DIABETICS PREDICTION SYSTEM USING MACHINE LEARNING

Dissertation submitted in partial fulfillment of the requirements for the award of the degree of

MASTER OF COMPUTER APPLICATIONS

BY

K.HARITHA

17VV1F0014

Under the esteemed guidance of

MR. W.ANIL , M.Tech.

Assistant Professor
Department of Information Technology
JNTUK-UCEV

DEPARTMENT OF INFORMATION TECHNOLOGY

JNTUK-UNIVERSITY COLLEGE OF ENGINEERING VIZIANAGARAM

VIZIANAGARAM-535003, ANDHRA PRADESH, INDIA

2019-2020

DEPARTMENT OF INFORMATION TECHNOLOGY

JNTUK-UNIVERSITY COLLEGE OF ENGINEERING VIZIANAGARAM

CERTIFICATE

This is to certify that the project report entitled "DIABETICS PREDICTION SYSTEM USING MACHINE LEARNING" that is being submitted by K.HARITHA (17VV1F0014) in partial fulfillment for the award of *Degree of Master of Computer Applications* in the department of INFORMATION TECHNOLOGY from Jawaharlal Nehru Technological University Kakinada - University College of Engineering Vizianagaram is a record of bonafide work carried out by her under my guidance and supervision in this department during the year 2019-2020.

The results embodied in this report have not been submitted to any other University or Institute for the award of any degree or diploma.

Signature of Guide
MR. W. ANIL
Assistant Professor
Dept. of Information Technology
JNTUK – UCEV

Signature of Head of the Department
Prof. G. JAYA SUMA
Professor & Head
Dept. of Information Technology
JNTUK – UCEV

(External Examiner)

DECLARATION

I, declare that this written submission represents my ideas in my own words and where others ideas or words have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea data fact source in my submission .I understand that any violation of the above will be cause for disciplinary action by the institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

(Signature)

K.HARITHA (17VV1F0014)

(Name of the student)

Place:

Date:

ACKNOWLEDGEMENT

The acknowledgement transcends the reality of formality when I express deep gratitude and respect to all those people behind the screen who inspired and helped us in completion of this project work.

I take privilege to express my heartfelt gratitude to project supervisor Mr**. W. ANIL,** Assistant Professor, Information Technology, JNTUK UCEV for his valuable suggestions and constant motivation that greatly helped me in successful completion of the project.

I express my sincere thanks to Prof**. G. JAYA SUMA**, Head of the Department of Information Technology, JNTUK UCEV for her continuous support.

I express my sincere thanks to Prof. **G. Swami Naidu** , Principal of JNTUK University College of Engineering Vizianagaram, for providing me with a good infrastructure and environment to carry out this project.

I am thankful to all Teaching and Non-Teaching staff of Information Technology Department, JNTUK University College of Engineering Vizianagaram for their direct and indirect help provided to me in completing the project.

Finally, I am extremely thankful to my parents and friends for their constant help and moral support.

K.HARITHA.

17VV1F0014

TABLE OF CONTENTS

ABSTRACT

Diabetes mellitus or simply diabetes is a disease caused due to the increase level of blood glucose.Diabetes is a chronic disease with the potential to cause a worldwide health care crisis.According to International Diabete Federation 382 million people are living with diabetes across the whole world. By 2035, this will be doubled as 592 million.

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 .

This project aims to predict diabetes via different supervised machine learning methods. This project is to develop a system which can perform early prediction of diabetes for a patient with a higher accuracy by different machine learning techniques.This project also aims to propose an effective technique for earlier detection of the diabetes disease.

LIST OF FIGURES

LIST OF TABELS

INTRODUCTION

1. INTRODUCTION

1.1 Introduction to Diabetics :

In understanding diabetes and how it develops, we need to understand what happens in the body without diabetes. Sugar (glucose) comes from the foods that we eat , specifically carbohydrate foods . Carbohydrate foods provide our body with its main energy source – everybody,even those people with diabetes,needs carbohydrate.Carbohydrate foods include bread,cereal,pasta, rice, fruit, dairy products and vegetables (especially starchy vegetables). When we eat these foods,the body breaks them down into glucose . The glucose moves around the body in the bloodstream . Some of the glucose is taken to our brain to help us think clearly and function. The remainder of the glucose is taken to the cells of our body for energy and also to our liver,where it is stored as energy that is used later by the body.In order for the body to use glucose for energy,insulin is required.

Insulin is a hormone that is produced by the beta cells in the pancreas. Insulin works like a key to a door. Insulin attaches itself to ‘doors’ on the cell, opening the door to allow glucose to move from the blood stream, through the door, and into the cell. If the pancreas is not able to produce enough insulin (insulin deficiency) or if the body can - not use the insulin it produces (insulin resistance), glucose builds up in the bloodstream (hyperglycemia) and diabetes develops. Diabetes Mellitus means high levels of sugar (glucose) in the blood stream and in the urine .

Signs or Symptoms of Diabetes:

- ➢ Frequent Urination
- ➢ Increased thirst
- ➢ Increased hunger
- ➢ Tired/Sleepiness
- ➢ Weightloss
- ➢ Blurred vision
- ➢ Mood swings
- ➢ Confusion and difficulty concentrating
- ➢ frequent infections / poor healing

Diabetes Mellitus :

Diabetes Mellitus(DM)can be divided into several distinct types.
However, there are two major clinical types,

- type 1 diabetes (T1D)
- type 2 diabetes (T2D)

According to the etiopathology of the disorder. T2D appears to be the most common form of diabetes (90% of all diabetic patients),mainly characterized by insulin resistance.The main causes of T2D include lifestyle, physical activity, dietary habits and heredity .

whereas T1D is thought to be due to auto immunological destruction of the Langerhans islets hosting pancreatic-β cells. T1D affects almost 10% of all diabetic patients worldwide, with 10% of them ultimately developing idiopathic diabetes. Other forms of DM, classified on the basis of insulin secretion profile , include Gestational Diabetes, endocrinopathies, MODY (Maturity Onset Diabetes of the Young), neonatal,mitochondrial, and pregnancy diabetes. The symptoms of DM include polyuria, polydipsia, and significant weight loss among others. Diagnosis depends on blood glucose levels (fasting plasmaglucose = 7.0 mmol/L.

Effects of diabetes

Diabetes is influenced by different parts of the body which incorporates.

- *Loss of vision*
- *Kidney neuropathy*
- *Liver problems*
- *Heart problems* Cardiovascular ailment
- The different issues may incorporate foot issues

1.2 Introduction to machineLearning :

What is MachineLearning :

Machine Learning is said as a subset of **artificial intelligence** that is mainly concerned with the development of algorithms which allow a computer to learn from the data and past experiences on their own. The term machine learning was first introduced by **Arthur Samuel** in **1959**. We can define it in a summarized way as:

> Machine learning enables a machine to automatically learn from data, improve performance from experiences, and predict things without being explicitly programmed.

Let's take some real-life examples to understand :-

- ➢ Google's self-driving car. A car that drives by itself without any human support
- ➢ virtual personal assistants such as Apple's Siri or Microsoft's Cortana? If you ask Siri what is the distance between Earth and Moon, it will immediately reply that the distance is 384,400km.
- ➢ Google maps. If you want to go from New Jersey to New York via road, google maps will show you the distance between these two places, the shortest route and also how much traffic is there along the road.

How does Machine Learning work :

A Machine Learning system "learns from historical data, builds the prediction models, and whenever it receives new data, predicts the output for it". The accuracy of predicted output depends upon the amount of data, as the huge amount of data helps to build a better model which predicts the output more accurately.

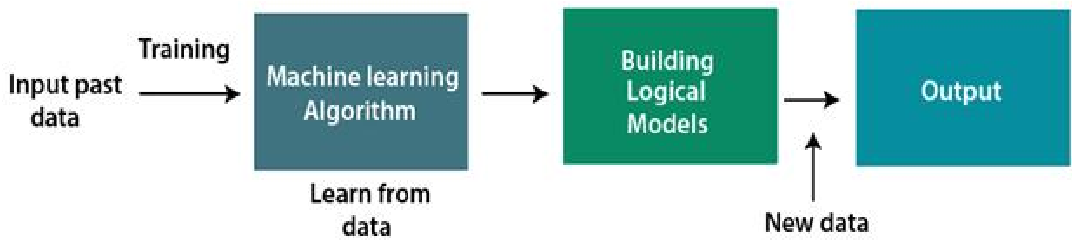


Fig 1 : workflow of machine learning

Classification of MachineLearning

At a broad level, machine learning can be classified into three types:

1. Supervised learning
2. Unsupervised learning
3. Reinforcement learning

Supervised Learning :- Supervised learning is a type of machine learning method in which we provide sample labeled data to the machine learning system in order to train it, and on that basis, it predicts the output.

The system creates a model using labeled data to understand the datasets and learn about each data, once the training and processing are done then we test the model by providing a sample data to check whether it is predicting the exact output or not.

The goal of supervised learning is to map input data with the output data. The supervised learning is based on supervision, and it is the same as when a student learns things in the supervision of the teacher.

Unsupervised learning: Unsupervised learning is a learning method in which a machine learns without any supervision.

The training is provided to the machine with the set of data that has not been labeled, classified, or categorized, and the algorithm needs to act on that data without any supervision. The goal of unsupervised learning is to restructure the input data into new features or a group of objects with similar patterns.

Reinforcement learning : Reinforcement learning is a feedback-based learning method, in which a learning agent gets a reward for each right action and gets a penalty for each wrong action. The agent learns automatically with these feedbacks and improves its performance. In reinforcement learning, the agent interacts with the environment and explores it. The goal of an agent is to get the most reward points, and hence, it improves its performance

1.3 Introduction to Classification

The Supervised Machine Learning algorithm can be broadly classified into Regression and Classification Algorithms. In Regression algorithms, we have predicted the output for continuous values, but to predict the categorical values, we need Classification algorithms.

What is the Classification Algorithm?

The Classification algorithm is a Supervised Learning technique that is used to identify the category of new observations on the basis of training data. In Classification, a program learns from the given dataset or observations and then classifies new observation into a number of classes or groups. Such as, **Yes or No, 0 or 1, Spam or Not Spam, cat or dog,** etc. Classes can be called as targets/labels or categories.

Unlike regression, the output variable of Classification is a category, not a value, such as "Green or Blue", "fruit or animal", etc. Since the Classification algorithm is a Supervised learning technique, hence it takes labeled input data, which means it contains input with the corresponding output.

In classification algorithm, a discrete output function(y) is mapped to input variable(x).

y=f(x), where y = categorical output

The best example of an ML classification algorithm is **Email Spam Detector**.

The main goal of the Classification algorithm is to identify the category of a given dataset, and these algorithms are mainly used to predict the output for the categorical data.

Classification algorithms can be better understood using the below diagram. In the below diagram, there are two classes, class A and Class B. These classes have features that are similar to each other and dissimilar to other classes.

