Accuracy (A)	Accuracy determines the accuracy of the algorithm in predicting instances.	A=(TP+TN) / (Total no of samples)
2. Precision (P)	Classifier¢¢s correctness/accuracy is measured by Precision.	P = TP / (TP+ FP)
3. Recall (R)	To measure the classifier¢¢s completeness or sensitivity, Recall is used.	R =TP / (TP+FN)
4. F-Measure	F-Measure is the weighted average of precision and recall.	F=2*(P*R) / (P+R)
5. ROC	ROC(Receiver Operating Curve) curves are used to compare the usefulness of tests.	

Table.03. 01: Accuracy measures and data

SAMPLE DATA:

The dataset consists of 769 samples, out of which 500 are non diabetic while 269 are diabetic people.

- All patients are females of at least 21years of age.
- The dataset has total 9 attributes out of which 8 are independent variables and one is the dependent variable i.e. target variable which determines whether patient is having diabetes or not.

Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	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

Table.03. 02: Sample Dataset

ATTRIBUTE DETAILS:

- **Pregnancies**: No. of times pregnant
- **Glucose:** Plasma Glucose Concentration a 2 hour in an oral glucose tolerance test (mg/dl).

A 2-hour value between 140 and 200 mg/dl is called impaired glucose tolerance. This is called "pre diabetes." It means you are at increased risk of developing diabetes over time. A glucose level of 200 mg/dl or higher is used to diagnose diabetes.

- **Blood Pressure:** Diastolic Blood Pressure(mmHg) If Diastolic B.P > 90 means High B.P (High Probability of Diabetes) Diastolic B.P < 60 means low B.P (Less Probability of Diabetes)
- **Skin Thickness:** Triceps Skin Fold Thickness (mm) – A value used to estimate body fat. Normal Triceps Skin fold Thickness in women is 23mm. Higher thickness leads to obesity and chances of diabetes increases.
- **Insulin:** 2-Hour Serum Insulin (mu U/ml)

Feature	Normal Insulin Level
2 Hours After Glucose	16-166 mIU/L

Table.03. 03: Insulin Attribute details

Values above this range can be alarming.

- **BMI:** Body Mass Index (weight in kg/ height in m2) Body Mass Index of 18.5 to 25 is within the normal range BMI between 25 and 30 then it falls within the overweight range. A BMI of 30 or over falls within the obese range.
- **Diabetes Pedigree Function:** It provides information about diabetes history in relatives and genetic relationship of those relatives with patients. Higher Pedigree Function means patient is more likely to have diabetes.

- **Age:**(in years)
- **Outcome:** Class Variable (0 or 1) where '0' denotes patient is not having diabetes and '1' denotes patient having diabetes. The dependent variable is whether the patient is having diabetes or not.

DATA UNDERSTANDING:

Diabetes Pedigree Function: It provides information about diabetes history in relatives and genetic relationship of those relatives with patients. Higher Pedigree Function means patient is more likely to have diabetes.

Age: (in years)

Outcome: Class Variable (0 or 1) where '0' denotes patient is not having diabetes and '1' denotes patient having diabetes.

The **dependent variable** is whether the patient is having diabetes or not.

DATA PREPARATION:

Data preparation stage includes data cleaning and transforming data if needed.

Various things have to be taken into consideration for data cleaning like:

Handling Zero/Null Values: The zeroes shown in the table are not zeroes but null values. We have deduced this based upon our inference that certain attributes like skin thickness, insulin, BMI etc. cannot be zero.

In sample data set there are lot of zero values. And the zero values have been replaced by the mean of that column.

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	\
0	6	148	72	35	0	33.6	
1	1	85	66	29	0	26.6	
2	8	183	64	0	0	23.3	
3	1	89	66	23	94	28.1	
4	0	137	40	35	168	43.1	

	DiabetesPedigreeFunction	Age	Outcome
0	0.627	50	1
1	0.351	31	0
2	0.672	32	1
3	0.167	21	0
4	2.288	33	1

Table.03. 04: Handling Zero/Null Values

Select appropriate attributes for analysis:

The dataset consists of 9 attributes i.e. **Pregnancies, Glucose, Blood Pressure, DiabetesPedigreeFunction,**

Age, Skin Thickness, Insulin, and BMI. These 8 are independent attributes and one i.e.**Outcome** is the dependent attribute. As all these attributes affect diabetes so we decided to keep all the independent variables for data mining process.

Data Splitting: Data was divided into training and testing data into 60:40 ratio. Sixty percent was training data and forty percent was testing data. Out of 498 records after data cleaning, 298 records were used for trained and 200 records were used for testing.

Different ranges were found out for each continuous variable in the data set. Based upon these ranges' categorization was done.

The features were categorized as per the below mentioned ranges and were denoted by 0, 1, 2 & 3, in order to use them for classification.

Glucose:

Plasma Glucose Test	Normal	Prediabetes	Diabetes
2 hour post-prandial	Below 140 mg/dl (0)	140 to 199 mg/dl (1)	200 mg/dl or more (2)

Table.03. 05: Plasma Glucose details

Blood Pressure (Diastolic):

Ranges	Low	Normal	High
	Below 60 (0)	60-90 (1)	90 or more (2)

Table.03. 06: Blood Pressure Details

Skin Thickness:

Ranges	Low	Normal	High
	<23 (0)	23 (1)	>23 (2)

Table.03. 07: Skin Thickness Details

Insulin:

Ranges	Low	Normal	High
2 Hours After Glucose	<16 mIU/L (0)	16-166 mIU/L (1)	>166 mIU/L (2)

Table.03. 08: Insulin Details

BMI (weight in kg/ height in m2):

Ranges	Under-weight	Normal	Over-weight	Obese
	<18.5 (0)	18.5-25 (1)	25-30 (2)	>30 (3)

Table.03. 09: BMI Details

Diabetes Pedigree Function:

	Low	Medium	High
	0-0.78 (0)	0.79-1.561 (1)	>1.57 (2)

Table.03. 10: Diabetes Pedigree Details

Age:

Ranges	Young	Adult	Old
	20-44 (0)	44-64 (1)	64-100 (2)

Table.03. 11:Age Details

No. of Pregnancies:

Ranges	Normal	Above Normal	Highest
	<6 (0)	6-12 (1)	>12 (2)

Table.03. 12: No.Of.Pregnancies Details

MODELING:

This phase includes application of appropriate model to the data.

Machine Learning Algorithms were used for modelling.

As we have to classify the data into patients having diabetes or not, we used Classification and Regression Tree Algorithm (CART) & K-Nearest Neighbor algorithms. Both of these algorithms are good for classifying dependent variables based upon categorized independent variables.

We compared both the algorithms to find the one which gives the best results based upon overall accuracy and precision.

