Ministry of Higher Education
& Scientific Research
University of Babylon
College of Science for Women

Health Care Prediction System for Diagnosis Diabetes based on Machine Learning techniques and Web Service

A Project
Submitted to the Council of the College of Science for Women at the University of Babylon in a Partial Fulfilment of the Requirements for the Degree of High Diploma in Computer Science

By
Rasha Ali Abdoul Raheem

Supervised By
Asst. Prof. Dr. Ali Kadhum M. Al-Qurabat

Feb. 2022 A. D. 1443 A.H

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

((قَالُوا سُبْحَانَكَ لَا عِلْمَ لَنَا إِلَّا مَا عَلَّمْتَنَا
إِنَّكَ أَنتَ الْعَلِيمُ الْحَكِيمُ))

صد ق اللّهّ العلي العظيم

سورة البقرة :الآية 32

DEDICATION

Above all, I would like to dedicate this work Sincerely to Allah Almighty,
And then,
To
The foundation of love and tenderness, by her prayers, miracles are achieved
My mother.
To

all People who have participated in my success

To

Everyone supported me

To
Everyone who loves me

SUPERVISOR CERTIFICATION

I certify that the project entitled "**Health Care Prediction System for Diagnosis Diabetes based on Machine Learning techniques and Web Service"** was prepared by **(Rasha Ali Abdoul Raheem)** and under my supervision at the Department of Computer Science/ College of Science for Women/ University of Babylon as partial fulfillment of the requirements of the degree of Higher Diploma in Computer Science.

Signature:

Name: Asst. Prof. Dr. Ali Kadhum M. Al-Qurabat

Date: / /2022

Address: University of Babylon/ College of Science for Women

THE HEAD OF THE DEPARTMENT CERTIFICATION

I in view of the available recommendations, I forward the project entitled "**Health Care Prediction System for Diagnosis Diabetes based on Machine Learning techniques and Web Service**" for debate by the examination committee.

Signature:

Name: Dr. Farah Mohammed Al-Shareefi

Date: / /2022

Address: University of Babylon/ College of Science for Women

Certification of the examination Committee

We are the member of the examine committee, certify that we have read this project entitled (**Health Care Prediction System for Diagnosis Diabetes based on Machine Learning techniques and Web Service**) presented by the student (**Rasha Ali Abdoul Raheem**) in its content at 31/3/2022, and that in our opinion it is accepted as a project for the degree of higher diploma in science\computer science with a degree **(excellent)**.

Committee Chairman:
Signature:
Name**: Enas Hamood Al-Saadi**
Scientific order: Assist. Prof. Dr.
Date: / /2022

Committee Member:
Signature:
Name: **Ali Y. Yousif Al-Sultan**
Scientific order: Lecture. Dr.
Date: / /2022

Committee Member (Supervisor):
Signature:
Name: **Ali Kadhum M. Al-Quraby**
Scientific order: Assist. Prof. Dr.
Date: / /2022

Deanship authentication of college of science for women.
Approved for the college committee of grade studies.
Signature:
Name: **Faez Ali Rashid Al-Maamoori**
Scientific order: Prof. Dr.
Address: Dean of College Science for Women
Date: / /2022

ACKNOWLEDGMENT

In the name of God, praise be to God, prayer and peace be upon the Messenger of Allah.
All praises and thanks be to Allah for easing my task to accomplish this work despite all the hardships.

I would like to express my deepest sincere thanks and gratitude to my supervisor **Asst. Prof. Dr. Ali Kadhum M. Al-Qurabat**, *whom I'm very proud to work with him, for his supervision, guidance and constant support in accomplishing this project within the limited time frame.*

I'm so grateful to the faculty and the staff of computer science Department, my friends and colleagues for their eligible help and support that played a key role in finishing this project.

I would like to warmly thank and gratitude my mother who has devoted a large part of her life to me. She has supported me So much.

Abstract

Diabetes is one of the dangerous and silent illness. It can occur at any time and may cause great injury to the organs of the body or damage them completely. So, people must investigate this disease at the beginning of its appearance and before it gets hard to treat. With the fast advancement of Machine Learning (ML), these approaches enhanced the efficiency of decision processes in a wide range of applications, including medical diagnostics. In this project, the medical application field has been chosen, and used six supervised learning algorithms and then chose the most efficient model to construct a high-accuracy prediction model for diabetes in humans at an early stage, before it progresses to the point of morbidity or fatality. The suggested model can extract hidden knowledge from diabetes-related data gathered from the Kaggle machine learning repository. This study will also benefit the health industry by offering users an online tool (i.e., web page) that allows them to input data and receive results that predict whether or not the person has diabetes. As a result, prior knowledge and ongoing monitoring of their diabetic health state will lower the risk of complications, morbidity, and death caused by this illness. The suggested system's key contribution is to improve healthcare quality, minimize hospitalizations, and lower the high expenditures of healthcare and drugs.

So, Microsoft Azure ML Studio has been selected to model these ML algorithms. This studio is an extremely powerful visual drag-and-drop authoring environment. There is no need for any particular systems or applications because it is a basic browser-based environment. After running numerous experiments with the classifier models to evaluate the proposed system, several performance indicators, including Recall, Precision, Accuracy, and f1-score, are measured for comparison. Based on the classification output. the algorithms scored accuracy as follows: the SVM scored 0.79, LR scored 80.7, DF scored 81.2, AP scored 78.5, ANN scored 78.5, and the BDT scored 79.6. it was determined that Decision Forest was a best strategy and produces best results than the other ML approaches.

TABLE OF CONTENTS

CHAPTER THREE - THE PROPOSED SYSTEM

CHAPTER FOUR - DESIGN AND IMPLEMENTATION

CHAPTER FIVE: CONCLUSIONS AND FUTURE WORKS

LIST OF FIGURES

LIST OF ABBREVIATIONS

AI	Artificial Intelligence
AML	Azure Machine Learning
AP	Averaged Perceptron
ANN	Artificial Neural Network
AUC	Area Under Curve
BDT	Boosted Decision Tree
BMI	Body Mass Index
CSV	Comma Separated Values
DF	Decision Forest
FN	False Negatives
FP	False Positives
IoT	Internet of Thing
KNN	k-Nearest Neighbors
LR	Logistic Regression
ML	Machine Learning
NB	Naïve Bayes
PID	Pima Indian Diabetes
ROC	Receiver Operating Characteristic
SVM	Support Vector Machine
TN	True Negatives
TP	True Positive
UCI	University Of California Irvine
Wi-Fi	Wireless Fidelity

Chapter One

CHAPTER ONE

INTRODUCTION

1.1. Introduction

Diabetes illness is a long-term illness characterized by high blood sugar. It can cause numerous complications. Thus, according to the increasing sickness during these years, in 2040, the world's diabetic patients will number 642 million, implying that one out of every ten people will be affected by diabetes [1]. Diabetes affects human functions by causing major damage to the eyes, heart, kidneys, and nerves. Diabetes is a long-term illness causes weakness in the body besides it one of the 10 causes of death worldwide [2]. Insulin is a hormone that regulates glucose absorption from the blood into most cells. (Fundamentally Cells of muscle and fat, yet not central nervous system cells). In this way in all types of diabetes mellitus, a deficiency of insulin or the cold-heartedness of its receptors plays a key role [3]. However, genetics and lifestyle may contribute as risk factors. Diabetes must be diagnosed early in order to live a healthy life [4]. And decrease the high costs of patient readmission [5]. Although diabetes is not curable in general, it can be managed with the right treatment [6]. Diabetes impacts organs such as the heart, kidney, nerves, eye, foot, and so on, making it hard for medical practitioners to detect early because of the intricate reliance on numerous elements [7].

Through the topic of prediction, several researches have been carried out for several diseases recently, to the level that some of today's clinicians now make use of machine learning models to predict different diseases. It is, therefore, imperative to design a diabetes classifier that is convenient, accurate and cost efficient. Artificial Intelligence techniques provide a wide range of ideas that are useful to human related fields of application like, medical diagnosis which is a process where a physician has to analyze lot of

factors before diagnosing diabetes which makes the physician's job difficult and time-consuming. Machine learning and data mining techniques have been considered very helpful in the design of automatic diagnosis system for various health conditions.

In recent times, many methods and algorithms have been discovered which can be used to mine biomedical datasets for hidden information including supervised learning techniques like Neural networks (NNs), Decision Forest(DF), Support Vector Machines (SVM), Decision Trees (DT), Naïve Bayes, and logistic regression; unsupervised learning techniques like clustering analysis, pattern recognition and image analysis; and reinforcement algorithms which are applied in the field of game theory, control theory and decision theory. This project intends to develop a prediction model with high degree of accuracy for diabetes in people at an early stage, before it becomes escalated to a point of morbidity or mortality using some supervised learning algorithms. This project will also contribute to the health sector by providing people with accurate prior knowledge about their health status as related to diabetes hence, reducing the rate of complications, morbidity and mortality being caused by this disease [3].

1.2. Project Problem

The prevalence of diabetes is on the increase across the world, the International Diabetes Federation has noted that; There are currently 382 million people living with diabetes in the world, with this number expected to double by 2035 equally five hundred and ninety-two (592) million. The increase might be due to the gradual adoption of unhealthy lifestyles and the aging of the population without preparedness for prevention and control, throwing up so many challenges to diabetes care which has now become a major health problem in most countries around the world. This issue, it is also possible attributed to such as “will the patient contract diabetes?”

Because the output might be either yes (1) or no (0), it's a two-class categorization issue. Though ML is becoming more regularized within the healthcare sector it can still be challenging to determine which algorithm to use for the data that has been collected. Several factors need to be considered when choosing algorithm, the size of the data, quality, and type of data.

1.3. Project Objective

1- Replace current diabetes diagnosis methods with modern technologies as our proposed system that predict the occurrence of diabetes which is save time.
2- The proposed system intends to employ various machine learning algorithms, such as the Support Vector Machine and the Decision Tree, among others that can save time and get more accurate results by using these algorithms to predict diabetes at the beginning of his appearance by using proposed system.
3- Comparing the performances of many algorithms and in the end, using the most efficient model.
4- Deploying the proposed ML diabetic's prediction model as a web service.

1.4. Related Works

As noted in the paragraph, this part will cover earlier works on the topic of diabetes mellitus prediction and diagnosis using machine learning.

Quan et al. (2018), To predict diabetes mellitus, researchers used Boosted decision trees, random forests, and neural networks. A model were tested using five-fold cross validation in this study. They chose several methods with superior performance to conduct independent test experiments in order to verify the methods' universal applicability. The results indicated that random forest prediction had the highest accuracy when all of the criteria were used. (ACC = 0.8084) [1].