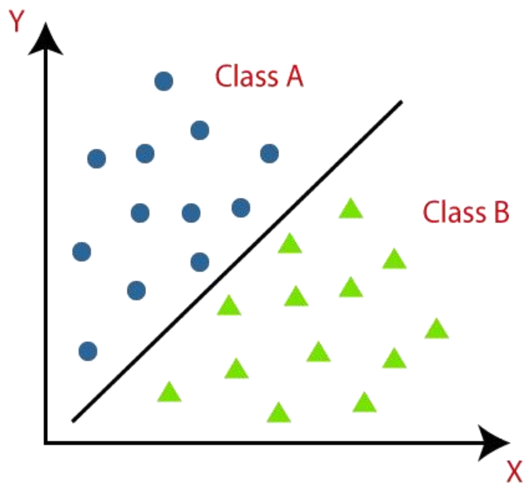


Fig 2 : classification

Classifiers :

The algorithm which implements the classification on a dataset is known as a classifier. There are two types of Classifiers:

- **Binary Classifier:** If the classification problem has only two possible outcomes, then it is called as Binary Classifier.
  Examples: YES or NO, MALE or FEMALE, SPAM or NOT SPAM, CAT or DOG, etc.
- **Multi-class Classifier:** If a classification problem has more than two outcomes, then it is called as Multi-class Classifier.
  Example: Classifications of types of crops, Classification of types of music

1.4 Classification Models :

1.4.1 Logistic Regression :

Logistic Regression is a Machine Learning algorithm which is used for the classification problems, it is a predictive analysis algorithm and based on the concept of probability.

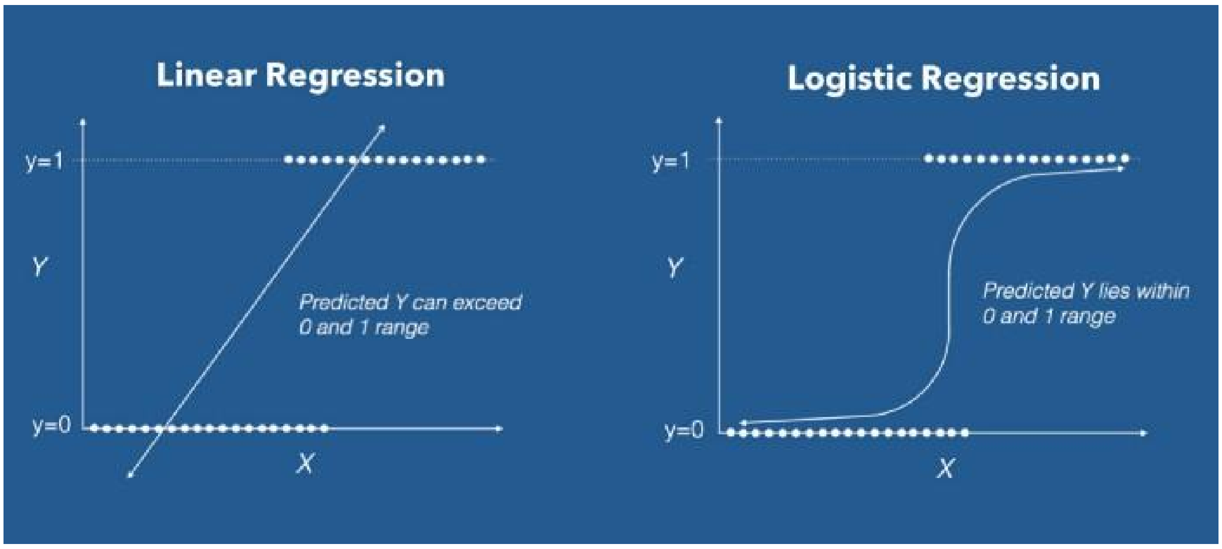


Fig 3 : Logistic Regression

Logistic Regression uses a more complex cost function, this cost function can be defined as the '**Sigmoid function**' or also known as the 'logistic function' instead of a linear function.

The hypothesis of logistic regression tends it to limit the cost function between 0 and 1. Therefore linear functions fail to represent it as it can have a value greater than 1 or less than 0 which is not possible as per the hypothesis of logistic regression.

$$0 \leq h_\theta(x) \leq 1$$

What is the Sigmoid Function?

In order to map predicted values to probabilities, we use the Sigmoid function. The function maps any real value into another value between 0 and 1. In machine learning, we use sigmoid to map predictions to probabilities.

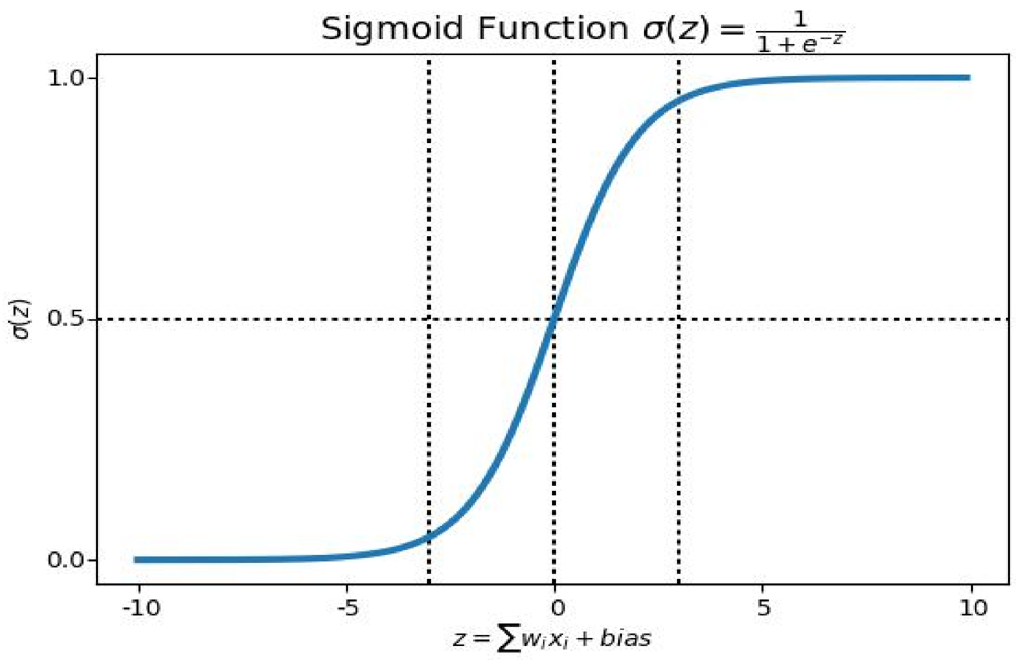


Fig 4 : Sigmoid function

$$f(x) = \frac{1}{1 + \mathrm{e}^{-(x)}}$$

1.4.2 Support Vector Machine

Support vector machines (SVMs) are powerful yet flexible supervised machine learning algorithms which are used both for classification and regression. But generally, they are used in classification problems. In 1960s, SVMs were first introduced but later they got refined in 1990. SVMs have their unique way of implementation as compared to other machine learning algorithms. Lately, they are extremely popular because of their ability to handle multiple continuous and categorical variables.

Working of SVM

An SVM model is basically a representation of different classes in a hyperplane in multidimensional space. The hyperplane will be generated in an iterative manner by SVM so that the error can be minimized. The goal of SVM is to divide the datasets into classes to find a maximum marginal hyperplane (MMH).

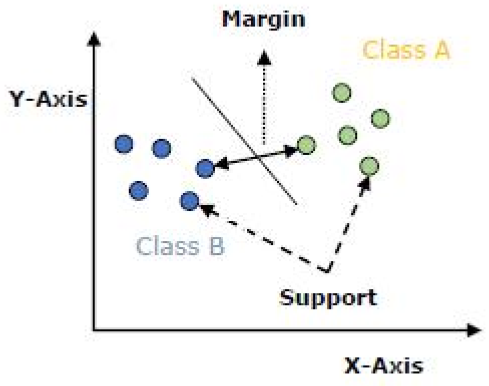


Fig 5 : SVM

The followings are important concepts in SVM −

➢ **Support Vectors** − Datapoints that are closest to the hyperplane is called support vectors. Separating line will be defined with the help of these data points.

➢ **Hyperplane** − As we can see in the above diagram, it is a decision plane or space which is divided between a set of objects having different classes.

➢ **Margin** − It may be defined as the gap between two lines on the closet data points of different classes. It can be calculated as the perpendicular

➢

distance from the line to the support vectors. Large margin is considered as a good margin and small margin is considered as a bad margin.

SVM Kernels

In practice, SVM algorithm is implemented with kernel that transforms an input data space into the required form. SVM uses a technique called the kernel trick in which kernel takes a low dimensional input space and transforms it into a higher dimensional space. In simple words, kernel converts non-separable problems into separable problems by adding more dimensions to it. It makes SVM more powerful, flexible and accurate. The following are some of the types of kernels used by SVM.

Linear Kernel

It can be used as a dot product between any two observations. The formula of linear kernel is as below −

$$K(x,xi)=sum(x*xi)\mathrm{K(x,xi)=sum(x*xi)}$$

From the above formula, we can see that the product between two vectors say

Polynomial Kernel

It is more generalized form of linear kernel and distinguish curved or nonlinear input space. Following is the formula for polynomial kernel −

$$k(X,Xi)=1+sum(X*Xi)\hat{}d\mathrm{k(X,Xi)=1+sum(X*Xi)\hat{}d}$$

Here d is the degree of polynomial, which we need to specify manually in the learning algorithm.

Radial Basis Function (RBF) Kernel

RBF kernel, mostly used in SVM classification, maps input space in indefinite dimensional space. Following formula explains it mathematically −

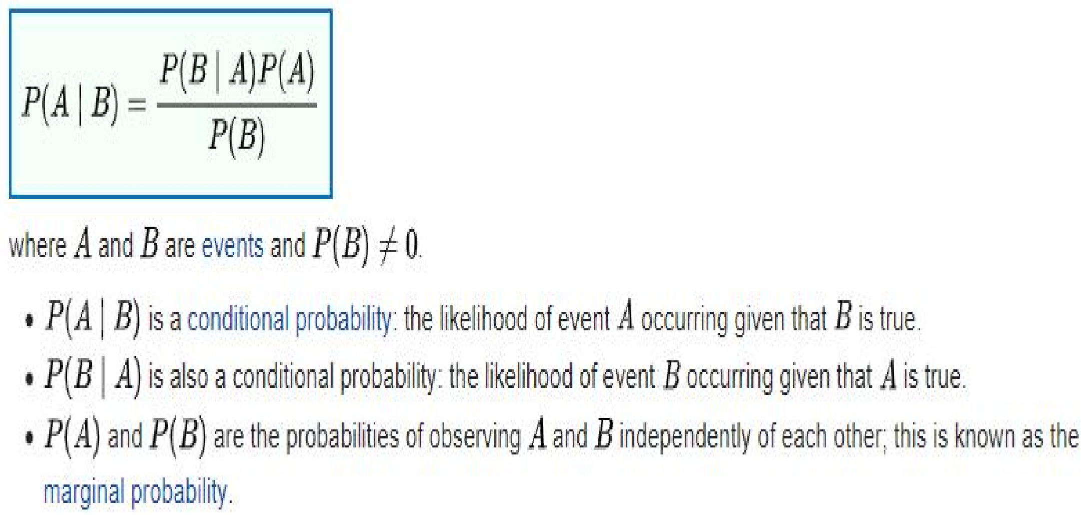
$$K(x,xi)=exp(-gamma*sum(x-xi\hat{}2))\mathrm{K(x,xi)=exp(-gamma*sum(x-xi\hat{}2))}$$

Here, *gamma* ranges from 0 to 1. We need to manually specify it in the learning algorithm. A good default value of *gamma* is 0.1.