Software Used: Python-Scikit Learn

Scikit-learn provides a range of supervised and unsupervised learning algorithms via a consistent interface in Python.

The library is built upon the SciPy (Scientific Python) that must be installed before we can use scikit-learn.

This stack includes:

NumPy: Base n-dimensional array package.

SciPy: Fundamental library for scientific computing.

Matplotlib: Comprehensive 2D/3D plotting.

I Python: Enhanced interactive console Symbolic mathematics Pandas: Data structures and analysis.

The file containing data set is loaded in pandas.

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	\
0	6	148.0	72.000000	35.000000	79.799479	33.600000	
1	1	85.0	66.000000	29.000000	79.799479	26.600000	
2	8	183.0	64.000000	20.536458	79.799479	23.300000	
3	1	89.0	66.000000	23.000000	94.000000	28.100000	
4	0	137.0	40.000000	35.000000	168.000000	43.100000	
5	5	116.0	74.000000	20.536458	79.799479	25.600000	
6	3	78.0	50.000000	32.000000	88.000000	31.000000	
7	10	115.0	69.105469	20.536458	79.799479	35.300000	
8	2	197.0	70.000000	45.000000	543.000000	30.500000	
9	8	125.0	96.000000	20.536458	79.799479	31.992578	

	DiabetesPedigreeFunction	Age	Outcome
0	0.627	50	1
1	0.351	31	0
2	0.672	32	1
3	0.167	21	0
4	2.288	33	1
5	0.201	30	0
6	0.248	26	1
7	0.134	29	0
8	0.158	53	1
9	0.232	54	1

Table.03. 13: Diabetes Dataset

CHAPTER 4
RESULTS AND DISSCUSSIONS

The potential of AI to enable diabetes solutions has been investigated in the context of multiple critical management issues. In this section, we use the following proposed diabetes management categories to summarize the latest contributions and the results described in the reviewed articles:

- Blood glucose control strategies
- Blood glucose prediction
- Detection of adverse glycemic events
- Insulin bolus calculators and advisory systems
- Risk and patient personalization
- Detection of meals, exercise and faults
- Lifestyle and daily-life support in diabetes management.

The performance of all the three algorithms is evaluated on various measures like **Precision, Accuracy, F-measure and Recall**.

Precision can be seen as a measure of exactness or quality, whereas **Recall** is a measure of completeness or quantity. In simple terms, high precision means that an algorithm returned substantially more relevant results than irrelevant ones, while high recall means that an algorithm returned most of the relevant results. **F1 Score** is used to measure a test's accuracy. F1 Score is the Harmonic Mean between precision and recall. The range for F1 Score is [0, 1]. It tells you how precise your classifier is (how many instances it classifies correctly), as well as how robust it is (it does not miss a significant number of instances). More commonly, **Accuracy** is a description of systematic errors as these cause a difference between a result and a "true" value.

4.1 KNN CLASIFICATION:

K nearest neighbors is a simple algorithm that stores all available cases and classifies new cases based on a similarity measure. KNN algorithm is one of the simplest classification algorithms. Even with such simplicity, it can give highly competitive results. KNN algorithm can also be used for regression problems. The only difference from the discussed methodology will be using averages of nearest neighbors rather than voting from nearest neighbors. It belongs to the supervised learning domain and finds intense application in pattern recognition, data mining and intrusion detection.

K-NN algorithm is arguably the simplest machine learning algorithm. Building the model consists only of storing the training data set. To make a prediction for a new data point, the algorithm finds

the closest data points in the training data set—its "nearest neighbors."

DATA:

The diabetes data set was originated from UCI Machine Learning Repository and can be downloaded from http://archive.ics.uci.edu/ml/index.php

The diabetes data set consists of 768 data points, with 9 features each.

In particular, all patients here are females at least 21 years old of Pima Indian heritage.

Number of Instances: 768

Number of Attributes: 9

Attributes

1. Number of times pregnant
2. Plasma glucose concentration 2 hours in an oral glucose tolerance test
3. Diastolic blood pressure (mm Hg)
4. Triceps skin fold thickness (mm)
5. 2-Hour serum insulin (mu U/ml)
6. Body mass index (weight in kg/ (height in m) ^2)
7. Diabetes pedigree function
8. Age (years)
9. Class variable (0 or 1)

Class Value	Number of instances
0	500
1	268

Table. 04.14: Graph representing growth of diabetes and AI

4.2 CODING PART:

STEP 1: READING DATASET FILE:

Index (['Pregnancies', 'Glucose', 'Blood Pressure', 'Skin Thickness', 'Insulin', 'BMI', 'DiabetesPedigreeFunction', 'Age', 'Outcome'], dtype='object')

```
from scipy import optimize
import pandas as pd
import numpy as np
from pandas import DataFrame
from sklearn.model_selection import train_test_split
from sklearn import metrics
from sklearn.tree import DecisionTreeClassifier
import matplotlib.pyplot as plt
%matplotlib inline

#step1 read the filein csv format
filename = 'diabetes.csv'
data = pd.read_csv(filename)
#print (data.shape)
print (data.describe())
```

Fig 04.02: Importing libraries and reading data set

RESULT 01:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	\
count	768.000000	768.000000	768.000000	768.000000	768.000000	
mean	3.845052	120.894531	69.105469	20.536458	79.799479	
std	3.369578	31.972618	19.355807	15.952218	115.244002	
min	0.000000	0.000000	0.000000	0.000000	0.000000	
25%	1.000000	99.000000	62.000000	0.000000	0.000000	
50%	3.000000	117.000000	72.000000	23.000000	30.500000	
75%	6.000000	140.250000	80.000000	32.000000	127.250000	
max	17.000000	199.000000	122.000000	99.000000	846.000000	

	BMI	DiabetesPedigreeFunction	Age	Outcome
count	768.000000	768.000000	768.000000	768.000000
mean	31.992578	0.471876	33.240885	0.348958
std	7.884160	0.331329	11.760232	0.476951
min	0.000000	0.078000	21.000000	0.000000
25%	27.300000	0.243750	24.000000	0.000000
50%	32.000000	0.372500	29.000000	0.000000
75%	36.600000	0.626250	41.000000	1.000000
max	67.100000	2.420000	81.000000	1.000000

Table 04.15: Importing libraries and reading data set

STEP 2: DATA VISUALIZATION

```
#DataVisualization

filt_df = df[['SkinThickness','Insulin']]
#filt_df = df[['Glucose','BloodPressure','SkinThickness','Insulin','BMI','DiabetesPedigreeFunction']]
#print(filt_df.head(10))
df.hist(figsize=(10,8))
```