Ismail et al. (2021), The present study helped allied health professionals and researchers in the field of diabetes prediction. For an imbalanced dataset, data balancing techniques could help to detect the minority class. different machine learning based prediction frameworks for healthcare and diabetes in particular were analyzed. The authors implement and evaluate the decision tree (DT)-based random forest (RF) and support vector machine (SVM) learning models for diabetes prediction as the mostly used approaches in the literature using this framework and The results indicated that random forest and SVM and DT predictions had the highest accuracy when all of the criteria were use and the accuracy was (ACC = 0.7800) [2].

Dey et al. (2018), The proposed a web based application for the successful prediction of Diabetes Diseases. The proposed system using machine learning algorithms, it has significant potential in the field of medical science for the detection of various medical data accurately. they have used and tested some sort of powerful machine learning models like SVM, KNN, Naive Bayes and ANN. For the successful evaluation of these models From different machine learning algorithms Artificial Neural Network (ANN) provides highest accuracy with Min Max Scaling Method on Indian Pima Dataset reach for 82.35% [8].

Aminul et al. (2017), investigated and compared various types of ML classification algorithms. The aim of this study is to use the results of machine learning classification algorithms to detect the start of diabetes in diabetic patients. logistic regression had the highest accuracy (78.01%) [9].

Jimmy et al. (2019) , the intent of their study is to collect and use data of diabetic patients who have either been readmitted or not readmitted and feed the data into four ML algorithms to determine which algorithm performs the best. Comparisons about true/false positives, training scale and accuracy are

performed on K-nearest Neighbor, Boosted decision tree, Logistic regression, and neural Networks. Accuracy for each algorithm (balanced data) was 78.5 [5].

Ramana et al. (2018), proposed model has ability to extract the hidden information from a huge amount of diabetes-related data gleaned through Web services archive of data. The evaluation experiment gives real time blood glucose level, which is, predicted on various lifetime events and it intakes insulin and measures from dynamic scenarios such as class - boosted tree algorithm as well as regularization, proportion of blood glucose levels shows diabetes positives, which are correctly predicted to 90% accurate using prediction function. Else the true negative rate, which measures the proportion of negatives that are correctly, identified percentage of NO diabetes. accuracy in diabetes prediction is 0.9 (90%) [3].

Clodagh (2019), Proposed predictive modeling as a method for detecting patterns in historical data. It learns from these patterns so that it can generate predictions automatically when new data becomes available. Predictive models include LR, BDT, SVM, NN, DF. The best performing model is selected to create a web service where users can input data and receive scored results connecting the user to the data. Metrics include accuracy, recall and AUC to measure the performance of the models where a Boosted Decision Tree achieved the highest results. Hospital readmission accuracy at 86% and diabetes diagnosis accuracy at 67% [10].

1.5. Project Outline

In addition to the current chapter, this research includes other chapters, as listed below**:**

Chapter 2 describes the theoretical background of prediction by machine learning.

C**hapter 3** presents the proposed system.

Chapter 4 covers Implementation and Results.

Chapter 5 presents Conclusions and Future Works.

Chapter Two

CHAPTER TWO

Theoretical Background

2.1. Introduction

This chapter provides background information on the project. To begin with, diabetes is a well-defined disease. The concept of Machine Learning is then described. After that, a depiction to the machine learning methods used in our research was provided. Lastly, methods for assessing algorithms of machine learning are defined.

According to the American Diabetes Association (ADA), "Diabetes mellitus (MEL-ih-tus), or simply, diabetes, is a group of diseases characterized by high blood glucose levels that result from defects in the body's ability to produce and/or use insulin". There are two prevalent forms of diabetes, Type 1 (T1DM) and Type 2 (T2DM). T1DM occurs when the body is no longer able to produce insulin. Onset of T1DM is common in childhood; this disease used to be known as juvenile diabetes. This form of diabetes is less common; only about 5-10% of people with diabetes have T1DM. T2DM occurs when the body is unable to utilize the insulin produced or not enough insulin is produced. T2DM is commonly associated with obesity; however, obesity is not the only high risk factor. Certain ethnicities are considered to be high risk groups, as large percentages of those ethnicities have diabetes.

Patients with diabetes need to control their blood glucose levels. Insulin or other medication may be used to control blood glucose levels. If blood glucose levels are not adequately controlled, the long term complications can be quite costly in terms of both health and finance. Such complications include increased risk for heart disease and stroke, blindness, kidney failure, and even death [11] [1-3].

2.2. Machine learning

Machine learning is a field of computer science – gives the ability to the computer to learn without being explicitly programmed. ML teaches computers to identify and recognize real-world things by using examples rather than instructions [12] It also enables them to learn from their own wrong decisions through trial - and - error. ML gives computers the ability to make their own decisions once they have learned enough about a subject. The goal of machine learning is to create computer systems that can adapt and learn from their experiences [7]. The field of machine learning is attentive with the development of algorithms and techniques that enable computers to learn and gain intelligence based on previous experience [4]. Now days, machine learning can answer queries. One of the missions is a prediction on disease data [13]. Machine learning methods are widely used in predicting diabetes, and they get preferable results [1].

2.3. Types of Machines Learning

Machine Learning can be divided into three categories: supervised, unsupervised, and reinforcement learning.

2.3.1. Supervised Learning

"In supervised learning, each data point is labeled or associated with a category or value of interest". As demonstrated in (Figure 2.1.), categorizing an image as a 'cat' or a 'dog' is an example of a categorical label. The sale price of a used car, for example, is an example of a value label [12]. The purpose of supervised learning is to analyze a large number of labeled samples, such as these, and then initiate predictions about future data points [14]. The system must "learn" inductively a function called target function,

which is an expression of a model characterizing the data, in supervised learning. The goal function is used to predict the value of a dependent

variable, known as an output variable, from a set of variables, moreover known as independent variables, input variables, characteristics, or attributes. The function's set of possible input values [7]. The goal is then for the machine to discover a relation between the input and the output value based on the given values. These models can be divided into classification tasks and regression tasks.

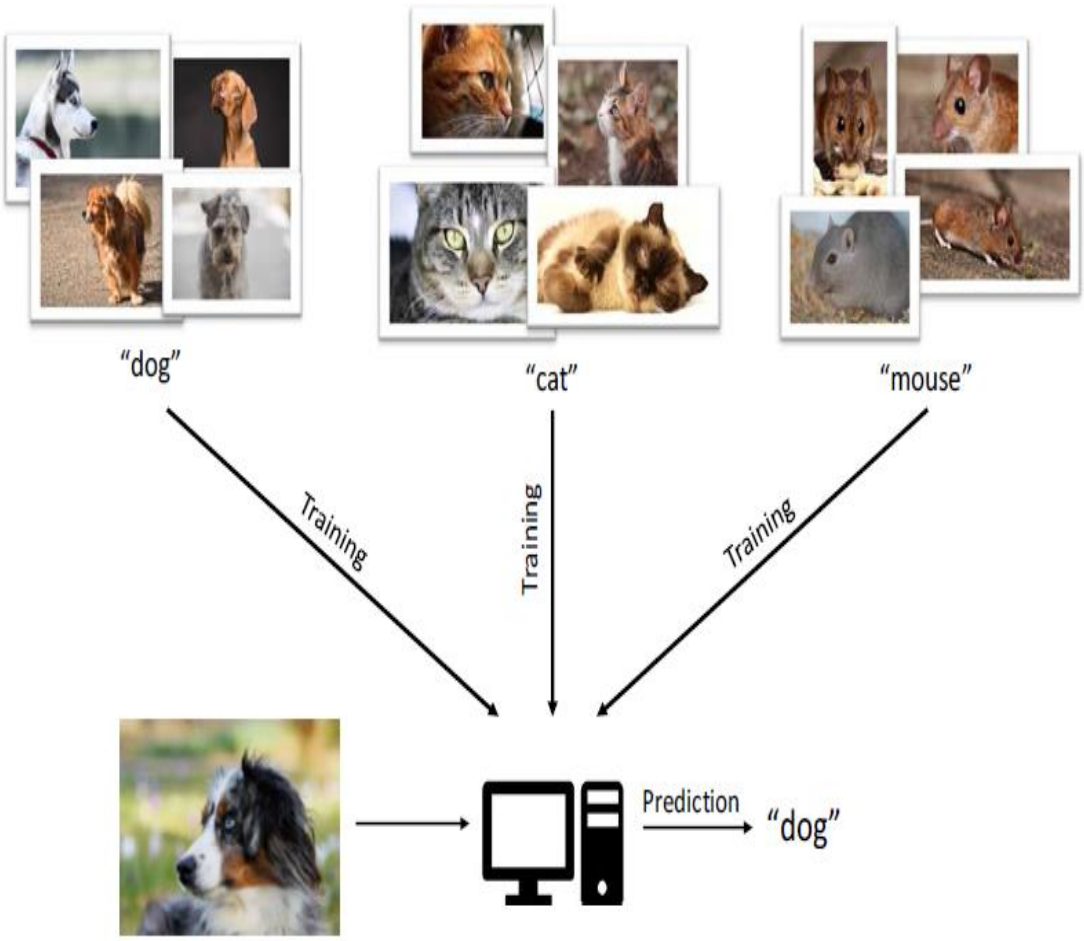


Figure 2.1. An example of supervised ML classification [12].

2.3.2. Unsupervised Learning

In unsupervised learning There are no titles associated with data points. As an alternative, Unsupervised learning algorithms are used to organize or define data in such a way [14]. The training data that is available is unlabeled. There is no previous information or training given to the system. In order to make decisions or predictions, the algorithm must investigate and detect

patterns from the supplied data [4-7]. Examples of unsupervised learning are: estimating what the typical person looks like in a specific neighborhood based on features such as hair color or eye-color.

2.3.3. Reinforcement Learning

Here, a machine is not explicitly supplied training data. In order to reach a goal, it must interact with the surroundings. Due to a lack of training data, it must learn from the ground up, relying on a trial-and-error method to make decisions and find its own correct paths. here is a consequence for every action the machine makes, and it is rewarded numerically for each consequence. This machine learning technique is extremely beneficial when dealing with highly dynamic circumstances were collecting and supplying training data is impossible [12].

2.4. Problems that Machine Learning Solves

ML has factually unleashed an endless number of ways for machines to complete tasks. Traditional programming techniques would not be able to accomplish it. Fortunately, all of these issues can be divided into a few different categories. These solutions include:

1- **Classification**
2- **Regression**
3- **Clustering**
4- **Anomaly Detection**

The proposed system will have concerned with classification algorithms.

2.5. Machine learning Algorithms

Six supervised algorithms have been selected, which are concerned with classification, in order to classify the patient as either 1 means diabetic or 0 means non-diabetic.