1.4.3 Navie Bayes Classifier

A Naive Bayes classifier is a probabilistic machine learning model that’s used for classification task. It is a classifier in a machine learning model that is used to d**iscriminate different objects** based on **certain features.**

Bayes Theorem:



$$P(A \mid B) = \frac{P(B \mid A)P(A)}{P(B)}$$

where A and B are events and $P(B) \neq 0$.

- $P(A \mid B)$ is a conditional probability: the likelihood of event A occurring given that B is true.
- $P(B \mid A)$ is also a conditional probability: the likelihood of event B occurring given that A is true.
- $P(A)$ and $P(B)$ are the probabilities of observing A and B independently of each other; this is known as the marginal probability.

Working of Naïve Bayes' Classifier:

Working of Naïve Bayes' Classifier can be understood with the help of the below example:

Suppose we have a dataset of **weather conditions** and corresponding target variable "**Play**". So using this dataset we need to decide that whether we should play or not on a particular day according to the weather conditions. So to solve this problem, we need to follow the below steps:

1. Convert the given dataset into frequency tables.
2. Generate Likelihood table by finding the probabilities of given features.
3. Now, use Bayes theorem to calculate the posterior probability.

1.4.4 Decision Tree Classifier

Decision Tree algorithm is one of the simplest yet powerful Supervised Machine Learning algorithms. Decision Tree algorithm can be used to solve both regression and classification problems in Machine Learning. That is why it is also known as CART or Classification and Regression Trees. As the name suggests, in Decision Tree, we form a tree-like model of decisions and their possible consequences.

Types of Decision Tree Algorithms

There are two types of decision trees. They are categorized based on the type of the target variable they have. If the decision tree has a categorical target variable, then it is called a ‘categorical variable decision tree’. Similarly, if it has a continuous target variable, it is called a ‘continuous variable decision tree’.

How Does a Decision Tree in Machine Learning Work?

The process of training and predicting the target features using a decision tree in Machine Learning is given below:

- ➢ Feed a dataset, containing a number of training instances, with a set of features and a target
- ➢ Train the decision tree classification or regression models with the help of DecisionTreeClassifier() DecisionTreeRegressor () methods, and add the required criterion while building the decision tree model
- ➢ visualize the decision tree model

DecisionTreeClassifier (): It is nothing but the decision tree classifier function to build a decision tree model in Machine Learning using Python
DecisionTreeRegressio (): It is the decision tree regressor function used to build a decision tree model in Machine Learning using Python.

1.5 Introduction to Ensemble Learning :

When you want to purchase a laptop, will you simply walk up to the store and pick any laptop? It’s highly unlikely.

You would likely browse a few web portals where people have posted their reviews and compare different laptop models, checking for their features, specifications and prices. You will also probably ask your friends and colleagues for their opinion. In short, you wouldn’t directly reach a conclusion, but will instead make a decision considering the opinions of other people as well.

Ensemble Learning. *An ensemble is itself a supervised learning algorithm, because it can be trained and then used to make predictions.*

Ensemble Algorithm. *The goal of* ***ensemble algorithms*** *is to combine the predictions of several base estimators built with a given learning algorithm in order to improve generalizability / robustness over a single estimator.*

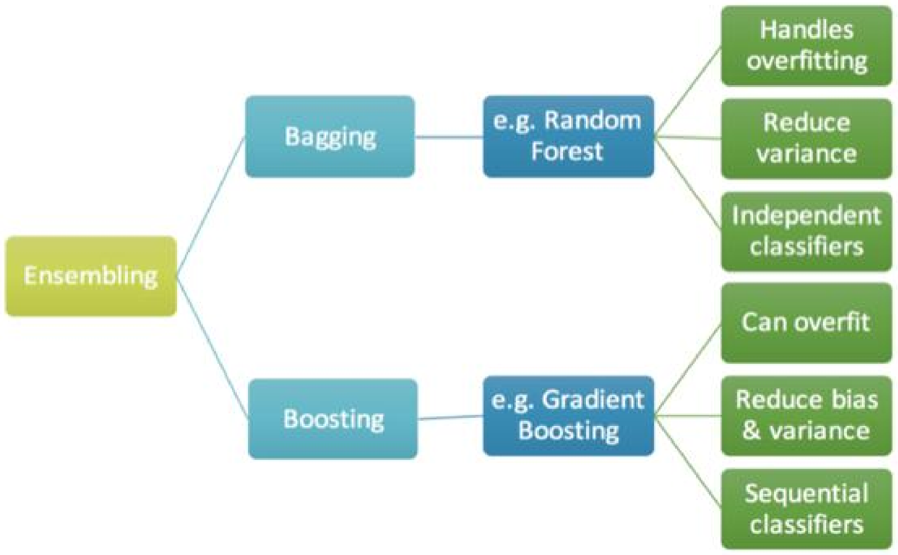


Fig 6 : some Ensembling technique

1.6 Ensemble models :

1.6.1 Random Forest

Random Forest is a popular machine learning algorithm that belongs to the supervised learning technique. It can be used for both Classification and Regression problems in ML. It is based on the concept of **ensemble learning,** which is a process of combining multiple classifiers to solve a complex problem and to improve the performance of the model.

As the name suggests, **"Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset."** Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output.

“The greater number of trees in the forest leads to higher accuracy and prevents the problem of overfitting”.

The below diagram explains the working of the Random Forest algorithm :

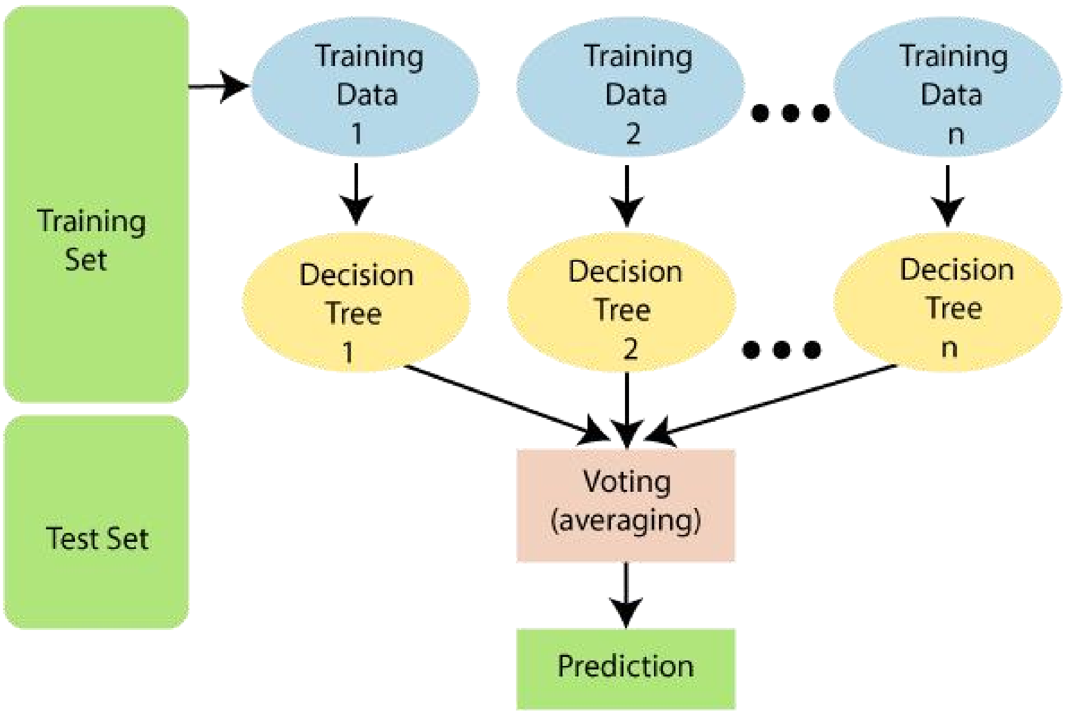


Fig 7 : working of random forest

How does Random Forest algorithm work?

Random Forest works in two-phase first is to create the random forest by combining N decision tree, and second is to make predictions for each tree created in the first phase.

The Working process can be explained in the below steps and diagram:

Step-1: Select random K data points from the training set.

Step-2: Build the decision trees associated with the selected data points (Subsets).

Step-3: Choose the number N for decision trees that you want to build.

Step-4: Repeat Step 1 & 2.

Step-5: For new data points, find the predictions of each decision tree, and assign the new data points to the category that wins the majority votes.

The working of the algorithm can be better understood by the below example:

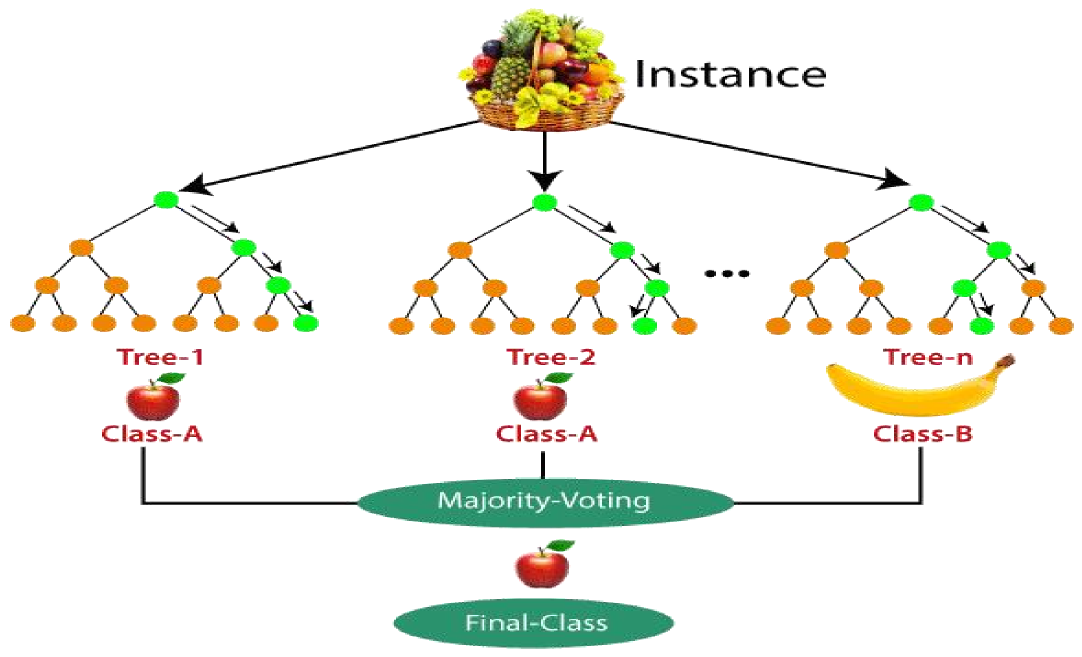


Fig 8 : Example of random forest

LITERATURE SURVEY

2.LITERATURE SURVEY

Several researcher have develop and design a systems for diabetes prediction based on various algorithms and methods. In reference [3] proposed a system for diabetes disease classification using Support Vector Machine (SVM). The authors used Pima Indian diabetes dataset for evaluation. The obtained accuracy was 78% base on using the Radial Basis Function (RBF) kernel of SVM as the classifier.

In reference[22] Diabetes is a standout among the most well-known non-transmittable diseases in the world. It is assessed to be the seventh leading cause for death . It is predicted that the diabetes rate in adults worldwide will become 642 million in 2040 . The early diagnosis of diabetes in patients has been a major goal for medical researchers and professionals.