Fig 04.03: Visualizing data

RESULT 02:

```
array([[<matplotlib.axes._subplots.AxesSubplot object at 0x12e5fd940>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x12f04abe0>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x12f0a12b0>],
       [<matplotlib.axes._subplots.AxesSubplot object at 0x12f10b278>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x12f15dd30>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x12f15dd68>],
       [<matplotlib.axes._subplots.AxesSubplot object at 0x12f2257f0>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x12f29dac8>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x12f2fc160>]], dtype=object)
```

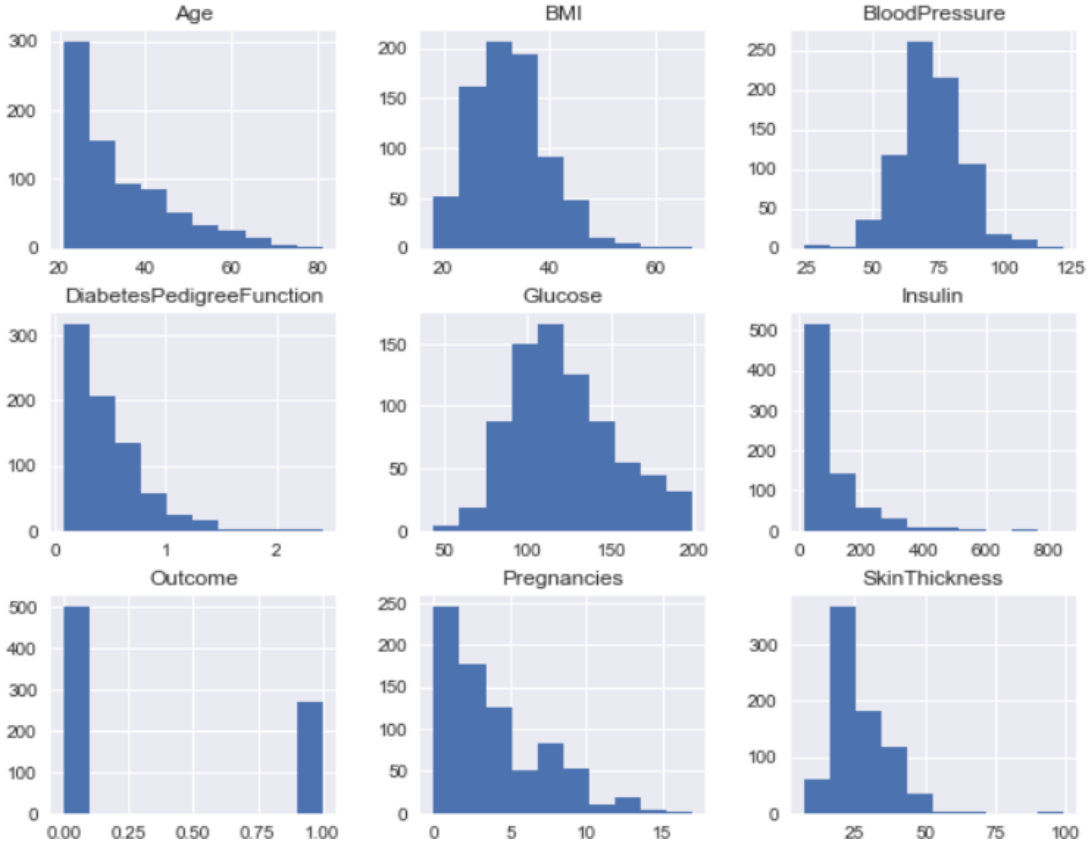

Fig 04.04: Graphs representing attributes

STEP 3:

```
df.plot(kind= 'box' , subplots=True, layout=(3,3), sharex=False, sharey=False, figsize=(10,8))
```

Fig 04.05: code showing subplots

RESULT 03:

```
Pregnancies                 AxesSubplot(0.125,0.657941;0.227941x0.222059)
Glucose                  AxesSubplot(0.398529,0.657941;0.227941x0.222059)
BloodPressure            AxesSubplot(0.672059,0.657941;0.227941x0.222059)
SkinThickness               AxesSubplot(0.125,0.391471;0.227941x0.222059)
Insulin                  AxesSubplot(0.398529,0.391471;0.227941x0.222059)
BMI                      AxesSubplot(0.672059,0.391471;0.227941x0.222059)
DiabetesPedigreeFunction       AxesSubplot(0.125,0.125;0.227941x0.222059)
Age                         AxesSubplot(0.398529,0.125;0.227941x0.222059)
Outcome                     AxesSubplot(0.672059,0.125;0.227941x0.222059)
dtype: object
```