A common question is "*Which machine learning algorithm should I use?*" The algorithm selection depends primarily on two different aspects of data science scenario:

- What you want to do with your data? Specifically, what is the business question you want to answer by learning from your past data?
- What are the requirements of your data science scenario? Specifically, what is the accuracy, training time, linearity, number of parameters, and number of features your solution supports?

This project is concerned with classification algorithms using Decision Forest, Support Vector Machine, Average Perceptron, Boosted-Decision-Tree, Logistic- Regression, and Neural Network algorithms as shown in Figure 2.2.

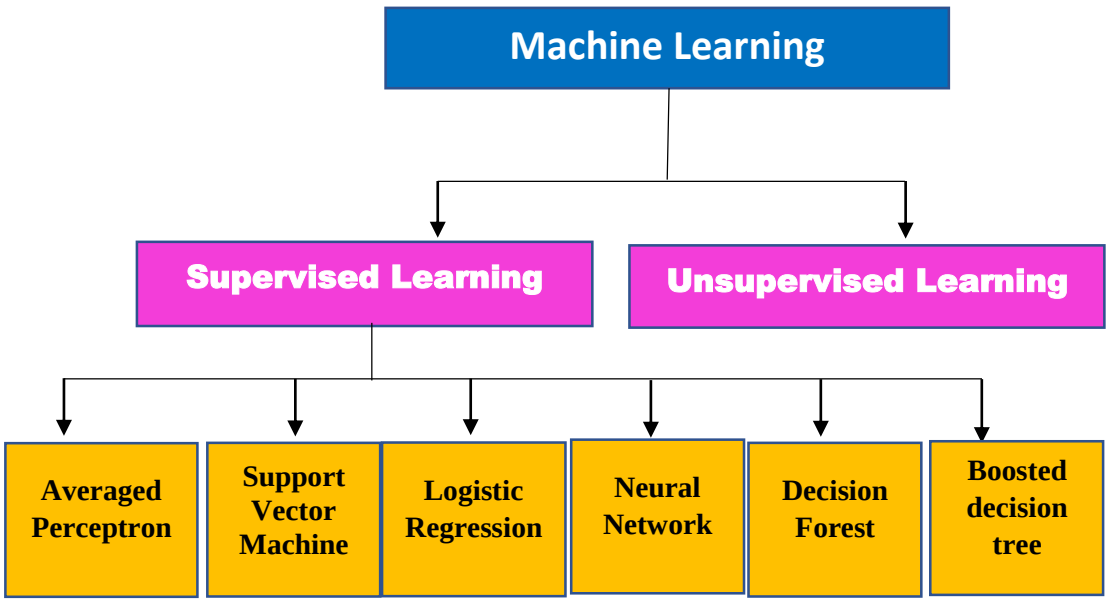


Figure 2.2. The ML used algorithms

2.5.1. Logistic Regression (LR)

Logistic regression is a type of probabilistic statistical classification model for analyzing a dataset in which there are one or more independent variables that determine an outcome [9]. LR is a successful algorithm to use with our dataset as it had an accuracy of 80.17% for the same dataset, and other similar studies with similar dataset resulted with a 79% accuracy [5]. it only contains data coded as 1 (TRUE, success, pregnant, etc.) or 0 (FALSE, failure, non-pregnant, etc.) [9].

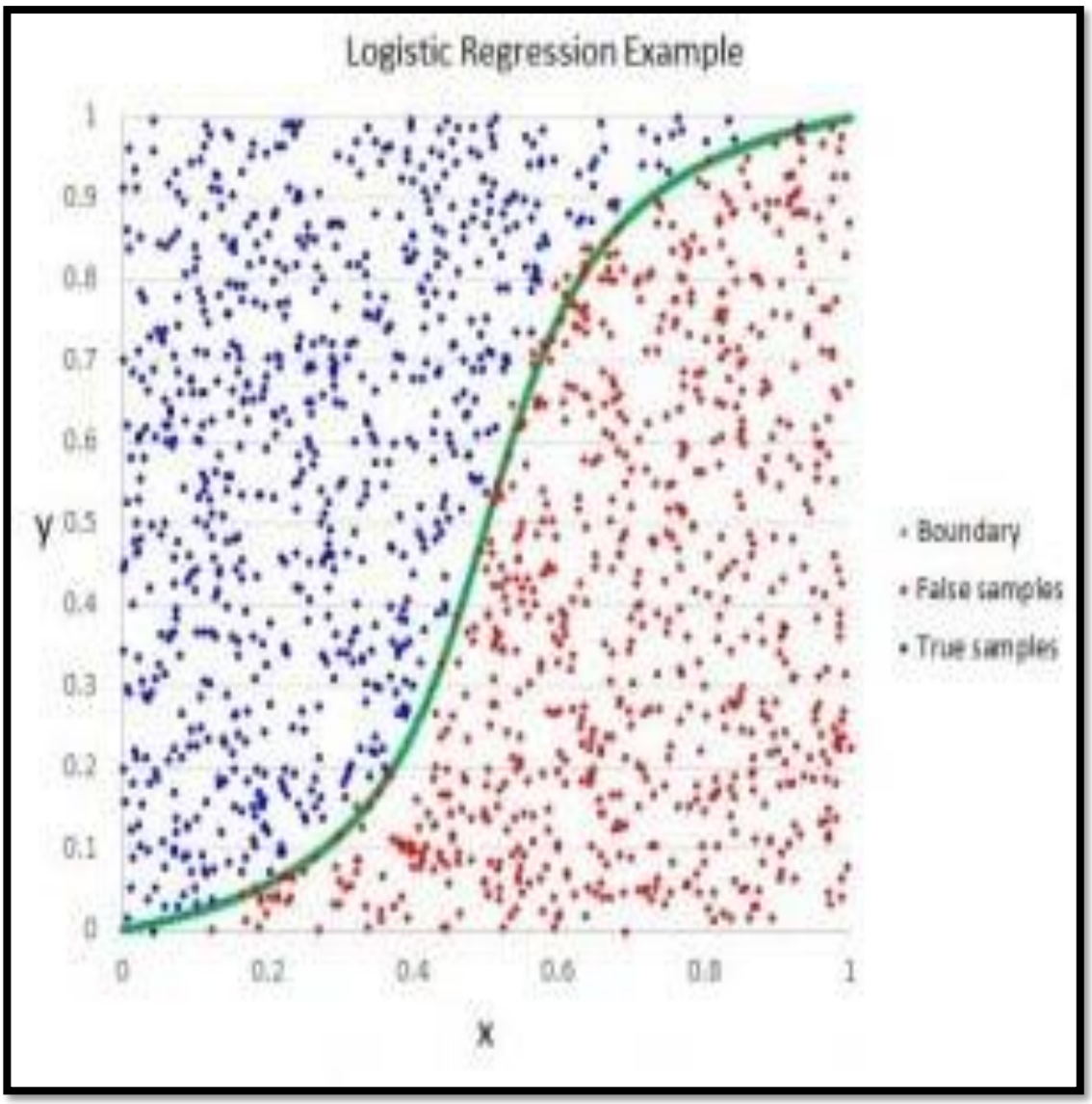


Figure 2.3. Shows Logistic Regression algorithm.

2.5.2. Artificial Neural Network (ANN)

Because of the vast amount of input data and only two possible outputs the ANN architecture is ideal for the dataset used in this study. The dataset already has the answer to the diabetic question; therefore, the supervised training model will be used. The input layer, hidden layer, and output layer are all significant aspects of a neural network. as shown in Figure 2.4. The input layer is in charge of receiving input data. The results can be obtained from the output layer. The hidden layer is the layer that sits between the input and output layers. Because they can't be seen from the outside. On the same layer, there is no connectivity between neurons [1,5].

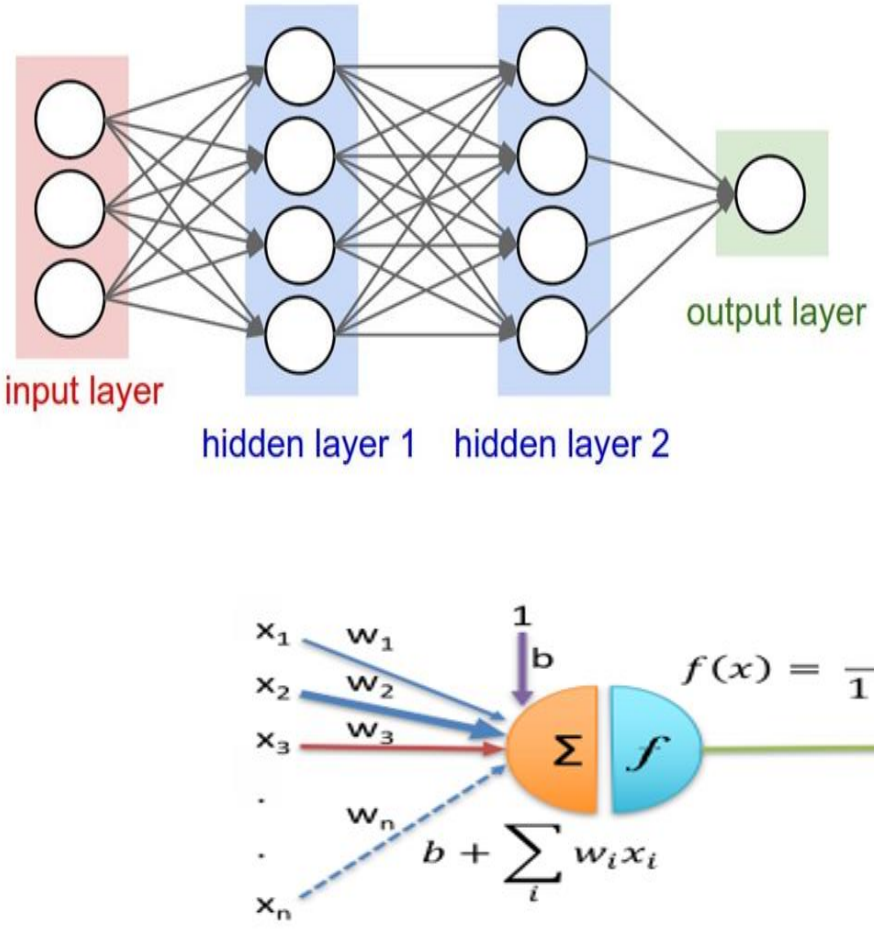


Figure 2.4. Shows the neural network algorithm.