In reference [18] designed a system for diabetes prediction, whose main aim is the prediction of diabetes for a candidate is suffering at a particular age. The proposed system is designed based on the concept of machine learning, by applying decision tree. Obtained results were satisfactory as the designed system works well in predicting the diabetes incidents at a particular age, with higher accuracy using Decision tree.

In reference [19] conducted a survey on data mining techniques on medical data for finding locally frequent diseases. The main focus of this survey is to analyse the data mining techniques required for medical data analysis that is especially used to discover locally frequent diseases such as heart lung cancer, ailments, breast cancer using classification and regression tree (CART) algorithm and Diabetes Prediction
Using Medical Data

In reference [4] used J48 Decision Tree and Naïve Bayes as a classifiers for classify the diagnosis of diabetes. Pima Indian diabetes dataset is used in the proposed system and the classification results was 74.8%, 79.5% for J48 Decision Tree and Naïve Bayes respectively.

In reference [5] proposed a system for diabetes classification based on using C4.5 algorithm for classification. The authors achieved classification rate of 91% by evaluating the training data through data feature relevance analysis.

In reference[6] proposed a system based on a technique in data mining for diabetes disease prediction. The proposed system has three main steps which are: preprocessing, feature extraction and parameter evaluation. In preprocessing step, the empty and anomalies sets are removed from the used dataset. Besides that, the helpful hidden patterns and relationships of the dataset are explored in the feature extraction step in order to improve the decision making result. Furthermore, the proposed system evaluated based on using J48, Naive Bayes and the achieved rates are 73.8%, 76.3% respectively.

In reference [7] fuzzy c means clustering and support vector machine for developing diabetes mellitus prediction. The authors used a dataset that consists of 768 cases and the obtained result was 59.5%.In reference [8] proposed a six various techniques to predict diabetic disease. The used techniques are Discriminant analysis, KNN Algorithm, Naïve Bayes, SVM with Linear Kernel function, and SVM with RBF Kernel function. The obtained results of the proposed system for the used techniques are 76.3% using discriminant analysis, 71.1% using KNN Algorithm, 76.1% using Naïve Bayes, 74.1% using SVM with Linear Kernel function, 74.1% using SVM with RBF Kernel function. However, several authors in [9] [10] [11] are used various methods in order to get the best prediction rate.

In reference[20] SVM is one of the standard set 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 hyperplane between the two classes.

In reference[21] Naive Bayes is a machine learning classifier which employs the Bayes Theorem. Using Bayes theorem posterior probability P(C|X) can be calculated from P(C),P(X) and P(X|C) [23].
Therefore, $P(C|X) = (P(X|C)\ P(C))/P(X)$

Where,

$P(C|X)$ = target class's posterior probability .

$P(X|C)$ = predictor class's probability.

$P(C)$ = class C's probability being true.

$P(X)$ = predictor's prior probability.

In reference [23] Iyer in their study proposed the use of the Naïve Bayes algorithm to predict the onset of diabetes. The study gave an accuracy result of 79.56%

In reference [9] G. Krishnaveni designed a model that uses the k-means algorithm and the logistic regression algorithm for predicting diabetes. The model attained a 95.42% accuracy

PROBLEM STATEMENT

3.Problem Statement

3.1 Problem Statement

Diabetes is considered as one of the deadliest and chronic diseases which causes an increase in blood sugar. Many complications occur if diabetes remains untreated and unidentified. The tedious identifying process results in visiting of a patient to a diagnostic center and consulting doctor. But the rise in machine learning approaches solves this critical problem. The motive of this study is to design a model which can prognosticate the likelihood of diabetes in patients with maximum accuracy. Therefore machine learning classification algorithms namely Logistic Regression,Support vector machine, Navie Bayes Classifier, Decision Tree, and ensemble technique Randomforest are used in this experiment to detect diabetes at an early stage. Experiments are performed on Pima Indians Diabetes Database (PIDD) which is sourced from UCI machine learning repository.

3.2 Objective

The primary aim of this project is to analyse the Diabetes Dataset and use Logistic Regression, Support Vector Machine, Naïve Bayes, Decision Tree and Random forest algorithms for prediction and to develop a prediction engine. The secondary aim is to develop a GUI application using tkinter for Diabetics prediction.

- Allow users to predict diabetes utilizing the prediction engine

3.3 Methodology

The methodology consists of 6 different phases as shown in Figure :

- ➢ Extracting the Diabetics patient Dataset
- ➢ Data preprocessing
- ➢ Applied various Algorithms
- ➢ Performance evalution on various measures
- ➢ Comparitive analysis based on accuracy and kfold crossvalidation
- ➢ Results

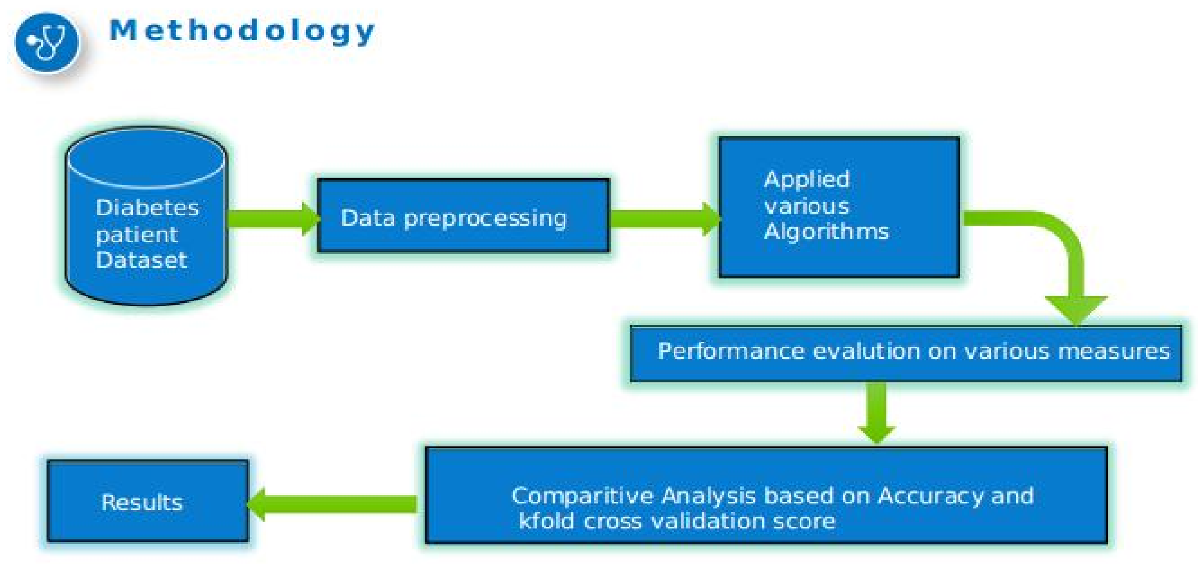

Methodology
Diabetes patient Dataset
Data preprocessing
Applied various Algorithms
Performance evalution on various measures
Comparitive Analysis based on Accuracy and kfold cross validation score
Results

Fig 9 : Methodology

The learning process starts with the gathering of data by different means, from various resources. Then the next step is to prepare the data, that is pre-process it in order to fix the data related issues and to reduce the dimensionality of the space by removing the irrelevant data (or selecting the data of interest). Since the amount of data that is being used for learning is large, it is difficult for the system to make decisions, so algorithms are designed using some logic, probability, statistics, control theory etc. to analyze the data and retrieve the knowledge from the past experiences. Next step is testing the model to calculate the accuracy and performance of the system. And finally optimization of the system.

DESIGN & IMPLEMENTATION

4.Design and Implementation

4.1 Data collection

The Pima Indian Diabetes data set obtained from UCI repository of machine learning was utilized for this study. The data set is comprised of 768 sample female patients from the Arizona, USA population who were examined for diabetes. The data set has a total of 8 attributes (representing medical diagnosis criteria) with one target class (which represents the status of each tested individual). In the data set there is a total of 268 tested positive instances and 500 tested negative instances. The attributes in the data set include the following:

- *Pregnancies*
- *Glucose* — The blood plasma glucose concentration after a 2 hour oral glucose tolerance test.
- *Blood Pressure* — Diastolic blood pressure (mm/HG).
- *Skin Thickness* — Skin fold thickness of the triceps (mm).
- *Insulin* — 2 hour serum insulin (mu U/ml).
- *BMI* — Body mass index (kg/m squared)
- *Diabetes Pedigree Function* — A function that determines the risk of type 2 diabetes based on family history, the larger the function, the higher the risk of type 2 diabetes.

Link to the dataset : https://www.kaggle.com/uciml/pima-indians-diabetes-database

4.2 Data visualization

Data Visualization is the presentation of data in graphical format. It helps people understand the significance of data by summarizing and presenting a huge amount of data in a simple and easy-to-understand format and helps communicate information clearly and effectively.

With the help of data visualization, we can see how the data looks like and what kind of correlation is held by the attributes of data. It is the fastest way to see if the features correspond to the output.

Lets us visualize Pima Indians Diabetics Dataset :

➢ Let’s also look at how many people in the dataset are diabetic and how many are not

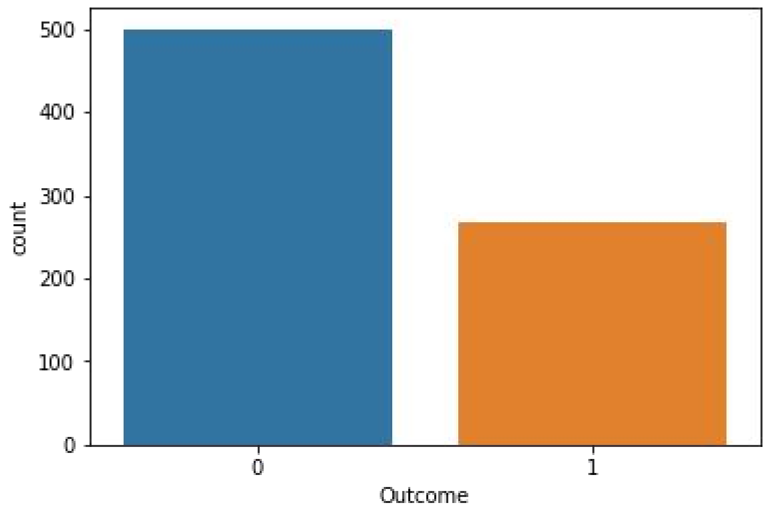
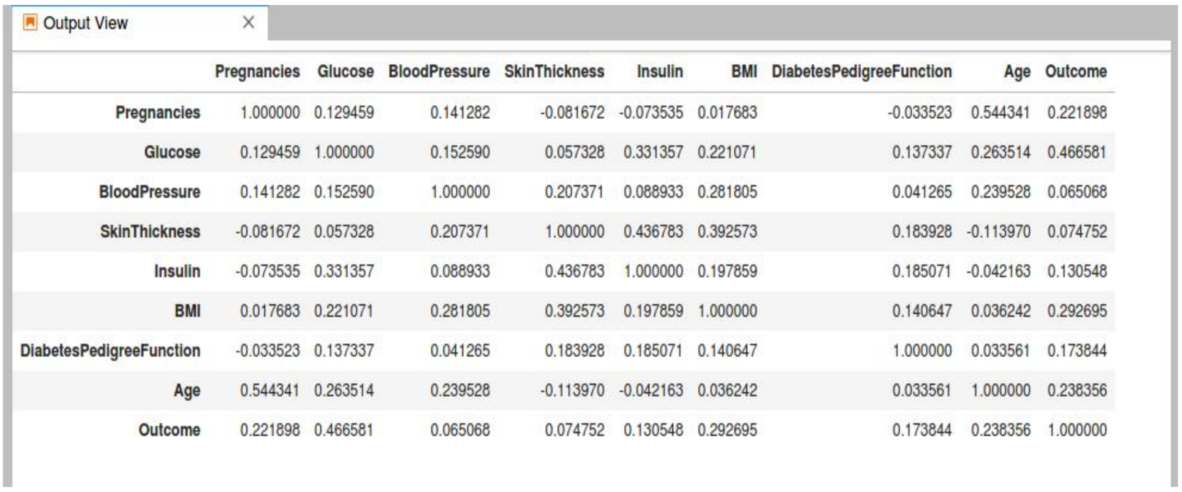


Fig 10 : graph of Outcome variable

“The countplot tells us that the dataset is imbalanced, as the number of patients who don’t have diabetes is more than those who do”.