Fig 04.06: Attributes axes subplot values

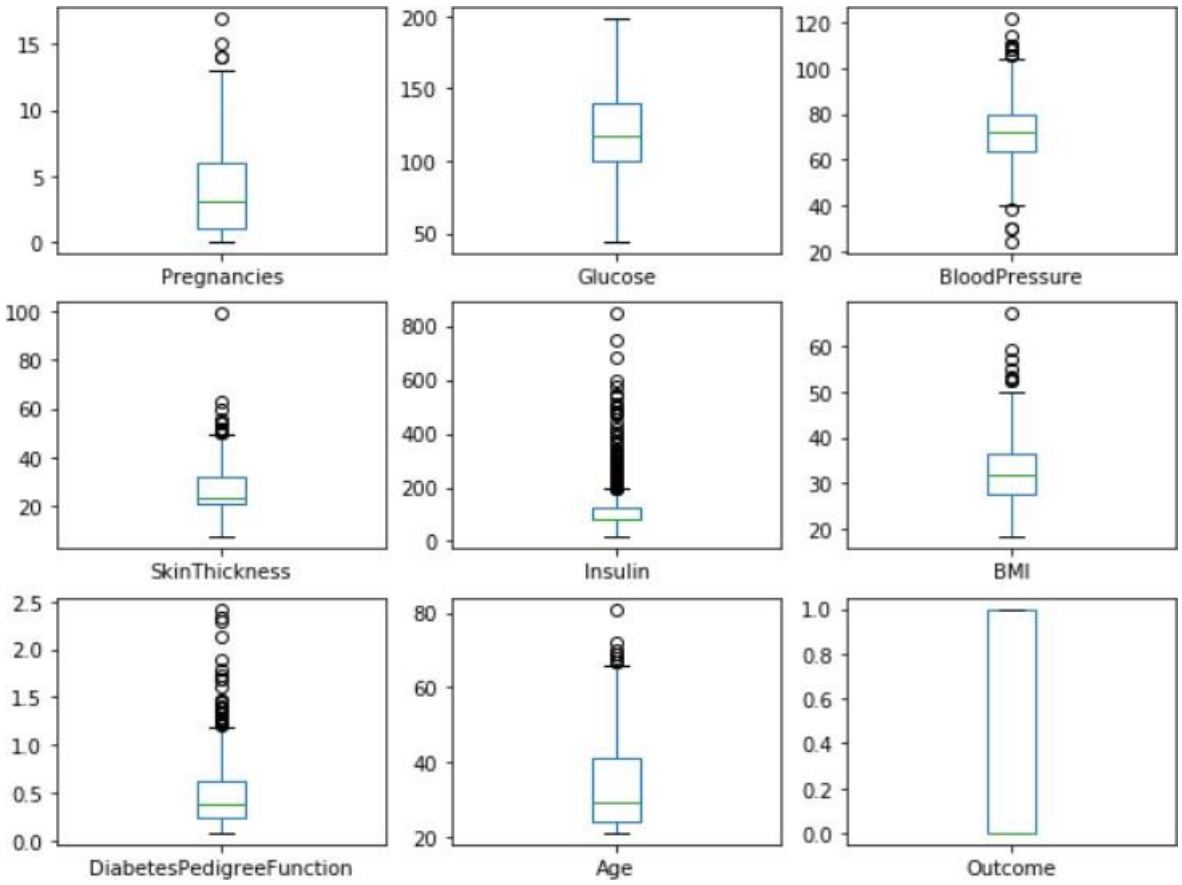

Fig 04.07: Subplots of attributes

STEP 4: DECISION TREE ACCURACY & FEATURE RANKING

```
#step4 use training dataset to apply algorithm
import seaborn as sns
model = DecisionTreeClassifier(max_depth=4, random_state=0)
tree_ = model.fit(feature,target)
test_input=test[test.columns[0:8]]
expected = test["Outcome"]
#print("******************Input*****************")
#print(test_input.head(2))
#print("******************Expected*****************")
#print(expected.head(2))
predicted = model.predict(test_input)
print(metrics.classification_report(expected, predicted))
conf = metrics.confusion_matrix(expected, predicted)
print(conf)
print("Decision Tree accuracy: ",model.score(test_input,expected))
dtreescore = model.score(test_input,expected)


label = ["0","1"]
sns.heatmap(conf, annot=True, xticklabels=label, yticklabels=label)
print (a)

#Feature Importance DecisionTreeClassifier
importance = model.feature_importances_
indices = np.argsort(importance)[::-1]
print("DecisionTree Feature ranking:")
for f in range(feature.shape[1]):
    print("%d. feature %s (%f)" % (f + 1, feat_names[indices[f]], importance[indices[f]]))
plt.figure(figsize=(15,5))
plt.title("DecisionTree Feature importances")
plt.bar(range(feature.shape[1]), importance[indices], color="y", align="center")
plt.xticks(range(feature.shape[1]), feat_names[indices])
plt.xlim([-1, feature.shape[1]])
plt.show()
```

Fig 04.08: Use of training data set to apply algorithm

RESULT 04:

```
              precision    recall  f1-score   support

           0       0.77      0.87      0.82       126
           1       0.72      0.57      0.64        74

avg / total       0.76      0.76      0.75       200

[[110  16]
 [ 32  42]]
Decision Tree accuracy:  0.76
DecisionTree Feature ranking:
1. feature Glucose (0.558787)
2. feature BMI (0.260081)
3. feature Pregnancies (0.134401)
4. feature Age (0.036539)
5. feature Insulin (0.010191)
6. feature DiabetesPedigreeFunction (0.000000)
7. feature SkinThickness (0.000000)
8. feature BloodPressure (0.000000)
```

Fig 04.09: Determining accuracy of decision tree and its feature ranking

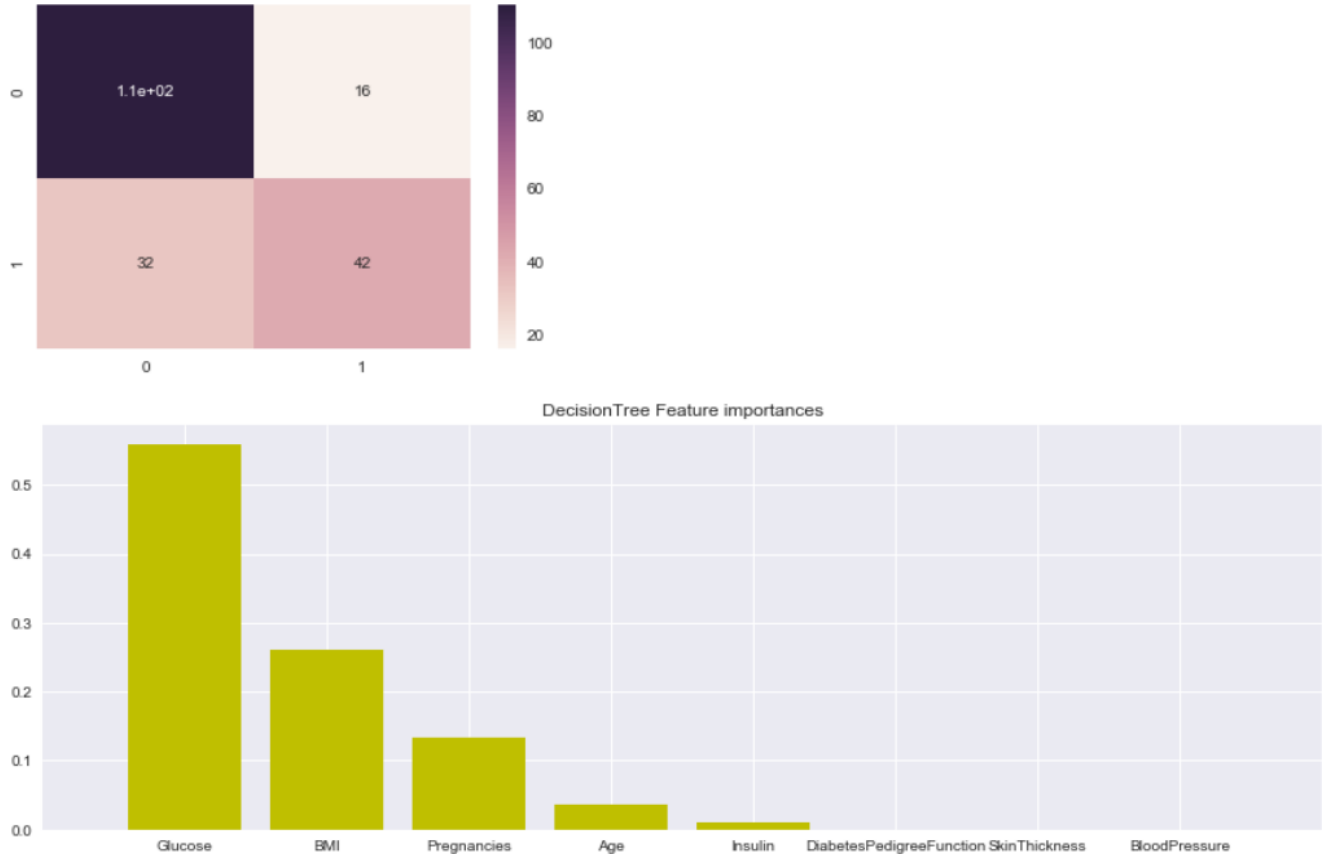

Fig 04.10: Feature importance of decision tree

Feature Importance With the help of decision tree, we were able to figure out which features played an important role. The above graph depicts the highest importance features.