2.5.3. Boosted Decision Tree (DT)

A decision tree classifies data into discrete categories using tree structure algorithms. The primary goal of decision trees is to reveal the structural information contained in data. The decision tree method is a supervised machine learning technique that builds a decision tree from a set of class labeled training samples during the machine learning process. as shown in Figure 2.5. The training samples and their associated class labels are used to start the decision tree algorithm. This training set is recursively partitioned into subsets depending on feature value, with each subset's data being purer than the parent set's data. Each internal node in a decision tree represents a test on the attribute (feature), each branch represents an outcome of the test and each leaf node represents the class label. When a classifier decision tree is used to determine the class label of an unknown sample, tracing a path from root to the leaf node, which holds the class label for that sample [15]. Machine-learning uses a decision tree as a predictive model which maps observations about an item to conclusions about the item's target value. More descriptive names for such tree models are classification trees or regression trees. Decision tree classifiers usually employ post-pruning techniques that evaluate the performance of decision trees, as they are pruned by using a validation set. Any node can be removed and assigned the most common class of the training instances that are sorted to it [16-17].

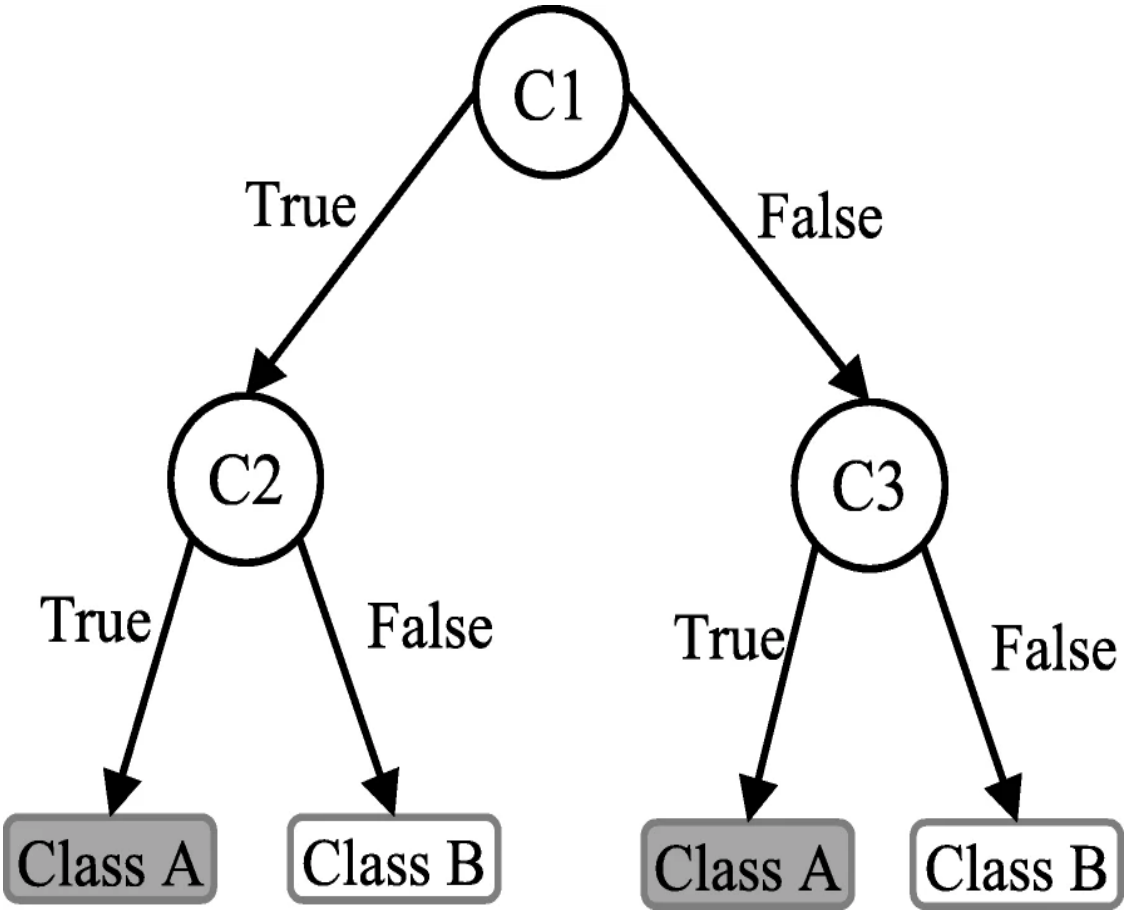


Figure 2.5. Shows Boosted decision tree algorithm.

2.5.4. Averaged Perceptron (AP)

Other well-known algorithms are based on the notion of perceptron. Perceptron (as shown in Figure 2.6) can be succinctly defined as: If X_1 through X_n are input feature values and w_1 through w_n are connection weights/prediction vector (typically real numbers in the interval [-1, 1]) then perceptron computes the sum of weighted inputs $\sum X_i w_i$ and output goes through an adjustable threshold: if the sum is above threshold, output is 1; else it is 0. The perceptron approach has been most typically used to learn from a batch of training sample by frequently applying the algorithm across the training set until it finds a prediction vector that is correct across the time. The labels of the test set are then predicted using this prediction rule [18].

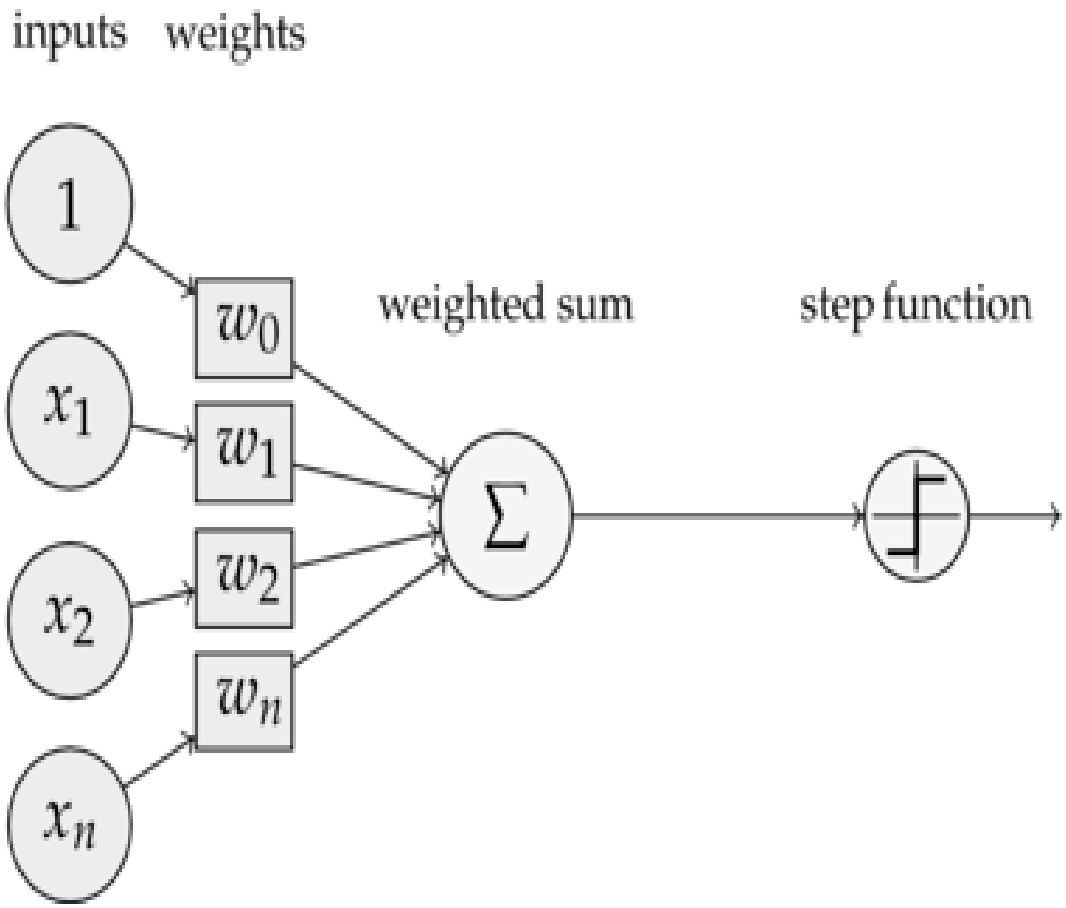


Figure 2.6. Shows the perceptron algorithm.

2.5.5. Decision Forest (DF)

The goal of a decision forest is to improve a single decision tree's predictive performance by training several trees and aggregating their predictions [19]. Building a decision forest is one method to fully utilize the possibilities of decision trees. As the name implies, a decision forest is made up of numerous decision trees whose predictions are combined to form a single final forecast. By constructing a forest, a single decision tree's errors are mitigated by the forest's other decision trees [19]. A Random Forest (RF) is a combination classifier made up of many DTs, like to how a forest is made up of several trees. Figure 2.7 represent an illustration of the Random–Forest algorithm [20, 21].