➢ Let’s find correlation of every pair of features (and the outcome variable), and visualize the correlations using a heatmap.



	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
Pregnancies	1.000000	0.129459	0.141282	-0.081672	-0.073535	0.017683	-0.033523	0.544341	0.221898
Glucose	0.129459	1.000000	0.152590	0.057328	0.331357	0.221071	0.137337	0.263514	0.466581
BloodPressure	0.141282	0.152590	1.000000	0.207371	0.088933	0.281805	0.041265	0.239528	0.065068
SkinThickness	-0.081672	0.057328	0.207371	1.000000	0.436783	0.392573	0.183928	-0.113970	0.074752
Insulin	-0.073535	0.331357	0.088933	0.436783	1.000000	0.197859	0.185071	-0.042163	0.130548
BMI	0.017683	0.221071	0.281805	0.392573	0.197859	1.000000	0.140647	0.036242	0.292695
DiabetesPedigreeFunction	-0.033523	0.137337	0.041265	0.183928	0.185071	0.140647	1.000000	0.033561	0.173844
Age	0.544341	0.263514	0.239528	-0.113970	-0.042163	0.036242	0.033561	1.000000	0.238356
Outcome	0.221898	0.466581	0.065068	0.074752	0.130548	0.292695	0.173844	0.238356	1.000000

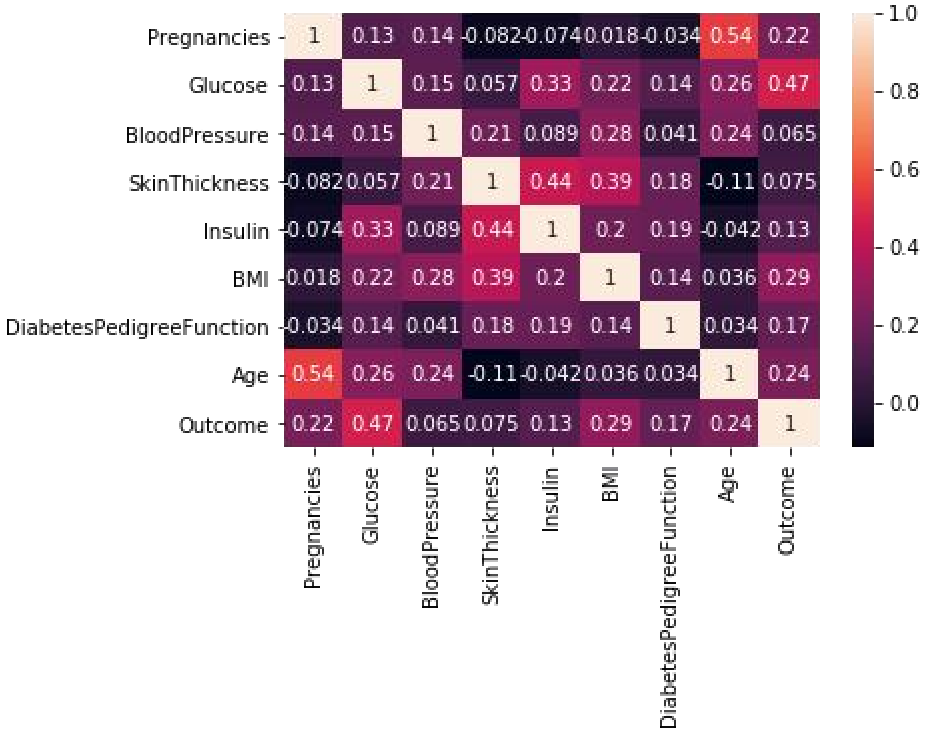


Fig 11 : Heatmap of dataset

“In the above heatmap, brighter colors indicate more correlation. As we can see from the table and the heatmap, glucose levels, age, BMI and number of pregnancies all have significant correlation with the outcome variable. Also notice the correlation between pairs of features, like age and pregnancies, or insulin and skin thickness”.

➢ Let us build the histogram for better visualization of the dataset .

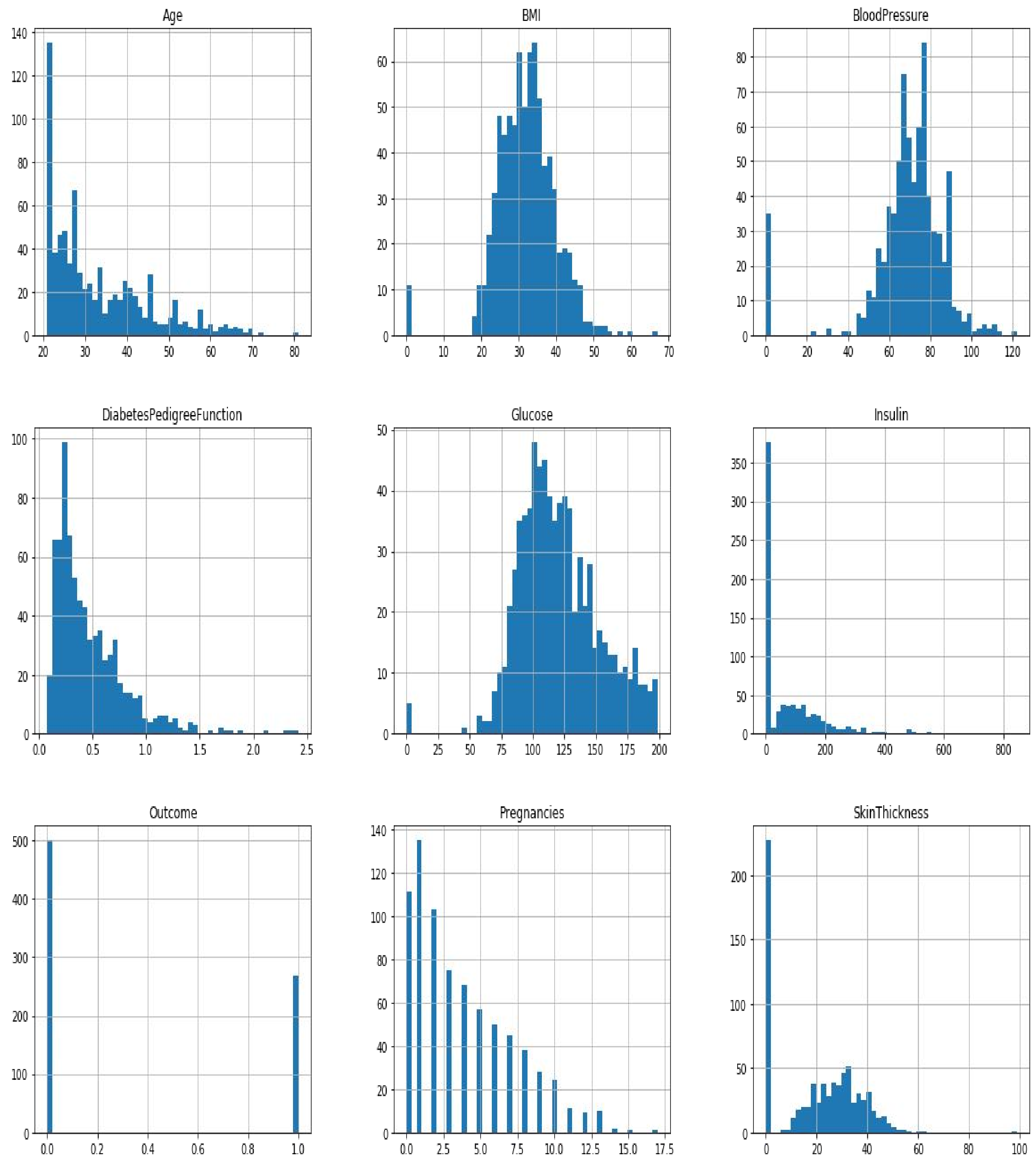


Fig 12 : Histogram of dataset

➢ With a graphical visualization of the data we have a better understanding of the various features values distribution: for example we can understand what's the average age of the people or the average BMI etc..

4.3 Data Preprocessing

Today's real world databases are highly susceptible to noisy, missing, and inconsistent data due to their typically huge sizes and their likely origin from multiple, heterogeneous sources . Data quality is an important factor in the data mining process for disease prediction and diagnosis, because low quality data may lead to inaccurate or low prediction result. In order to make our original dataset more productive and applicable for predicting diabetes, we applied several preprocessing techniques.

it is observed that Plasma glucose concentration, Diastolic blood pressure, Skin fold thickness, 2hr serum insulin and Body mass index have a min value of 0. Medical knowledge explains that such attributes (medical result) cannot be 0; therefore it suggests that the dataset contains a missing value that if not handledcan impair the quality of our model result and accuracy.

Blood pressure: By observing the data we can see that there are 0 values for blood pressure. And it is evident that the readings of the data set seem wrong because a living person cannot have a diastolic blood pressure of zero. By observing the data we can see 35 counts where the value is 0.now calculate the median value and replace the null values with that median .

Plasma glucose levels: Even after fasting glucose levels would not be as low as zero. Therefore zero is an invalid reading. By observing the data we can see 5 counts where the value is 0.now calculate the median value and replace the null values with that median .

Skin Fold Thickness: For normal people, skin fold thickness can’t be less than 10 mm better yet zero. Total count where value is 0: 227.now calculate the median value and replace the null values with that median .

BMI: Should not be 0 or close to zero unless the person is really underweight which could be life-threatening..now calculate the median value and replace the null values with that median .

Insulin: In a rare situation a person can have zero insulin but by observing the data, we can find that there is a total of 374 counts.now calculate the median value and replace the null values with that median .

Here are several ways to handle invalid data values :

- ➢ Ignore/remove these cases: This is not actually possible in most cases because that would mean losing valuable information. And in this case “skin thickness” and “insulin” columns mean to have a lot of invalid points. But it might work for “BMI”, “glucose ”and “blood pressure” data points.
- ➢ Put average/mean values: This might work for some data sets, but in our case putting a mean value to the blood pressure column would send a wrong signal to the model.
- ➢ Avoid using features: It is possible to not use the features with a lot of invalid values for the model. This may work for “skin thickness” but it's hard to predict that.

4.4 Preparing the data :

➢ Splitting the Dataset :

Now that we have transformed the data we need to split the dataset in two parts: a training dataset and a test dataset. Splitting the dataset is a very important step for supervised machine learning models. Basically we are going to use the first part to train the model (ignoring the column with the pre assigned label), then we use the trained model to make predictions on new data (which is the test dataset, not part of the training set) and compare the predicted value with the pre assigned label.