STEP 5: KNN CLASSIFICATION

```
#KNN
from sklearn.neighbors import KNeighborsClassifier
neigh = KNeighborsClassifier(n_neighbors=21)
neigh.fit(feature,target)
knnpredicted = neigh.predict(test_input)
print(metrics.classification_report(expected, knnpredicted))
print(metrics.confusion_matrix(expected, knnpredicted))
print("KNN accuracy: ",neigh.score(test_input,expected))
knnscore=neigh.score(test_input,expected)
```

Fig 04.11: KNN classification

RESULT 05:

```
             precision    recall  f1-score   support

          0       0.75      0.82      0.78       126
          1       0.63      0.53      0.57        74

avg / total       0.70      0.71      0.70       200

[[103  23]
 [ 35  39]]
KNN accuracy:  0.71
```

Fig 04.12: Determining accuracy of KNN

STEP 6: DETERMINING ACCURACY:

```
names_ = []
results_ = []
results_.append(dtreescore)
results_.append(knnscore)
names_.append("DT")
names_.append("KNN")

#ax.set_xticklabels(names)
res = pd.DataFrame()
res['y']=results_
res['x']=names_
ax = sns.boxplot(x='x',y='y',data=res)
```

Fig 04.13: Accuracy of decision tree and KNN

RESULT 06:

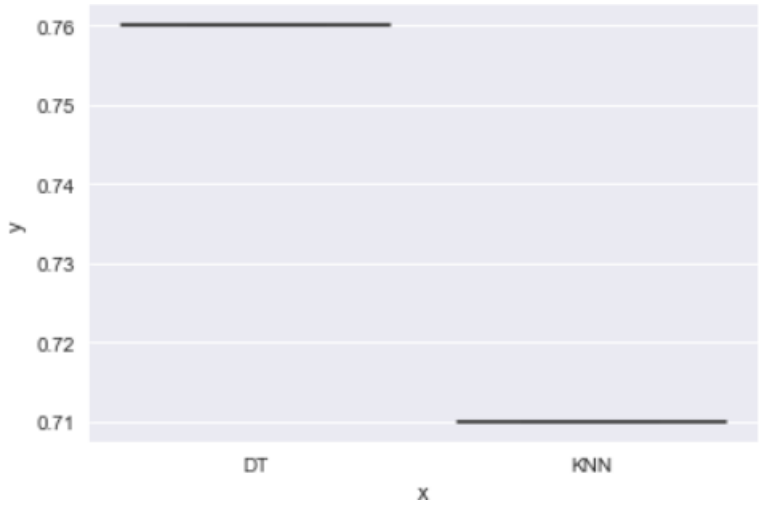

Fig 04.14: Representation of accuracy

Accuracy of K-NN classifier: 0.71

Accuracy of Decision tree: 0.76

STEP 7: FINAL DATA IMAGE REPRESENTATION:

```
import graphviz
import pydotplus
from IPython.display import Image
from sklearn.tree import export_graphviz
from sklearn.externals.six import StringIO
dot_data=StringIO()
dot_data = export_graphviz(model, out_file = None, feature_names=feat_names, class_names=target_classes,
                filled=True, rounded=True, special_characters=True)

graph = pydotplus.graph_from_dot_data(dot_data)
print(dot_data)
Image(graph.create_png())
#graph.write_pdf("diabetes.pdf")
```

Fig 04.15: Final data representation

RESULT 07 : The decision tree model i.e. CART is applied on the training dataset. The Decision Tree obtained gives the best result. The depth taken for this tree is 4 and total number of nodes are 21.

8 -> 20 ;