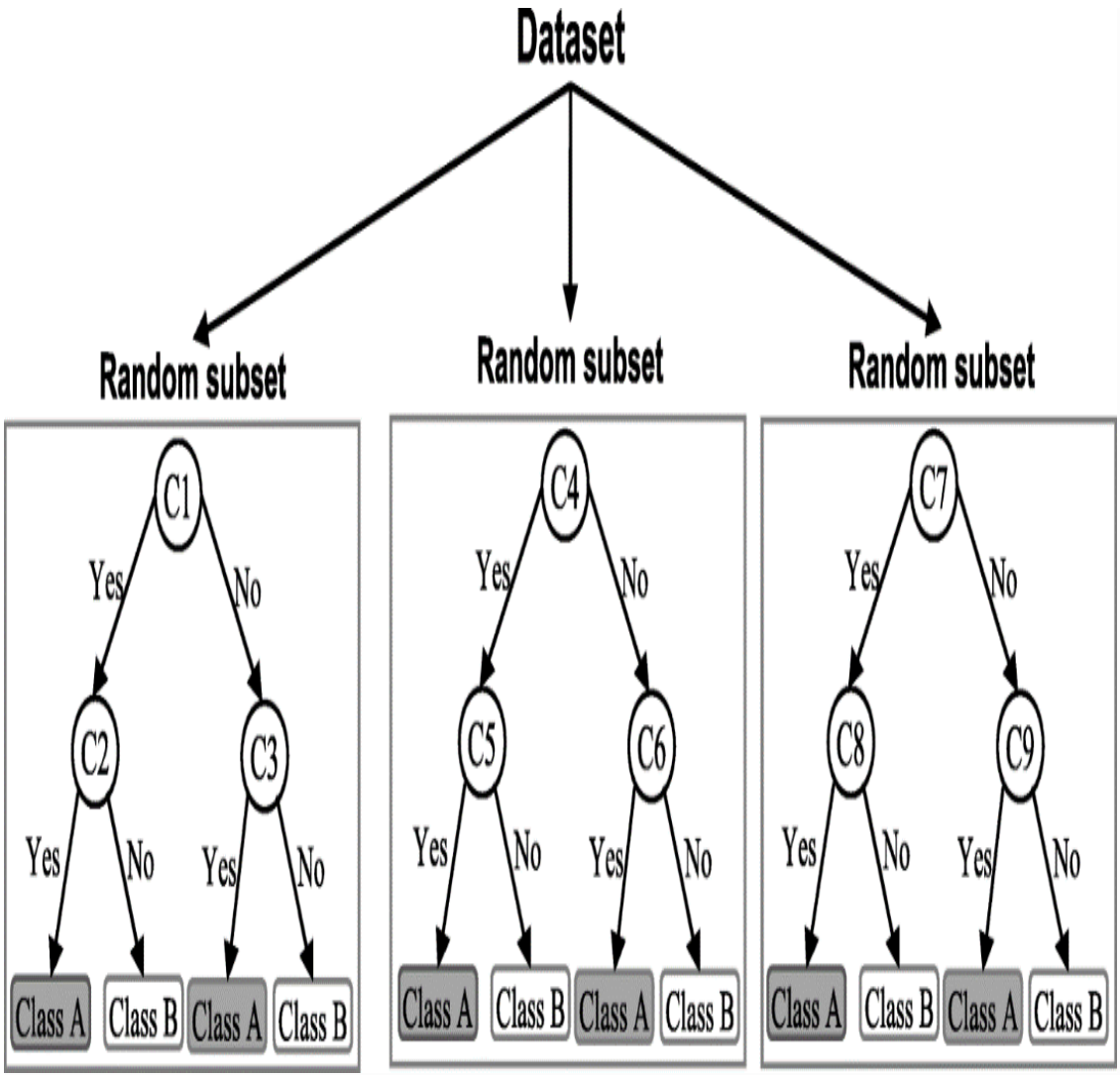


Figure 2.7. Decision forest.

2.5.6. Support Vector Machine (SVM)

The Support Vector Machine (SVM) is utilized in each of classification as well as regression. In SVM model, the data points are represented on the space and are categorized into groups and the points with similar properties falls in the same group. In linear SVM the given data set is considered as A p-dimensional vector that can be separated by maximum of p-1 planes called hyper-planes. These planes separate the data space or set the boundaries among the data groups for classification or regression problems as shown in Figure 2.8. The optimal hyper-plane could be chosen from a large number of hyper-planes based on the distance that splits the two

classes. The maximum hyper-plane is the plane with the largest margin between the two classes [4, 22].

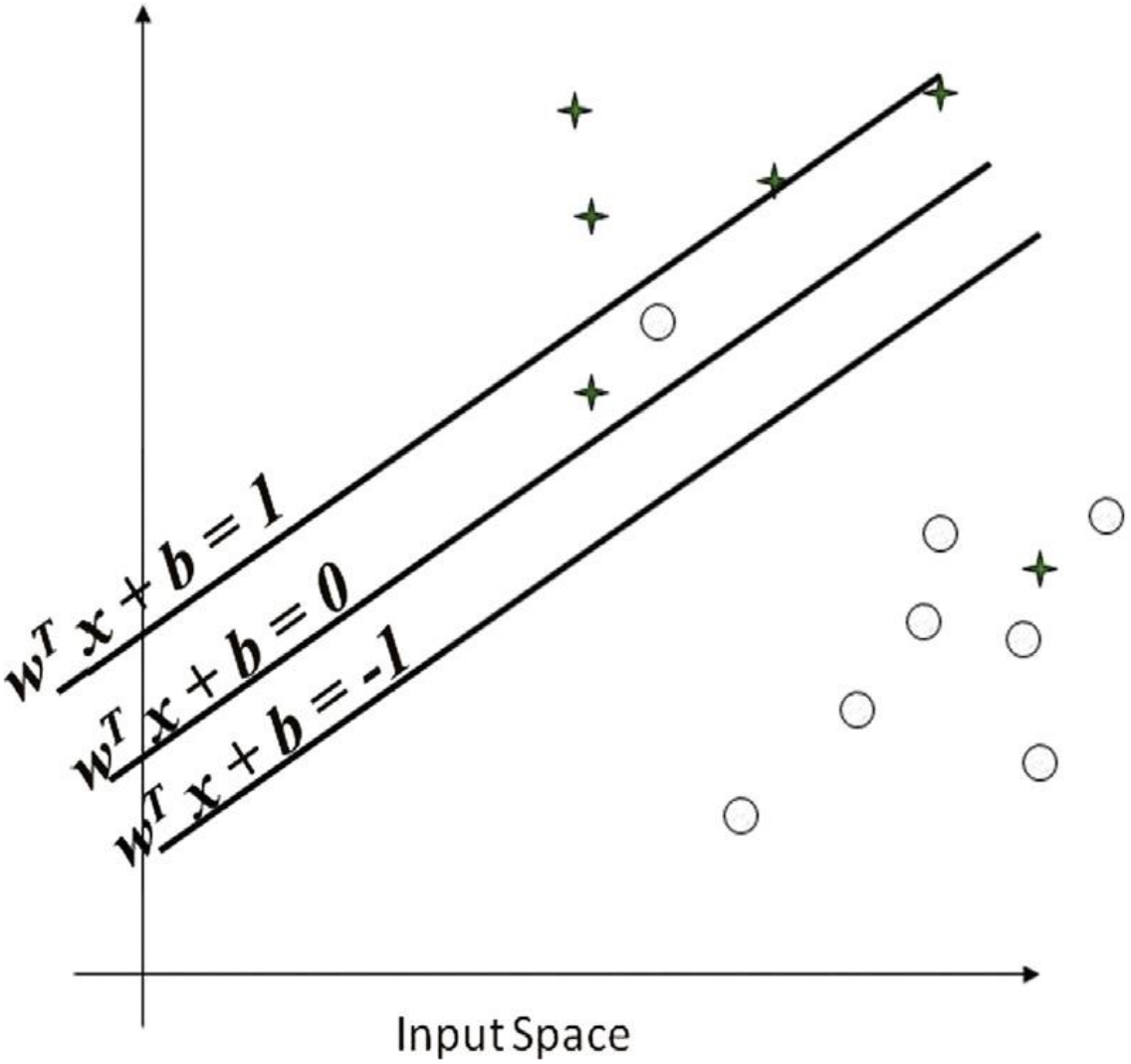


Figure 2.8. Support vector machine.

2.6. Performance Evaluation Techniques

A confusion matrix is a technique for emphasizing the performance of a classification algorithm. When there are more than two classes in a model, or when each class has an unequal number of examples, accuracy alone can be perplexing. One benefit of a confusion matrix is that it's simple to detect if the system is unable to differentiate between two categorizes. (i.e., frequently mislabeling one as another). Table 2.1. shows the confusion matrix of classification algorithm.

TABLE 2.1. Confusion Matrix of Classification Algorithm.

		Prediction	
		0	**1**
Actual	**0**	**TN**	**FP**
	1	**FN**	**TP**

- **True Positive (TP):** When the actual classification of the data element is marked true also the ones that were predicted are likewise true at the finale
- **True Negatives (TN):** When the data point's actual class is false and the anticipated ones are also false at the end.
- **False Positives (FP):** When the data point's real class is False, but the predicted ones are true in the end.
- **False Negatives (FN):** When the real class of a data item is true, but the expected classes are false in the final.

Sensitivity and specificity are two objective factors in measuring the performance of the classifiers which were also employed in the study. Sensitivity is also referred to as the rate of recall, and measures the proportion of actual positives predicted correctly as such; the percentage of people who are correctly predicted as having a disease. Specificity determines the number of negatives that are accurately predicted as the percentage of healthy people who are accurately classified as not having the disease. To identify the best classifier in diabetic's disease classification, some of the performance parameters need to be selected to get a better outcome. As far as calculating the best results, the higher the Accuracy value, the better the classifier is. The measurements below are used to define the performance metrics that are widely used in machine learning.

- **The Accuracy:** Accuracy is determined as the number of correctly predicted instances divided by the total number of instances. This implies that accuracy is the percentage of correctly predicted positive cases within the total cases.

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \times 100$$

- **The Precision:** Precision refers to the consistency or precision of a true positive class , so regarded as a predictor with a positive value. This is the number of occurrences as a percentage. that accurately have classified x / Total forecast as class x. So, the accurate findings were reported essentially in high precision and it takes all related data but just returns the highest performance.

$$Precision = \frac{TP}{TP + FP} \times 100$$

- **Recall:** Recall provides problem responsiveness and it processes values or quantity or completeness of the component. It returned the most relevant documents or results that are closely related to the problem. In simple words, for all the positive classes, how many are accurately predicting. This should be as raised as practically possible.

$$Recall = \frac{TP}{TP + FN} \times 100$$

- **F1_Score :** The F1_score is a calculation that attempts to achieve a balance by combining the percentages of accuracy and recall.

$$F1 - Score = \frac{2 \times Precision \times Recall}{Precision + Recall} \times 100$$

- **AUC-Score:** AUC A curve is a representation of performance at various thresholds, for classification problems. AUC reflects the degree of separability measurement. It shows how much a model can differentiate between classes[21, 23].

Chapter Three

CHAPTER THREE

THE PROPOSED SYSTEM

3.1 Introduction

In this chapter, we proposed a system based on machine learning that uses a dataset and different machine learning algorithms with supervised, tends to predict whether diabetic or non-diabetic.

A Machine Learning model is proposed to illustrate the flow of research, To predict the outcomes for diabetic patients, several steps have to be taken to construct a good model for decision-making. Figure 3.1 displays eight chief steps along with some sub-step to build and carry out the final model. The first step of this research was Data collection, which required finding and download dataset related to the problem. This project used an available diabetes dataset from Machine Learning repository Kaggle named "Pima Indian diabetes". The following step is the second, the data was pre-processed since it can include many gaps, lost values, and outliers. The third step is, the dataset was analysed in terms of description and split into two groups as per the model required. One for training the model and the other for scoring data is reserved for later testing the trained model to calculate its loss, the data divided as (75 – 25) as will be explained later. In the fourth stage, different Machine Learning algorithms were executed on the training data set as per different parametric presentation. In the fifth stage, the execution was done in Microsoft azure studio environments. In the sixth stage, the results were interpreted and analyzed for different algorithms in Microsoft azure environments. In the seventh step, the score process will be done for each algorithm and the result of predicting will appear. The final step, is the process of comparing among algorithms will take place to determine the most efficient model for deploying it in the web service.