Here we are spliting the data into two parts 70% of the data as training data and 30% of the data as testing data.and then separate labels from the rest of the dataset .

➢ Feature Scaling :

One of the most important data transformations we need to apply is the features scaling. Basically most of the machine learning algorithms don't work very well if the features have a different set of values. In our case for example the Age ranges from 20 to 80 years old, while the number of times a patient has been pregnant ranges from 0 to 17. For this reason we need to apply a proper transformation.

4.5 Comparing models :

It's not possible to know in advance which algorithm will work better with our dataset. We need to compare a few and select the one with the "best score".

➢ Comparing multiple algorithms :

To compare multiple algorithms with the same dataset, there is a very nice utility in sklearn called model_selection. We create a list of algorithms and then we score them using the comparison method. At the end we pick the one with the best score.

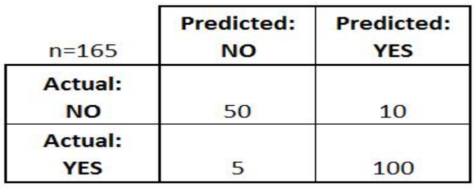
Now import all the necessary algorithms and prepare an array with all the algorithms and test the accuracy of all the models and cross validation scores using kfolds cross Validation .

4.6 Choosing the best model :

Metrics to Evaluate your Machine Learning Algorithm :

Classification Accuracy : Classification Accuracy is what we usually mean, when we use the term accuracy. It is the ratio of number of correct predictions to the total number of input samples.

Confusion Matrix : Confusion Matrix as the name suggests gives us a matrix as output and describes the complete performance of the model.



n=165	Predicted: NO	Predicted: YES
Actual: NO	50	10
Actual: YES	5	100

Precision : It is the number of correct positive results divided by the number of positive results predicted by the classifier.

Recall : It is the number of correct positive results divided by the number of ***all*** relevant samples (all samples that should have been identified as positive).

F1 Score : it 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

The Accuracy Scores and Cross validation scores of the algorithms Logistic regression , support vector machine , Navie bayes , Decision tree and Random forest is as fallows :

Accuracy scores :

Algorithm	**Accuracy**
Logistic Regresion	0.753247
Navie bayes	0.753247
Support vector	**0.766234**
Decision Tree	0.740260
RandomForest	0.740260

Table 1 : Accuracy Scores of tested Algorithms

Accuracy measures :

Algorithm	**Confusion matrix**	**Precision**	**Recall**	**F1 score**
Logistic Regresion	[[138, 19], [38, 36]]	0.654545	0.486486	0.558140
Navie bayes	[[137, 20], [37, 37]]	0.649123	0.500000	0.564885
Support vector	[[139, 18], [36, 38]]	0.678571	0.513514	0.584615
Decision Tree	[[128, 29], [31, 43]]	0.611111	0.594595	0.602740
Random Forest	[[137, 20], [40, 34]]	0.600000	0.405405	0.483871

Table 2 : other accuracy measures

K-fold cross validation scores :

K-fold cross validation scores	
Algorithm	**Cross validation score**
Logistic Regresion	0.763487
Navie bayes	0.735535
Support vector	**0.765339**
Decision Tree	0.653389
Random Forest	0.729839

Table 3 : Cross validation Scores of tested Algorithms

By observing the above accuracy and cross validation scores Support vector gives best accuracy as well as cross validation score.In k-fold cross validation we split the data-set into k number of subsets(known as folds) then we perform training on the all the subsets but leave one(k-1) subset for the evaluation of the trained model. In this method, we iterate k times with a different subset reserved for testing purpose each time.

So, we are using support vector for prediction of diabetics in our project.

4.7 Predicting diabetics

Find the best parameters for SVC and predicting :

The default parameters for an algorithm are rarely the best ones for our dataset. Using sklearn we can easily build a parameters grid and try all the possible combinations. At the end we inspect the best_estimator_ property and get the best ones for our dataset.

SVM also has some hyper-parameters (like what C or gamma values to use) and finding optimal hyper-parameter is a very hard task to solve. But it can be found by just trying all combinations and see what parameters work best. The main idea behind it is to create a grid of hyper-parameters and just try all of their combinations (hence, this method is called **Gridsearch).**Scikit-learn has this functionality built-in with GridSearchCV.

GridSearchCV takes a dictionary that describes the parameters that could be tried on a model to train it. The grid of parameters is defined as a dictionary, where the keys are the parameters and the values are the settings to be tested.

Finally apply the obtained best parameters to the model and train the model after training the model create a fake person data and predict it for diabetics

CREATING GUI APPLICATION

5.Creating GUI application for diabetics prediction using tkinter

5.1 Introduction to tkinter :

Tkinter is the standard GUI library for Python. Python when combined with Tkinter provides a fast and easy way to create GUI applications. Tkinter provides a powerful object-oriented interface to the Tk GUI toolkit.

Creating a GUI application using Tkinter is an easy task. All you need to do is perform the following steps −

- ➢ Import the *Tkinter* module.
- ➢ Create the GUI application main window.
- ➢ Add required widgets to the GUI application.
- ➢ Enter the main event loop to take action against each event triggered by the user.

Example :

```
import Tkinter

top = Tkinter.Tk()

# Code to add widgets will go here...

top.mainloop()
```

This would create a following window −

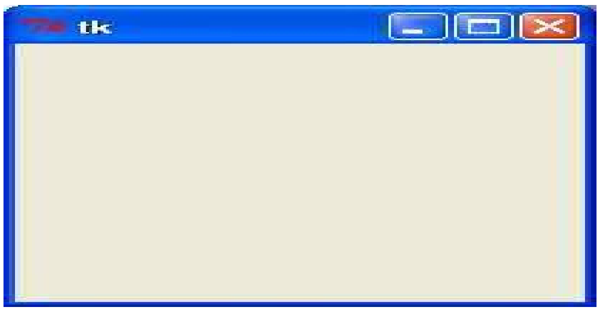


Fig 13 : sample Tkinter window

5.2 Elements in Gui :

In our gui application we have lables , entries and button which are described as fallows :

Label : The Label is used to specify the container box where we can place the text or images. This widget is used to provide the message to the user about other widgets used in the python application.

Our application window contain 9 labels ,The first one represents the title of the window as DiabeticsPredictionSystem and remaining are to represent the enties boxes these are labeled as pregnancies,Glucose,Bloodpressure,SkinThickness, Insuline,BMI,Pedigree,Age respectively.

Entry : The Entry widget is used to provde the single line text-box to the user to accept a value from the user.

Our application window contain 8 entries which are used to enter the values of pregnancies,Glucose,Bloodpressure,SkinThickness, Insuline,BMI,Pedigree,Age respectively.

Button : The Button widget is used to add buttons in a Python application. These buttons can display text or images that convey the purpose of the buttons. You can attach a function or a method to a button which is called automatically when you click the button.

Our application window contain button called PREDICT . On pressing these button it predicts whether the person has diabetics or not .

5.3 Working of Gui

Our gui window looks like as below :

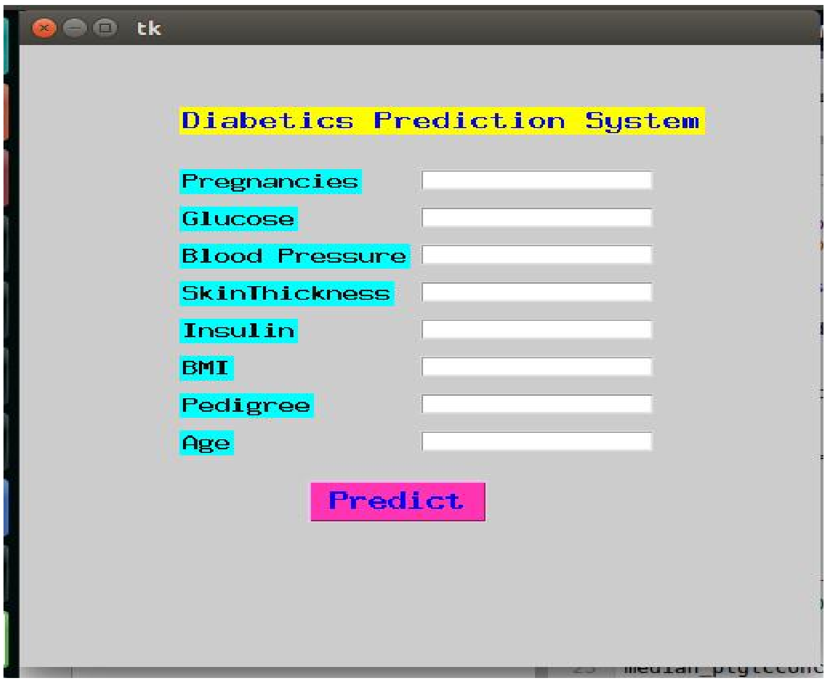


Fig 14 : Our GUI window

➢ On looking to our application window it contains various entries to enter the values of various attributes such as pregnancy count , glucose level,blood pressure, skinthickness,Insulin,body mass index,pedigree function ,and age of the person

➢ After entering all these values we need to click on PREDICT button which predict the whether the patient has diabetics or not and display the result.

➢ In the background the predict button calls the function called predicting which is defined for prediction of diabetics.

➢ At the end the message '**Tested_positive**' or '**Tested_negative**' get displayed on the application window.Tested_positive is for the person who have risk of diabetics and Tested_negative is for the person who are risk free from diabetics.