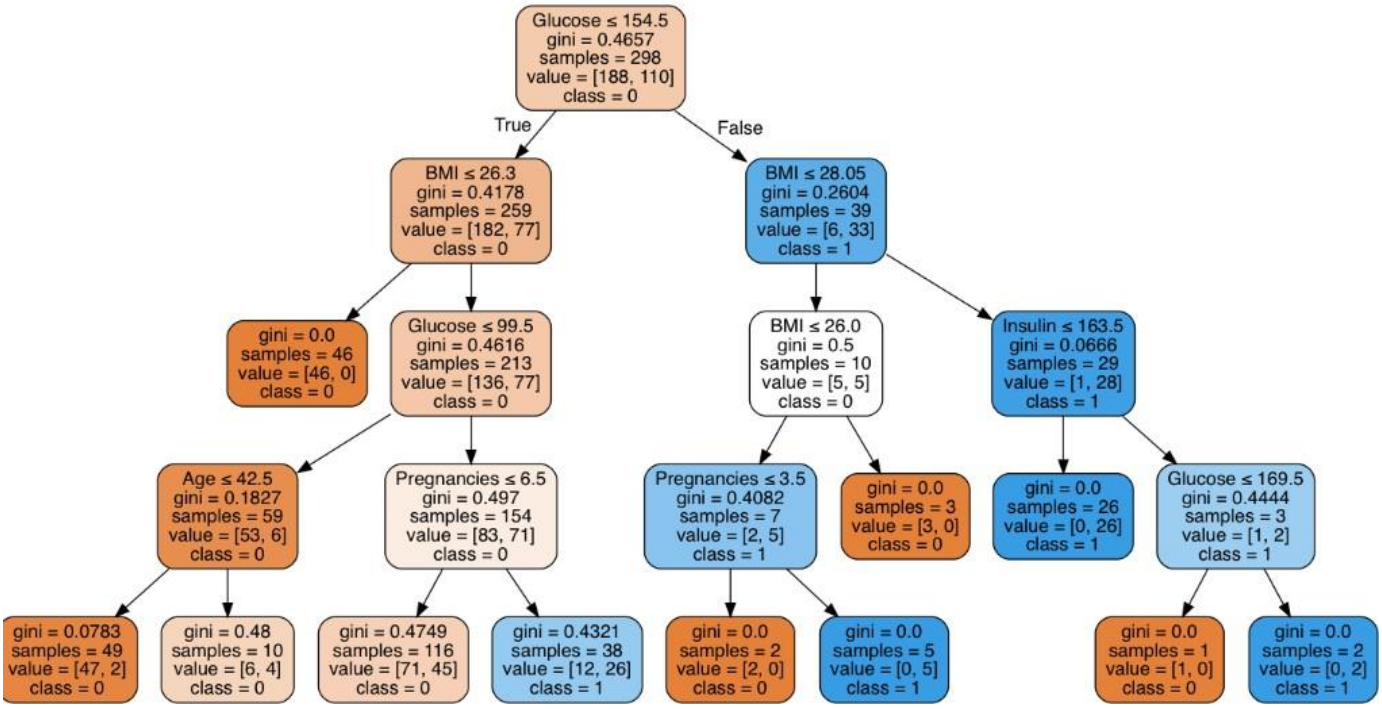

Fig 04.16: Final Decision tree

STEP 8: ROC CURVE FOR DECISION TREE

ROC stands for Receiver Operating Characteristic. In its current usage, **ROC** curves are a nice way to see how any predictive model can distinguish between the true positives and negatives.

```
#Evaluation DecisionTreeClassifier
from sklearn.metrics import roc_curve, auc
import random

fpr,tpr,thres = roc_curve(expected, predicted)
roc_auc = auc(fpr, tpr)
plt.title('DecisionTreeClassifier-Receiver Operating Characteristic Test Data')
plt.plot(fpr, tpr, color='green', lw=2, label='DecisionTree ROC curve (area = %0.2f)' % roc_auc)
plt.legend(loc='lower right')
plt.plot([0,1],[0,1],'r--')
plt.xlim([-0.1,1.2])
plt.ylim([-0.1,1.2])
plt.ylabel('True Positive Rate')
plt.xlabel('False Positive Rate')
plt.show()
```

Fig 04.17: Evaluation of decision tree classifier

RESULT 08: The Area Under Curve (AUC) for Decision Tree is 0.72.

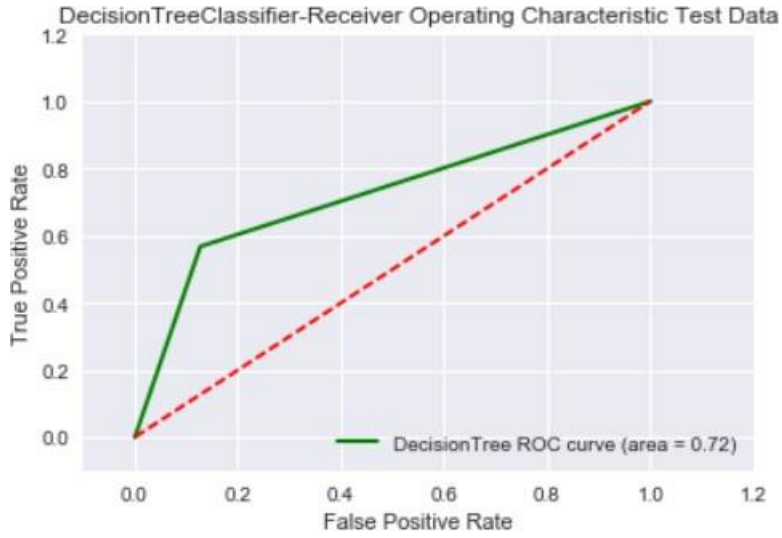

Fig 04.18: Representing AUC for decision tree

STEP 9: ROC CURVE FOR KNN CLASSIFIER

```
#KNeighborsClassifier-ROC curve
kfpr,ktpr,kthres = roc_curve(expected, knnpredicted)
kroc_auc = auc(kfpr, ktpr)
plt.title('KNeighborsClassifier- Receiver Operating Characteristic')
plt.plot(kfpr, ktpr, color='darkorange', lw=2, label='KNeighbors ROC curve (area = %0.2f)' % kroc_auc)
plt.legend(loc='lower right')
plt.plot([0,1],[0,1],'r--')
plt.xlim([-0.1,1.2])
plt.ylim([-0.1,1.2])
plt.ylabel('True Positive Rate')
plt.xlabel('False Positive Rate')
plt.show()
```

Fig 04.19: Evaluation of decision tree classifier

RESULT 09: The Area Under Curve (AUC) for KNN is 0.67.

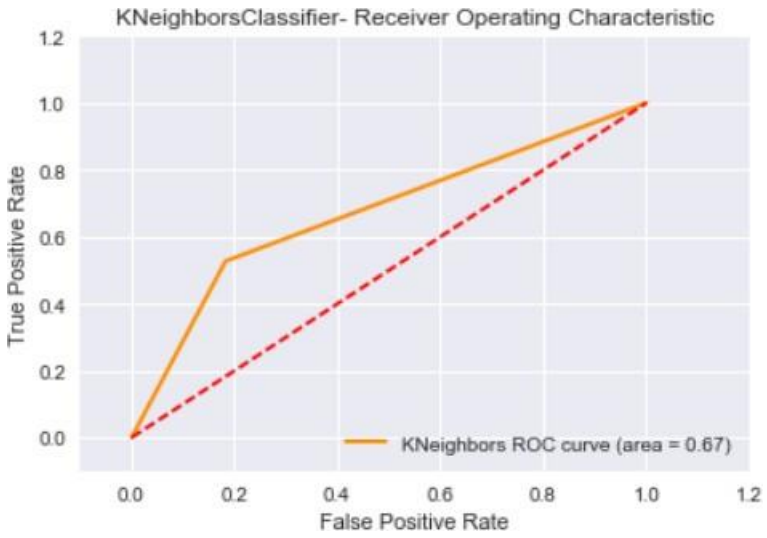

Fig 04.20: AUC for KNN

Decision Rules:

S.No.	Antecedent	Consequent	Support	Confidence
1.	If Glucose<= 154.5 & BMI <= 26.3	No Diabetes	46/298	1
2.	If Glucose<= 154.5 & BMI <= 26.3 & Glucose <= 99.5 & Age <= 42.5	No Diabetes	47/298	0.95
3.	If Glucose<= 154.5 & BMI <= 26.3 & Glucose <= 99.5 & Age <= 42.5	No Diabetes	6/298	0.60
4.	If Glucose<= 154.5 & BMI <= 26.3 & Glucose <= 99.5 & Pregnancies<= 6.5	No Diabetes	71/298	0.61
5.	If Glucose<= 154.5 & BMI <= 26.3 & Glucose <= 99.5 & Pregnancies<= 6.5	Diabetes	26/298	0.68
6.	If Glucose<= 154.5 & BMI <= 28.05 & BMI <= 26.00	No Diabetes	3/298	1
7.	If Glucose<= 154.5 & BMI <= 28.05 & BMI <= 26.00 & Pregnancies <= 3.5	No Diabetes	2/298	1
8.	If Glucose<= 154.5 & BMI <= 28.05 & BMI <= 26.00 & Pregnancies <= 3.5	Diabetes	5/298	1
9.	If Glucose<= 154.5 & BMI <= 28.05 & Insulin <= 163.5	Diabetes	26/298	1
10.	If Glucose<= 154.5 & BMI <= 28.05 & Insulin <= 163.5 & Glucose <= 169.5	No Diabetes	1/298	1
11.	If Glucose<= 154.5 & BMI <= 28.05 & Insulin <= 163.5 & Glucose <= 169.5	Diabetes	2/298	1

Table 04.16: Decision rules

4.3 EVALUATION:

The graph is called a Receiver Operating Characteristic curve (or ROC curve.) It is a plot of the true positive rate against the false positive rate for the different points.