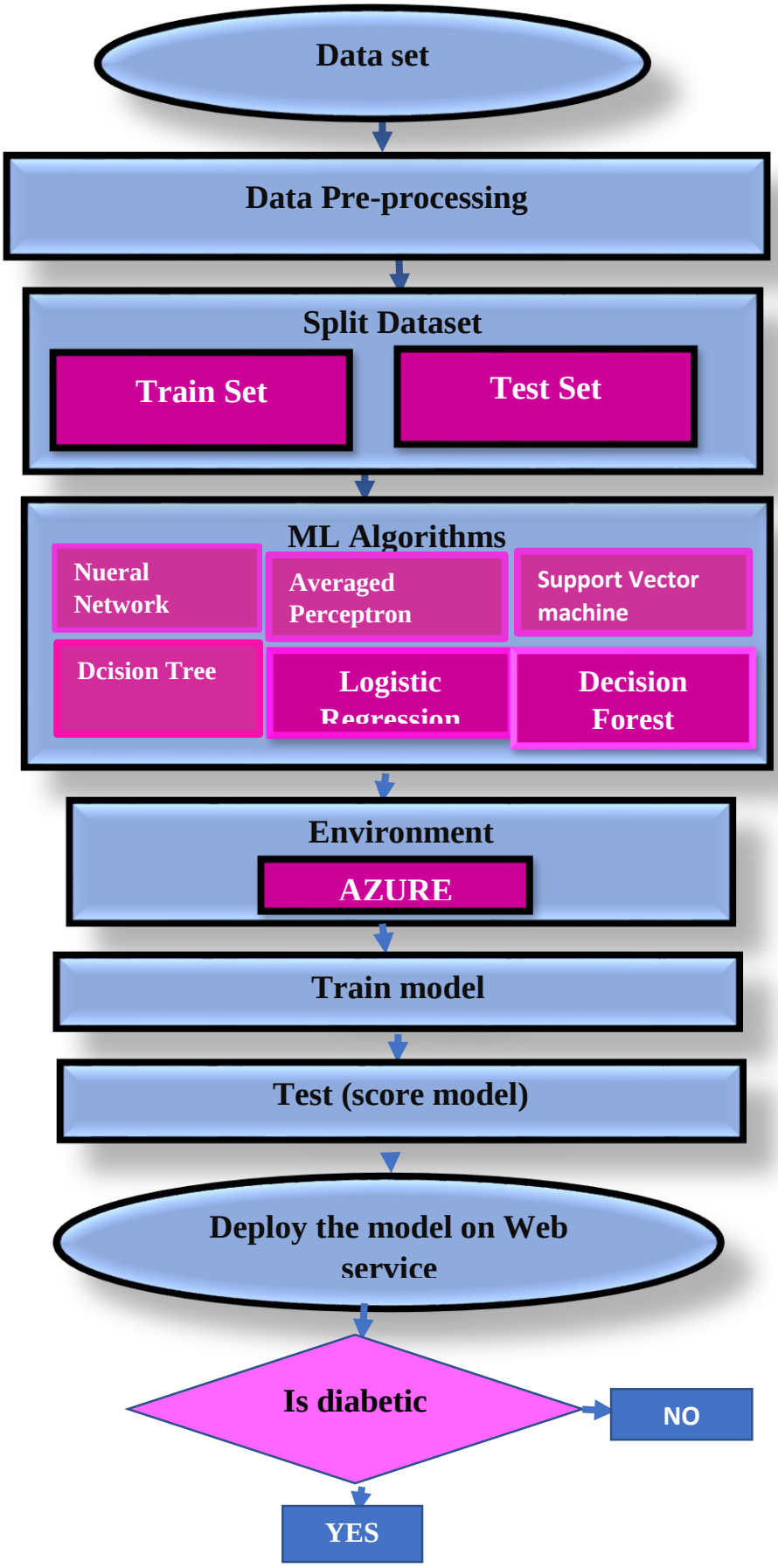


Figure 3.1. Proposed Model.

3.2. Selection of Datasets

In today's world, there is a huge amount of data available through different resources like the internet, surveys, experiments…etc. There are different repositories available for Data Science such as UCI, Data World, and Kaggle, etc. which are well-known to provide datasets suitable for Machine Learning Algorithms in almost areas. The most suitable and well-known dataset available on all three repositories UCI, Data World, and Kaggle for Machine Learning purposes was the "Pima Indian Diabetes (PID)" dataset.

This dataset is broadly used to experimentally understand the data by the Machine Learning community from newbies to professionals. This dataset is standardly published in various research papers and studies for various Machine Learning Algorithms.

The PID data set has been created by collecting the information among the Pima Indian female population near Phoenix, Arizona. The PID repository includes 768 Indian female patients. The conditional response factor takes two classes ' 0 ' or ' 1 ' values, where ' 1 ' is a positive diabetes mean diabetic test and ' 0 ' is a negative diabetes result mean non diabetic. Class ' 1 ' has 268 (37.8%) cases and Class ' 0 ' has 500 (62.2%) cases

In this data set eight medical factors are evaluated to decide the occurrence of diabetes as class '1' or class '0' with no missing or out-liars reported. These medical factors considered to be high risk for diabetes. The 8 factors or attributes are listed in tables 3.1 with their data types.

Table 3.1. Attributes of PID Dataset

No.	List of Attributes	Range (Min – Max)	Description
1.	No. of Times Pregnant	0 – 17	Pregnancy frequency
2.	Serum Insulin	0 – 846	2 Hours Serum Insulin values obtained
3.	Tri-Skin-Fold thickness	0 – 99	Thickness Fold of Triceps Skin
4.	Diastolic BP	0 – 122	Diastolic Blood Pressure
5.	Plasma Glucose	0 – 199	Tolerance values of Oral Glucose Test
6.	AGE	21 – 81	Age
7.	Diabetes pedigree	0.078 – 2.42	Functions of Diabetes Pedigree
8.	BMI	0 – 67.1	Body Mass Index
9.	Outcome (class variables)	0 – 1	Positive and Negative

3.3. Creating an Account

It's not necessary to have an Azure subscription to use AML Studio. It can quickly create a Microsoft Azure Studio account. After logging in to AML Studio, choose Blank Experiment, and a new screen appears, which can be saved to the workspace file for free that lets us use it as a Launchpad for our experiments. The workspace appears entirely empty during this stage.

3.3.1. Importing Data

Data is the most important part of a machine learning solution. The very first thing to do is import data that will help us train our model. To be able to predict diabetes in apatient, Fortunately, there are online repositories that

provide access to quality datasets for free. You can download a dataset from intenet and later import it inside ML Studio, which provides support for importing data in a lot of formats such as CSV, Excel, SQL, etc. you must download dataset concern with diabetes desies. Drag and drop the dataset to

It's a good idea to explore what's inside the dataset. That will not only give you a view of the data that Studio will be using for training, it will also help you in picking the right features for predictions. In experiment canvas, right-click the dataset and select dataset ➤ Visualize. You can also click the dataset's output port (the little circle at bottom-middle) and click Visualize.

3.3.2. Data Pre-processing

Data pre-processing has a significant role when the research study is based on a model to understand the data. Because mostly the data to be analyzed is full of outliers, null values, and special characters. the preprocessing operation include removing all these bad data. Regardless of how well data is constructed, if data contains poor-quality input, the resulting output will always be poor. Therefore, data pre-processing is a very necessary step to start with data analysis and machine learning.

Pre-processing significantly reduces the size of the input data. The cleaning of data in this project depends on three key criteria for data:

i. **Relevancy**—All columns (features) are related to each other and to the target column.
ii. **Connectedness**—There is zero or minimal amount of missing data values.
iii. **Accuracy**—Most of the rows have accurate (and optionally precise) data values.

This dataset was containing many missing values and those were shown with zeros "0". By performing Exploratory Data Analysis (EDA) using Pandas

library from Python it was observed from the histograms that there are many rows containing zero and that's impossible. For example, let's consider Blood Pressure, in some rows it is 0 (impossible), even minimum Blood Pressure can never be zero as shown in Figure 3.2

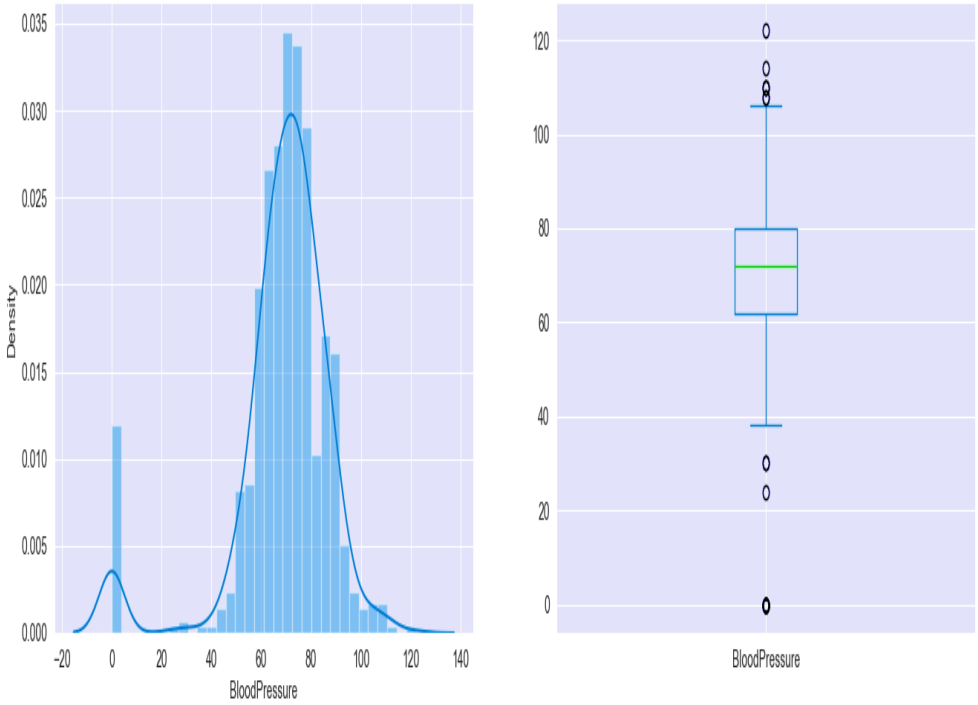


Figure 3.2. the Histogram for Blood Pressure.