IMPLEMENTATION CODE

6.Implementation code

6.1 Source code :

```
import gui
from tkinter import *
# import the libraries needed to read the dataset
import os
import pandas as pd
import numpy as np
dataset= pd.read_csv('dia.csv')
dataset.shape
# Calculate the median value for BMI
median_bmi = dataset['BMI'].median()
# Substitute it in the BMI column of the
# dataset where values are 0
dataset['BMI'] = dataset['BMI'].replace(to_replace=0, value=median_bmi)

# Calculate the median value for BloodP
median_bloodp = dataset['BloodPressure'].median()
# Substitute it in the BloodP column of the
# dataset where values are 0
dataset['BloodPressure'] = dataset['BloodPressure'].replace(
    to_replace=0, value=median_bloodp)

# Calculate the median value for PlGlcConc
median_plglcconc = dataset['Glucose'].median()
# Substitute it in the PlGlcConc column of the
# dataset where values are 0
dataset['Glucose'] = dataset['Glucose'].replace(
    to_replace=0, value=median_plglcconc)

# Calculate the median value for SkinThick
median_skinthick = dataset['SkinThickness'].median()
# Substitute it in the SkinThick column of the
# dataset where values are 0
dataset['SkinThickness'] = dataset['SkinThickness'].replace(
    to_replace=0, value=median_skinthick)

# Calculate the median value for TwoHourSerIns
median_twohourserins = dataset['Insulin'].median()
# Substitute it in the TwoHourSerIns column of the
# dataset where values are 0
dataset['Insulin'] = dataset['Insulin'].replace(
    to_replace=0, value=median_twohourserins)

X=dataset.drop("Outcome",axis=1)
y=dataset.Outcome


from sklearn.model_selection import train_test_split
from sklearn.model_selection import train_test_split

x_train,x_test,y_train,y_test=train_test_split(X,y,test_size=0.3,random_state=0)

# Import all the algorithms we want to test
from sklearn.linear_model import LogisticRegression
from sklearn.naive_bayes import GaussianNB
from sklearn.svm import SVC
from sklearn.svm import LinearSVC
from sklearn.tree import DecisionTreeRegressor
from sklearn.ensemble import RandomForestClassifier
from sklearn.ensemble import GradientBoostingClassifier
from sklearn.metrics import accuracy_score
from sklearn.metrics import confusion_matrix
from sklearn.metrics import precision_score
from sklearn.metrics import recall_score
from sklearn.metrics import f1_score

from sklearn import model_selection

models = []
models.append(('LogisticRegre ', LogisticRegression()))
models.append(('Navie bayes ', GaussianNB()))
models.append(('Support vector', SVC(kernel='linear')))
models.append(('Decision Tree ', DecisionTreeRegressor()))
models.append(('RandomForest ', RandomForestClassifier()))

#finding Accuracy of the models
names = []
scores = []
mat=[]
precision=[]
recall=[]
fscore=[]
print('Accuracy Scores')
for name, model in models:
    model.fit(x_train, y_train)
    y_pred = model.predict(x_test)
    scores.append(accuracy_score(y_test, y_pred))
    mat.append(confusion_matrix(y_test,y_pred))
    precision.append(precision_score(y_test,y_pred))
    recall.append(recall_score(y_test,y_pred))
    fscore.append(f1_score(y_test,y_pred))

    names.append(name)

tr_split = pd.DataFrame({'Name': names, 'Score': scores,'confusin matrix':
mat,'Precision' :precision,'recall ': recall,'f_score' : fscore})
print(tr_split)
```

```
# Apply a scaler
from sklearn.preprocessing import MinMaxScaler as Scaler
scaler = Scaler()
scaler.fit(x_train)
train_set_scaled = scaler.transform(x_train)
test_set_scaled = scaler.transform(x_test)




# Prepare an array with all the algorithms
models = []
models.append(('LogisticRegre ', LogisticRegression()))
models.append(('Navie bayes ', GaussianNB()))
models.append(('Support vector', SVC(kernel='linear')))
models.append(('Decision Tree ', DecisionTreeRegressor()))
models.append(('RandomForest ', RandomForestClassifier()))


# Prepare the configuration to run the test
seed = 7
results = []
names = []
X = train_set_scaled
Y = y_train

# finding cross validation score
for name, model in models:

      kfold = model_selection.KFold(n_splits=10, random_state=seed)
      #print(kfold)
      cv_results = model_selection.cross_val_score(model, X, Y, cv=kfold,
scoring='accuracy')
      #print("cv",cv_results)
      results.append(cv_results)
      names.append(name)
      msg = "%s :   %f (%f)" % (name, cv_results.mean(), cv_results.std())
      ##print(msg)

#Find the best parameters for SVC
from sklearn.model_selection import GridSearchCV

param_grid = {
      'C': [1.0, 10.0, 50.0],
      'kernel': ['linear', 'rbf', 'poly', 'sigmoid'],
      'shrinking': [True, False],
      'gamma': ['auto', 1, 0.1],
      'coef0': [0.0, 0.1, 0.5]
}
```

```
model_svc = SVC()

grid_search = GridSearchCV(model_svc, param_grid, cv=10, scoring='accuracy')
#print(grid_search)
grid_search.fit(train_set_scaled, y_train)



# Create an instance of the algorithm using parameters

# from best_estimator_ property

svc = grid_search.best_estimator_



# Use the whole dataset to train the model

X = np.append(train_set_scaled, test_set_scaled, axis=0)

Y = np.append(y_train,y_test, axis=0)



# Train the model

svc.fit(X, Y)

# predicting through normal funtion
#defining a funtion for prediction of diabetics
def predicting(report):
	new_df_scaled = scaler.transform(report)
	prediction = svc.predict(new_df_scaled)
	print(prediction)
	if prediction==1:
		print("*** <WARNING> you have Diabetics***")
	else:
		print("***Cool!!! you are Diabetics free***")
new_df = pd.DataFrame([[6, 168, 72, 35, 0, 43.6, 0.627, 65]])
new_df1=pd.DataFrame([[1,103,30,38,83,43.3,0.183,33]])


#defining a funtion for prediction of diabetics used to predict in GUI

def predicting():
	p=float(gui.preg.get())
	g=float(gui.glucose.get())
	bp=float(gui.bp.get())
	s=float(gui.skin.get())
	i=float(gui.ins.get())
```

```
    b=float(gui.bmi.get())
    p=float(gui.pedi.get())
    a=float(gui.age.get())
    report=pd.DataFrame([[p,g,bp,s,i,b,p,a]])
    new_df_scaled = scaler.transform(report)
    prediction = svc.predict(new_df_scaled)
    print(prediction)


    if prediction==1:

        result=Label(gui.window,text="Tested_Positive",font=("arial",20,"bold"),bg
        ="red")
        result.place(x=100,y=400)
    else:

        result=Label(gui.window,text="Tested_Negtive",font=("arial",20,"bold"),bg
        ="green yellow")
        result.place(x=100,y=400)
```

6.3 Gui using tkinter code

```
from tkinter import *
import model
import tkinter as tk
window=tk.Tk()
window.geometry("500x500")
window.configure(background='gray80')

label=Label(window,text="Diabetics Prediction
System",fg="blue",bg="yellow",font=("arial",14,"bold"))
label.place(x=100,y=50)

label1=Label(window,text="Pregnancies",font=("arial",12,"bold"),bg="cyan")
label1.place(x=100,y=100)

preg=Entry(window,text="no of pregnancies")
preg.place(x=250,y=100)

label2=Label(window,text="Glucose",font=("arial",12,"bold"),bg="cyan")
label2.place(x=100,y=130)

glucose=Entry(window,text="enter glucose")
glucose.place(x=250,y=130)

label3=Label(window,text="Blood Pressure",font=("arial",12,"bold"),bg="cyan")
label3.place(x=100,y=160)
```

```
bp=Entry(window,text="blood pressure")
bp.place(x=250,y=160)

label4=Label(window,text="SkinThickness",font=("arial",12,"bold"),bg="cyan")
label4.place(x=100,y=190)

skin=Entry(window,text="SkinThickness")
skin.place(x=250,y=190)

label5=Label(window,text="Insulin",font=("arial",12,"bold"),bg="cyan")
label5.place(x=100,y=220)

ins=Entry(window,text="insulin level")
ins.place(x=250,y=220)

label2=Label(window,text="BMI",font=("arial",12,"bold"),bg="cyan")
label2.place(x=100,y=250)

bmi=Entry(window,text="body mass index")
bmi.place(x=250,y=250)




label2=Label(window,text="Pedigree",font=("arial",12,"bold"),bg="cyan")
label2.place(x=100,y=280)

pedi=Entry(window,text="Diabetics pedigree funtion")
pedi.place(x=250,y=280)

label2=Label(window,text="Age",font=("arial",12,"bold"),bg="cyan")
label2.place(x=100,y=310)

age=Entry(window,text="AGE")
age.place(x=250,y=310)

button=Button(window,text="Predict",bg="maroon1",fg="blue",font=("arial",15,"b
old"),command=model.predicting)
button.place(x=180,y=350)

window.mainloop()
```

RESULTS

7.Results

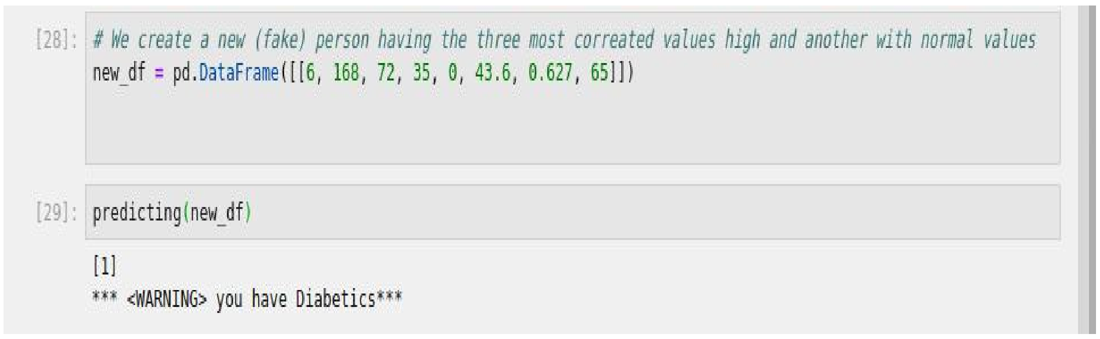
7.1 Results

➢ We create a new person data and check for the diabetics as fallows:

```
new_df = pd.DataFrame([[6, 168, 72, 35, 0, 43.6, 0.627, 65]])
predicting(new_df)
```

➢ Then the result is displayed as fallows :

```
[1]
*** <WARNING> you have Diabetics***
```



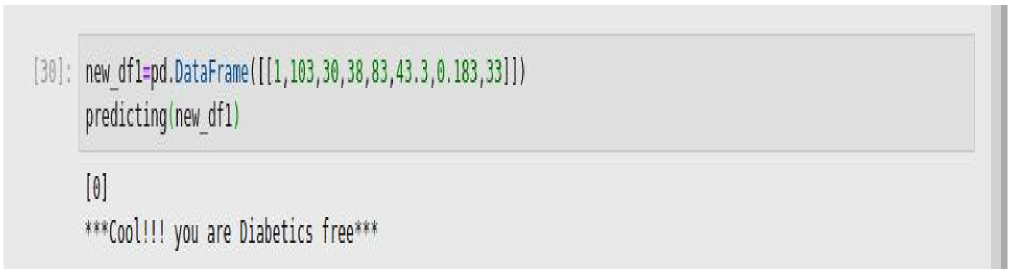
```
[28]: # We create a new (fake) person having the three most correated values high and another with normal values
new_df = pd.DataFrame([[6, 168, 72, 35, 0, 43.6, 0.627, 65]])

[29]: predicting(new_df)

[1]
*** <WARNING> you have Diabetics***
```

➢ Now check for another person :

```
new_df1=pd.DataFrame([[1,103,30,38,83,43.3,0.183,33]])
predicting(new_df1
```

Result :

```
[0]
***Cool!!! you are Diabetics free***
```



```
[30]: new_df1=pd.DataFrame([[1,103,30,38,83,43.3,0.183,33]])
predicting(new_df1)

[0]
***Cool!!! you are Diabetics free***
```

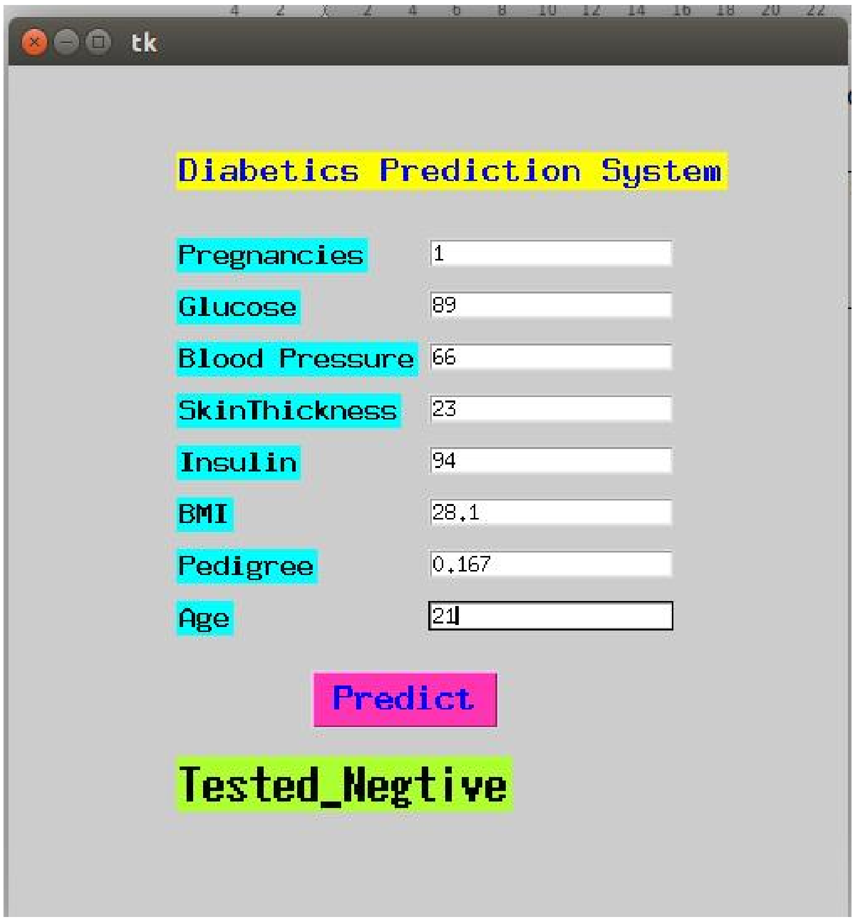
7.2 Results through GUI application

➢ Now check the results through GUI application simple by clicking on the predict button .