▪ It shows the tradeoff between True Positive Rate (TPR) and False Positive Rate (FPR) (any increase in (TPR) will be accompanied by a decrease in (FPR)).

▪ The closer the curve towards TPR of the ROC space, the more accurate the test.

▪ The closer the curve towards the FPR of the ROC space, the less accurate the test.

▪ The Area Under the Curve (AUC) is a measure to determine the model effectiveness. The more AUC for a model comes to 1, the better it is. So, models with higher AUCs are preferred over those with lower AUCs.

CONFUSION MATRIX:

	Predicted Yes	**Predicted No**
Actual Yes	tp (true positive)	fp (false positive)
Actual No	fn (false negative)	tn (true negative)

Table 04.17: Confusion matrix

Confusion Matrix for the Decision Tree:

	Predicted Diabetes	**Predicted No Diabetes**
Actual Diabetes	**110**	**16**
Actual No Diabetes	**32**	**42**

Table 04.18: Confusion matrix

•**true positives (TP):** These are cases in which we correctly predicted diabetes as result.

•**true negatives (TN):** We correctly predicted no diabetes and they don't have the disease.

•**false positives (FP):** We correctly predicted no diabetes, but they actually had the disease. (Also known as a "Type I error.")

•**false negatives (FN):** We correctly predicted diabetes, but they actually had no disease. (Also known as a "Type II error.")

```
             precision    recall  f1-score   support

          0       0.77      0.87      0.82       126
          1       0.72      0.57      0.64        74

avg / total       0.76      0.76      0.75       200

[[110  16]
 [ 32  42]]
Decision Tree accuracy:  0.76
```

Fig 04.21: Accuracy of decision tree

The Accuracy for the Decision Tree is 76%

ALTERNATE MODEL COMPARISION:

We tried another machine learning algorithm to compare the accuracy of the model. We used K-Nearest Neighbor algorithm. The results obtained are as under:

KNN:

	Predicted Diabetes	Predicted No Diabetes
Actual Diabetes	103	23
Actual No Diabetes	35	39

Table 04.19: Results for KNN

```
             precision    recall  f1-score   support

          0       0.75      0.82      0.78       126
          1       0.63      0.53      0.57        74

avg / total       0.70      0.71      0.70       200

[[103  23]
 [ 35  39]]
KNN accuracy:  0.71
```

Fig 04.22: Accuracy of KNN

The Accuracy of KNN Model is 71%

Decision Tree:

	Predicted Diabetes	Predicted No Diabetes
Actual Diabetes	110	16
Actual No Diabetes	32	42

Table 04.20: Results for decision tree

```
             precision    recall  f1-score   support

          0       0.77      0.87      0.82       126
          1       0.72      0.57      0.64        74

avg / total       0.76      0.76      0.75       200

[[110  16]
 [ 32  42]]
Decision Tree accuracy:  0.76
```

Fig 04.23.Decision tree accuracy

The Accuracy of Decision Tree is 76%.

So, by comparing both the results we infer that Decision Tree Model is showing better results in comparison to K-Nearest Neighbour Model**.**

Plan Deployment: Planning basically includes the strategy to be formulated for implementing the model in real world.

This model can now be used in medical organizations for easy and early detection of diabetes in patients.

GOAL: Goal of this project is to identify the probability of diabetes in patients using data mining techniques.

CHAPTER 5

CONCLUSION AND FUTURESCOPE:

One of the important real-world medical problems is the detection of diabetes at its early stage. In this study, systematic efforts are made in designing a system which results in the prediction of disease like diabetes. During this work, three machine learning classification algorithms are studied and evaluated on various measures. Experiments are performed on Pima Indians Diabetes Database. Experimental results determine the adequacy of the designed system with an achieved accuracy of 76% using the Decision tree classification algorithm. In future, the designed system with the used machine learning classification algorithms can be used to predict or diagnose other diseases. The work can be extended and improved for the automation of diabetes analysis including some other machine learning algorithms.

Artificial Intelligence (AI) is the simulation of human intelligence by machines. In other words, it is the method of machines demonstrating certain aspects of human intelligence like self- correction and learning and reasoning. Of its inception, AI has demonstrated unprecedented growth. Sophia the AI Robot, is the quintessential example of this. The future of Artificial intelligence is hazy. But by going the bounds of progress, AI is being making, it is clear that AI will permeate every sphere of your life.
While machine learning emphasizes on making predictions about the future, artificial intelligence typically concentrates on programming computers to make decisions. If you use an intelligent program that involves human-like behaviour, it can be artificial intelligence. However, if the parameters are not automatically learned (or derived) from data, it’s not machine learning.

FUTURE SCOPE:

We obtained evidence of an acceleration of research activity aimed at developing artificial intelligence-powered tools for prediction and prevention of complications associated with diabetes. Our results indicate that artificial intelligence methods are being progressively established as suitable for use in clinical daily practice, as well as for the self-management of diabetes. Consequently, these methods provide powerful tools for improving patients quality of life.

Advantage of this project the rules derived will be helpful for doctors to identify patients suffering from diabetes. Further predicting the disease early leads to treating the patient before it becomes critical.

BIBLIOGRAPHY

1. Aishwarya, R., Gayathri, P., Jaisankar, N., 2013. A Method for Classification Using Machine Learning Technique for Diabetes. International Journal of Engineering and Technology (IJET) 5, 2903–2908.

2. Aljumah, A.A., Ahamad, M.G., Siddiqui, M.K., 2013. Application of data mining: Diabetes health care in young and old patients. Journal of King Saud University - Computer and Information Sciences 25, 127–136. Doi: 10.1016/j.jksuci.2012.10.003.

3. Arora, R., Suman, 2012. Comparative Analysis of Classification Algorithms on Different Datasets using WEKA. International Journal of Computer Applications 54, 21–25. Doi: 10.5120/8626-2492.

4. Bamnote, M.P., G.R., 2014. Design of Classifier for Detection of Diabetes Mellitus Using Genetic Programming. Advances in Intelligent Systems and Computing 1, 763–770. Doi: 10.1007/978-3-319-11933-

5. Choubey, D.K., Paul, S., Kumar, S., Kumar, S., 2017. Classification of Pima indian diabetes dataset using naive bayes with genetic algorithm as an attribute selection, in: Communication and Computing Systems: Proceedings of the International Conference on Communication and Computing System (ICCCS 2016), pp. 451–455.