And also the Histogram of others features as follows :

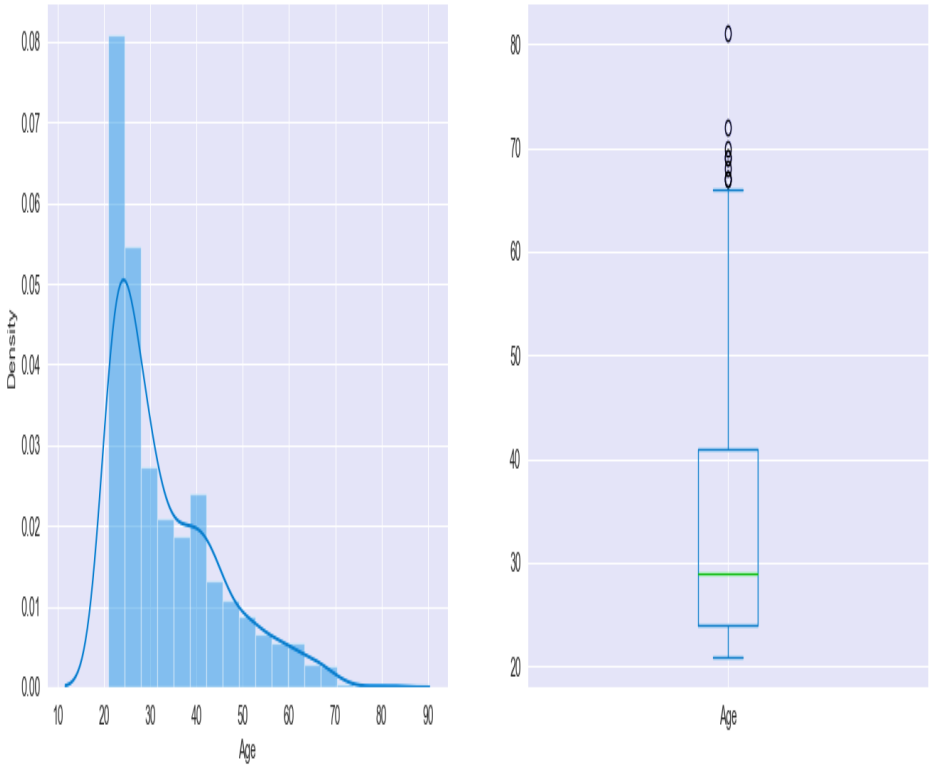


Figure 3.3. the Histogram for Age.

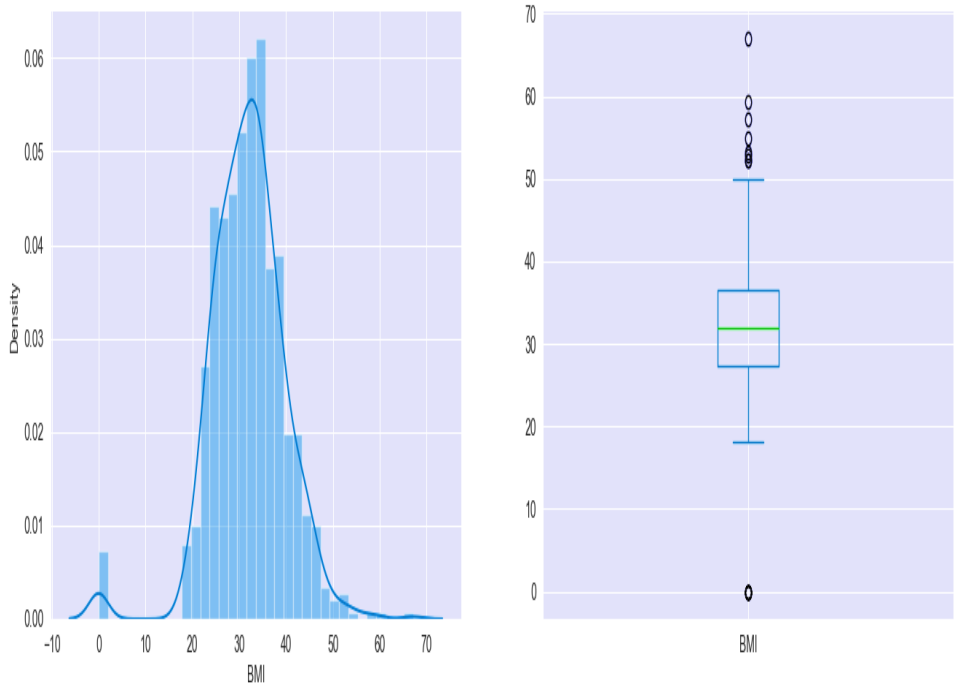


Figure 3.4. the Histogram for BMI.

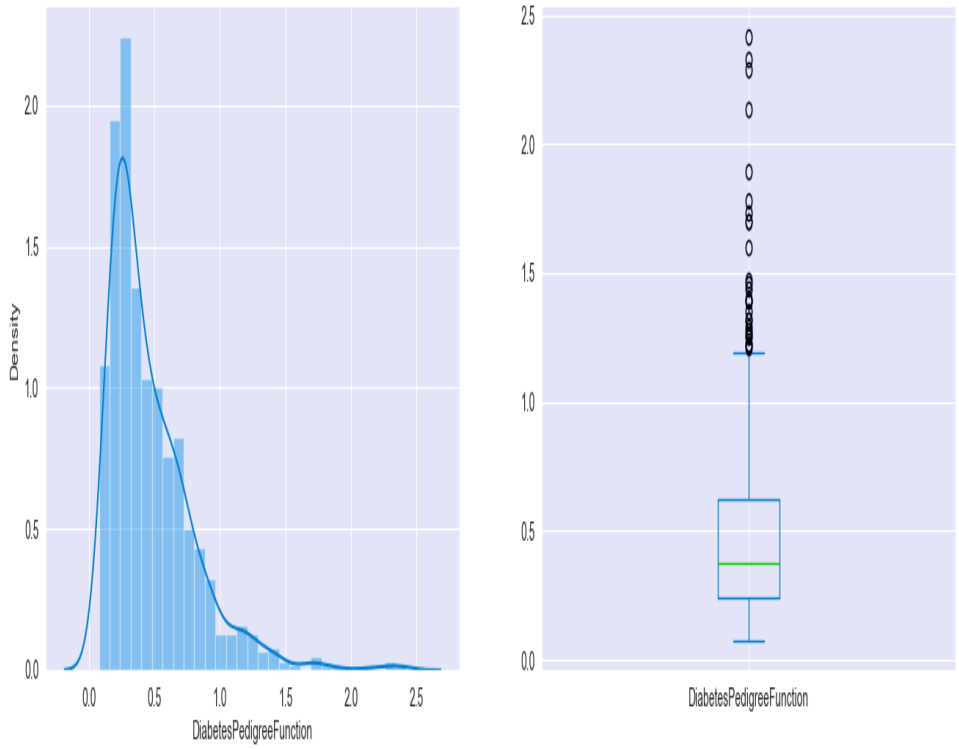


Figure 3.5. the Histogram for the Diabetes bedigree function

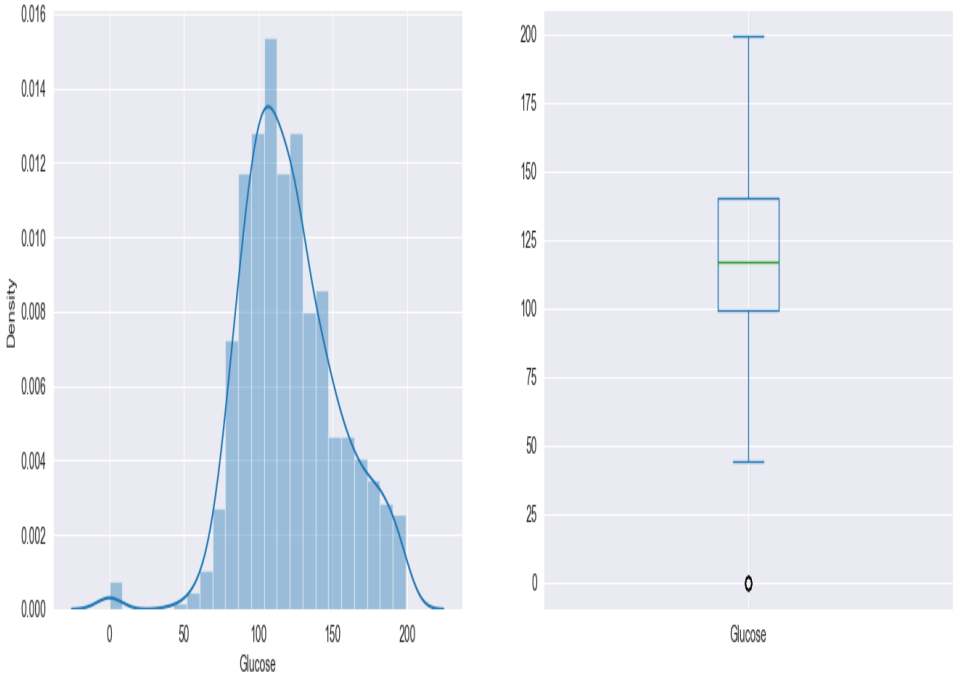
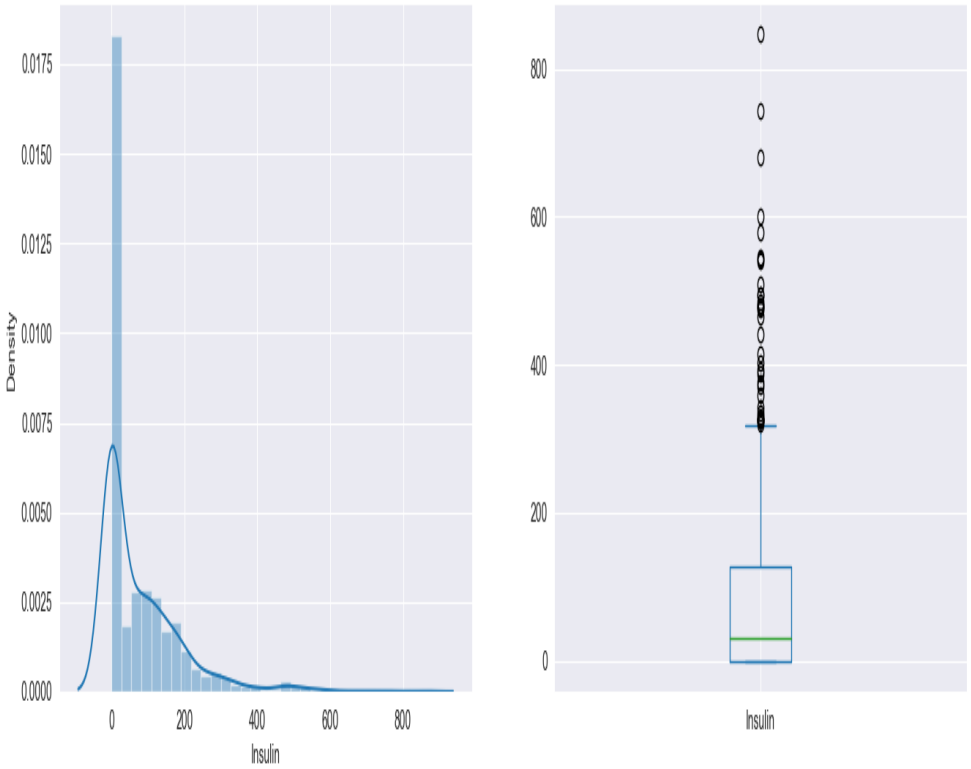