◆ Enter fallowing values in GUI and press PREDICT button :

preg	plas	blpres	skin	insu	bmi	pedi	age
1	89	66	23	94	28.1	0.167	21

RESULT : Tested negative


tk

Diabetics Prediction System

Pregnancies	1
Glucose	89
Blood Pressure	66
SkinThickness	23
Insulin	94
BMI	28.1
Pedigree	0.167
Age	21

Tested_Negtive

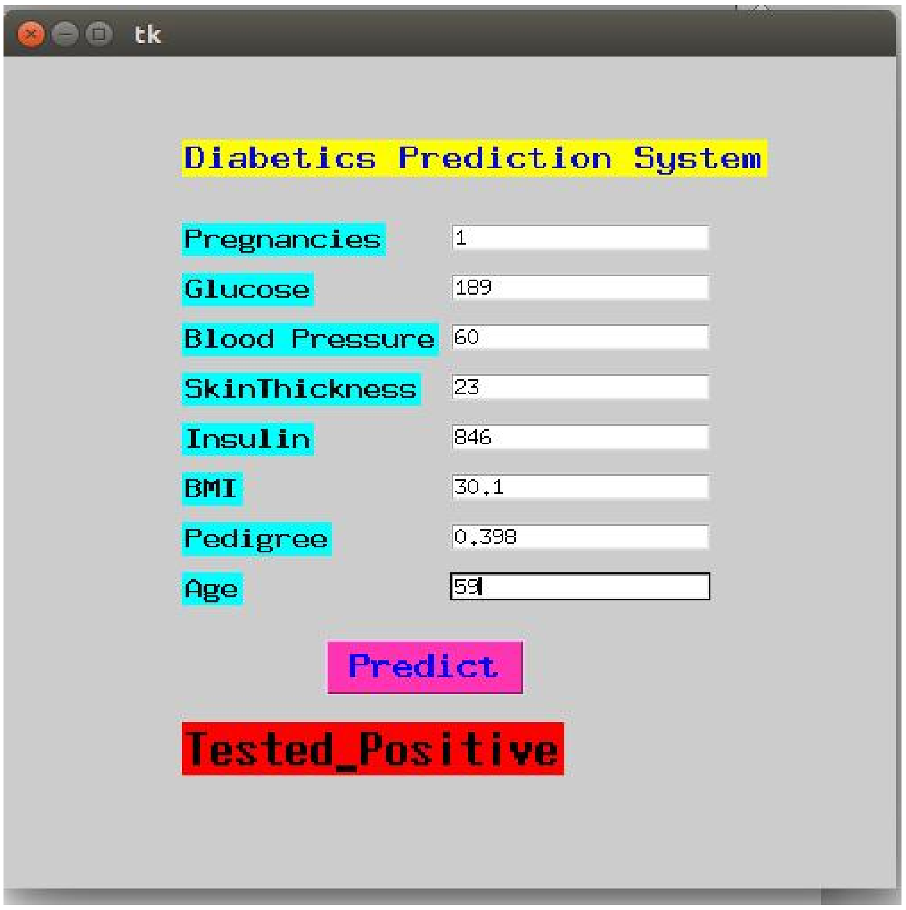
Fig 15 : Results(1)

- Now check for another person with fallowing values

preg	plas	blpres	skin	insu	bmi	pedi	age
1	189	60	23	846	30.1	0.398	59

Enter fallowing values in GUI and pess PREDICT button

RESULT : Tested Positive


tk

Diabetics Prediction System

Pregnancies	1
Glucose	189
Blood Pressure	60
SkinThickness	23
Insulin	846
BMI	30.1
Pedigree	0.398
Age	59

Tested_Positive

Fig 16 : Results(2)

CONCLUSION

8.Conclusion

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. The aim of this work was to design an efficient model for the prediction of diabetes. After a careful observation we select the support vector machine for prediction of diabetics which gives the best accuracy(77%) and best cross validation score (77%) among various models .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.

POSSIBLE IMPROVEMENTS

9.Possible Improvements

This model developed to predict diabetes in women of Pima Indian heritage is a great starting point. However, more work needs to be done to improve the model in order to properly assist the community in their future health efforts.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.There may be still space for further analysis and optimization, for example trying different data transformations or trying algorithms that haven't been tested yet.

SYSTEM REQUIREMENTS

10.System Requirements

Software Requirements :

Operating system : Ubuntu / windows

Programming language : Python

Tools : Anaconda navigator / Jupiter notebook

Hardware Requirements :

**Processor : ** corei3 or higher

Ram : 4GB(min)

Hard Disc : 500GB

REFERENCES

11.References

[1] Yilmaz N., Inan O., Uzer M.S., “ A new data preparation method based on clustering algorithms for diagnosis systems of heart and diabetes diseases,” J Med Syst, vol. 38, no. 5 2014.

[2] Lowongtrakool C., Hiransakolwong N., “Noise filtering in unsupervised clustering using computation intelligence,” International Journal of Math, vol. 6, no. 59, pp. 2911–2920, 2012.

[3] V. Anuja and R.Chitra., “Classification Of Diabetes Disease Using Support Vector Machine”, International Journal of Engineering Research and Applications (IJERA), vol.3,Issue 2, pp. 1797-1801, 2013.

[4] Aiswarya I., S. Jeyalatha and Ronak S., “Diagnosis Of Diabetes Using Classification Mining Techniques”, International Journal of Data Mining & Knowledge Management Process (IJDKP), vol.5, ,No. 1, pp. 1-14, 2015.

[5] K.Rajesh and V.Sangeetha,”Application of Data Mining Methods and Techniques for Diabetes Diagnosis,” in proceedings of International journal of Engineering and Innovative Technology, vol.2, Issue 3, pp. 43-46, 2012.

[6] Harleen and Dr. Pankaj B.,”A Prediction Technique in Data Mining for Diabetes Mellitus,” Journal of Management Sciences and Technology, vol. 4, Issue 1, pp. 1-12, 2016.

[7] Ravi S. and Smt T., ”Prognosis of Diabetes Using Data mining Approach-Fuzzy C Means Clustering and Support Vector Machine,” International Journal of Computer Trends and Technology (IJCTT), vol. 11, No. 2, pp. 94-98, 2014.

[8] G. Krishnaveni*, T. Sudha,” A Novel Technique To Predict Diabetic Disease Using Data Mining Classification Techniques” in International Conference on Innovative Applications in Engineering and Information Technology (ICIAEIT2017), vol. 3, Issue 1, pp. 5-11, 2017.

[9] Raj A., Vishnu P., and Kavita B.,"K-Fold Cross Validation and Classification Accuracy of PIMA Indian Diabetes Data Set Using Higher Order Neural Network and

PCA", International Journal of Soft Computing and Engineering (IJSCE), Volume-2, Issue-6, pp. 436-438, January 2013.

[10] Vrushali B., and Rakhi W., "Review on Prediction of Diabetes using Data Mining Technique", International Journal of Research and Scientific Innovation (IJRSI), Volume IV, Issue IA, pp. 43-46, January 2017.

[11] Thirumal P., and Nagarajan N.," Utilization of Data Mining Techniques for Diagnosis of Diabetes Mellitus - A Case Study", ARPN Journal of Engineering and Applied Sciences, Vol. 10, No. 1, pp. 8-13, January 2015.

[12] Jain A.K., Murty M.N., Flynn P.J., "Data clustering: A review," ACM Computing Surveys, vol. 31, no. 3, pp. 264-323, 1999.

[13] Iyer A., Jeyalatha S., Sumbaly R., "Diagnosis of diabetes using classification mining techniques," International Journal of Data Mining & Knowledge Management Process (IJDKP), vol. 5, no. 1, 2015.

[14] Kumari A.V., Chitra R., "Classification of diabetes disease using support vector machine," Int J Eng Res Appl, vol. 3, no. 2, pp. 1797-1801, 2013.

[15] Yilmaz N., Inan O., Uzer M.S., " A new data preparation method based on clustering algorithms for diagnosis systems of heart and diabetes diseases," J Med Syst, vol. 38, no. 5 2014.

[16] Çalişir D., Doğantekin E., "An automatic diabetes diagnosis system based on LDA-Wavelet Support Vector Machine classifier," Expert Syst Appl, vol. 38, no. 7, pp. 8311–8315, 2011.

[17] Sanakal S., Jayakumari S.T., "Prognosis of diabetes using data mining approach-Fuzzy C means clustering and support vector machine," International Journal of Computer Trends and Technology (IJCTT), vol. 11, no. 2, 2014.

[18] Orabi, K.M., Kamal, Y.M., Rabah, T.M., 2016. Early Predictive System for Diabetes Mellitus Disease, in: Industrial Conference on Data
Mining, Springer. Springer. pp. 420–427.

[19] Mohammed, A.K., Sateesh, K. P., Dash G. N., 2013, "A Survey of Data Mining Techniques on Medical Data for Finding Locally Frequent Diseases" International Journal of Advanced Research in Computer Science and Software Engineering, 3(8), pp. 149-153.

[20] Sisodia, D., Shrivastava, S.K., Jain, R.C., 2010. ISVM for face recognition. Proceedings - 2010 International Conference on Computational
Intelligence and Communication Networks, CICN 2010 ,
554–559doi:10.1109/CICN.2010.109.

[21] Rish, I., 2001. An empirical study of the naive Bayes classifier, in: IJCAI 2001 workshop on empirical methods in artificial intelligence, IBM.
pp. 41–46.

[22] Khandegar Anjali. Khushbu Pawar diagnosis of diabetes mellitus using PCA, neural Network and cultural algorithm. Int J Digital Appl Contemp Res 2017;5(6).

[23] Iyer A, Jeyalatha S, Sumbaly R. Diagnosis of diabetes using classification mining techniques. Int J Data Min Knowl Manag Process (IJDKP) 2015;5(1).