6. Dhomse Kanchan B., M.K.M., 2016. Study of Machine Learning Algorithms for Special Disease Prediction using Principal of Component Analysis, in: 2016 International Conference on Global Trends in Signal Processing, Information Computing and Communication, IEEE. pp. 5–10.

7. Esposito, F., Malerba, D., Semeraro, G., Kay, J., 1997. A comparative analysis of methods for pruning decision trees. IEEE Transactions on Pattern Analysis and Machine Intelligence 19, 476–491. doi:10.1109/34.589207.

8. Fatima, M., Pasha, M., 2017. Survey of Machine Learning Algorithms for Disease Diagnostic. Journal of Intelligent Learning Systems and Applications 09, 1–16. doi:10.4236/jilsa.2017.91001.

9. Garner,S.R.,1995.Weka: The Waikato Environment for Knowledge Analysis ,in :Proceedings of the NewZealand computer science research students conference, Citeseer. pp. 57–64.

10. Han, J., Rodriguez, J.C., Beheshti, M., 2008. Discovering decision tree-based diabetes prediction model, in: International Conference on Advanced Software Engineering and Its Applications, Springer. pp. 99–109.

11. Iyer, A., S, J., Sumbaly, R., 2015. Diagnosis of Diabetes Using Classification Mining Techniques.

International Journal of Data Mining & Knowledge Management Process 5, 1–14. doi:10.5121/ijdkp.2015.5101, arXiv:1502.03774.

12. Kavakiotis, I., Tsave, O., Salifoglou, A., Maglaveras, N., Vlahavas, I., Chouvarda, I., 2017. Machine Learning and Data Mining Methods in Diabetes Research. Computational and Structural Biotechnology Journal 15, 104–116. Doi: 10.1016/j.csbj.2016.12.005.

13. Kayaer, K., Tulay, 2003. Medical diagnosis on Pima Indian diabetes using general regression neural networks, in: Proceedings of the international conference on artificial neural networks and neural information processing (ICANN/ICONIP), pp. 181–184.

APPENDIX-A

Source Code

```
from scipy import optimize
import pandas as pd
import numpy as np
from pandas import DataFrame
from sklearn.model_selection import train_test_split
from sklearn import metrics
from sklearn import tree
import matplotlib.pyplot as plt
import seaborn as sb
%matplotlib inline

#read the file in csv format
from google.colab import files
uploaded = files.upload()
data=pd.read_csv("diabetes.csv")
print(data.shape)
print(data.describe())

#DataVisualization
# filt_df = data[['SkinThickness','Insulin']]
filt_df = data[['Glucose','BloodPressure','SkinThickness','Insulin','BMI','DiabetesPe
digreeFunction']]
print(filt_df.head(10))
data.hist(figsize=(10,8))
data.plot(kind= 'box',subplots=True,layout=(3,3),sharex=False,sharey=False,figsize=(1
0,8))
sb.pairplot(data, hue="Outcome")
features = list(data.columns[:8])
y = data['Outcome']
x = data[features]
featureTrain , featureTest, labelTrain, labelTest = train_test_split(x,y)
#use training dataset to apply algorithm
Tree = tree.DecisionTreeClassifier(max_depth=4,random_state = 0
                                   )
Tree = Tree.fit(featureTrain,labelTrain)
predicted=Tree.predict(featureTest)
print(metrics.classification_report(labelTest,predicted))
conf=metrics.confusion_matrix(labelTest,predicted)
print(conf)
dTreeScore=Tree.score(featureTest,labelTest)
print("Decision Tree accuracy: ",dTreeScore)

label=["0","1"]
sb.heatmap(conf,annot=True,xticklabels=label,yticklabels=label)
```

```
#Feature Importance DecisionTreeClassifier
importance = Tree.feature_importances_
indices = np.argsort(importance)[::-1]
print("DecisionTree Feature ranking:")
for f in range(len(features)):
    print("%d. feature %s (%f)" %(f+1,features[indices[f]],importance[indices[f]]))

# plt.figure(figure=(15,5))
plt.title("DecisionTree Feature importances")
plt.barh(features,importance[indices],align="center")
plt.show()

#KNN
from sklearn.neighbors import KNeighborsClassifier
neigh=KNeighborsClassifier(n_neighbors=21)
neigh.fit(featureTrain,labelTrain)
knnpredicted=neigh.predict(featureTest)
print(metrics.classification_report(labelTest,knnpredicted))
print(metrics.confusion_matrix(labelTest,knnpredicted))
knnScore = neigh.score(featureTest,labelTest)
print("KNN accuracy : ",knnScore)
label = y.unique()
sb.heatmap(conf,annot=True,xticklabels=label,yticklabels=label)
names_=[]
results_=[]
results_.append(dTreeScore)
results_.append(knnScore)
names_.append("DT")
names_.append("KNN")

#ax.set_xticklabels(names)
res=pd.DataFrame()
res['y']=results_
res['x']=names_
ax=sb.boxplot(x='x',y='y',data=res)
import graphviz
import pydotplus
from IPython.display import Image
from sklearn.tree import export_graphviz
from sklearn.externals.six import StringIO
target_classes = ["0","1"]
dot_data=StringIO()
dot_data=export_graphviz(Tree,out_file=None,feature_names=features,class_names = targ
et_classes,
                         filled=True,rounded=True,special_characters=True)
graph=pydotplus.graph_from_dot_data(dot_data)
print(dot_data)
Image(graph.create_png())
```

```
#Evaluation DecisionTreeClassifier
from sklearn.metrics import roc_curve,auc
import random
fpr,tpr,thres=roc_curve(labelTest,predicted)
roc_auc=auc(fpr,tpr)
plt.title('DecisionTreeClassifier-Receiver Operating Characteristic Test Data')
plt.plot(fpr,tpr,color='green',lw=2,label='DecisionTree ROC curve(area=%0.2f)'%roc_au
c)
plt.legend(loc='lower right')
plt.plot([0,1],[0,1],'r--')
plt.xlim([-0.1,1.2])
plt.ylim([-0.1,1.2])
plt.ylabel('True Positive Rate')
plt.xlabel('False Positive Rate')
plt.show()
#KNeighborsClassifier-ROC curve
kfpr,ktpr,kthres=roc_curve(labelTest,knnpredicted)
kroc_auc=auc(kfpr,ktpr)
plt.title('KNeighborsClassifier-Receiver Operating Characteristic')
plt.plot(kfpr,ktpr,color='darkorange',lw=2,label='KNeighbors ROC curve(area=%0.2f)'%k
roc_auc)
plt.legend(loc='lower right')
plt.plot([0,1],[0,1],'r--')
plt.xlim([-0.1,1.2])
plt.ylim([-0.1,1.2])
plt.ylabel('True Positive Rate')
plt.xlabel('False Positive Rate')
plt.show()
```