Figure 3.6. the Histogram for Glugose



. Figure 3.7. the Histogram for Insulin

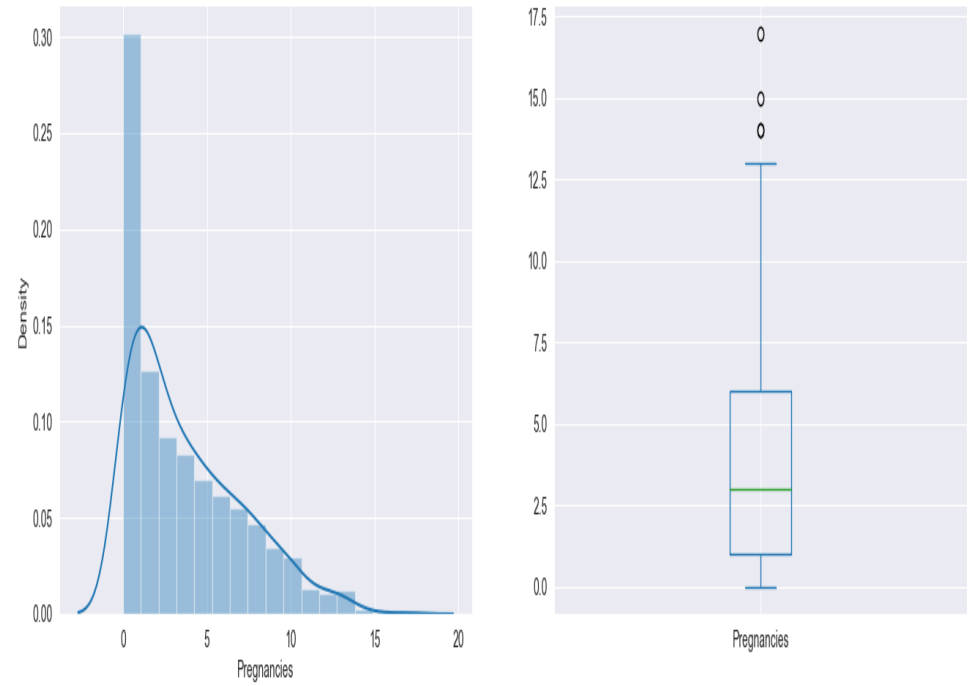


Figure 3.8. the Histogram for Pregnancy

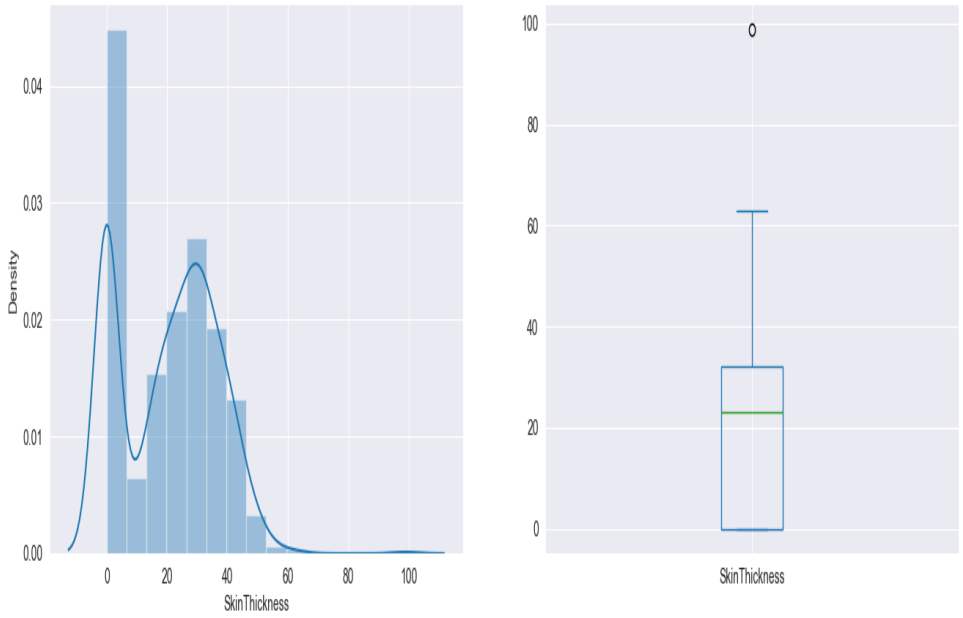


Figure 3.9. the Histogram for skin thickness

The columns that are most significant to our experiment are as follows: "Glucose" (which denotes sugar level), "BloodPressure," "BMI," and "Age". The Category Variables column indicates whether or not the person has been estimated to have diabetes (1 or 0). With features obtained from other columns, this column will perform as our output (target). Examine a few cases to understand how different health factors affect the outcome. When observing attentively, you will see that certain rows have zero values for columns where 0 makes absolutely no sense (e.g., SkinThickness, Insulin) These could be lost values that those sicker were unable to obtain. We can either use intelligent estimations to fill in all the missing values or eliminate them entirely to improve the dataset's quality. The former would necessitate a high level of data science expertise. For the sake of simplicity, let's remove all the rows with missing data. It's important to note that the Insulin and SkinThickness columns appear to have a large number of values

that have vanished. To avoid a huge number of rows being lost while trying to clean our data, we must remove the whole column. We want to use as many different records as we can in order to enhance the accuracy of our trained model.

3.3.2.1. Removing Columns that have missing value

From the left sidebar, go to Data Transformation ➤ Manipulation and drag-and-drop Select Columns in Dataset onto the canvas, just below the dataset. Connect the dataset's output port with the newly added module's input port by dragging the former's output port toward the Select Columns module's input port. This will draw an arrow pointing from dataset to the module.

Clicking the Select Columns module once brings up its properties in the right sidebar. From there, click the Launch Column Selector button and exclude the 2-Hour Serum Insulin column,

3.3.2.2. Renaming Columns

The default names of columns in our dataset are too long and descriptive. Let's simplify the column names. Search for and add the Edit Metadata module in canvas, just below Select Columns. Connect it to the module above. In module's properties pane, click Launch Column Selector and select all columns except 2-Hour Serum Insulin. Coming back to the properties pane, add the following new names in the New Column Names field, leaving all other fields unchanged:

pregnancy_count,glucose_level,diastolic_blood_pressure, bmi, diabetes_pedigree_function, age, diabetic Click the Run button to run your experiment. After it's finished running, all modules that ran successfully will have a green check mark. At this point, right-click the Edit Metadata module

and visualize the data. You should see the same dataset with renamed columns.

3.3.2.3. Removing Rows with Missing Values

This is usually done using the Clean Missing Data module, which can remove rows where all or certain columns have no specified values. Since our data has no explicit missing values, this module will not work for us. We need something that can exclude rows where certain columns are set to 0 (e.g., glucose_level, bmi, etc.). Search for and add the Apply SQL Transformation module in canvas, just below Edit Metadata. Connect its first input port to the module above. Enter the following query in the SQL Query Script field in the module's properties pane:

select * from t1 where glucose_level != 0 and diastolic_blood_pressure != 0 and bmi != 0 and age !=0; .

Run your experiment. Once it's finished, visualize your data by right-clicking the Apply SQL Transformation module. You notice the number of row become less than before.

3.3.3. Splitting Data

dataset should be divided into two parts—training data and scoring data. While training data is used to actually train the model, scoring data is reserved for later testing the trained model to calculate its loss. Search for and add the Split Data module in canvas, just below the second Select Columns module. Connect it to the module above. In its properties page, set 0.75 as the value for Fraction of Rows in the First Output Dataset. Leave all other fields as such. Run the experiment to ensure there are no errors. Once it's completed, you can visualize the Split Data module's left output port to check training data and right output port to check scoring data. You should find both in the ratio 75-25 percent. The dataset becomes 724 after pre-processing, this will allocate 543 rows for training and 181 for testing model.

3.3.4. Applying an ML Algorithm

Now you must pick an algorithm that must be appropriate with required output to train the model. Search for and add the module called tow-class Decision Forest in canvas, just to the left of Split Data. This algorithm is known for its accuracy and fast training times. Algorithm modules do not have an input port, so you will not connect it to any other module at the moment.

This project deals with the classification algorithms such as Decision Forest, support vector Machine, Average Perceptron, Boosted Decision Tree, Logistic Regression and Neural Network algorithms and the following represent the pseudo code for the six algorithms have been used:

1-Logistic-Regression(LR) algorithm

The algorithm of LR as follows :

step1: Function grad (predictor_attributes, target_attribute, weights)

{

Calculate gradient_descent;

Return weights+ learning rate * gradient_descent;

Step2: Normalize the dataset;

Step3: Repeat

{

Weights=grad(params);

Update weights;

} until convergence

Step4: z =dot product of predictor variables and updated weights;

Step5: prediction_limit=sigmoid function (z);

Step6: Predict the target class

2- Random Forest (RF) algorithm

The algorithm of RF as follows:

Algorithm Random Forest (Decision Tree Ensemble)

Prerequisite: Specify the training set

S := (x1.y1).... (xn. Yn) the set of all features F, and the number of trees to be included in the forest B

1: function RANDOM FOREST (S,F)

2: H ← 0

3: for i ϵ 1,..., B do

4: $S^{(i)}$ ← 1 Bootstrap sample drawn from S

5: h_i RandomizedTreelearn($S^{(i)}$, F)

6: H ← H ∪ $\{h_i\}$

7: end for

8: return H

9: end function

10: function RANDOMIZEDTREELEARN (S,F)

11: At each node:

12: f ← Draw a small subset of F

13: Split the best feature in f

14: return The Learned Tree (Model)

15: end function

3- Support Vector Machine (SVM) algorithm

The algorithm of SVM represents as follows:

AttributeSupportVector(ASV)

={Closest Attribute Pair from Opposite Classes)

1: while margin constraint violating points exist do

2: Find the violator

3: ASV = ASV U Violator

4: if any $\propto_p < 0$ because of addition of c to S then

5: ASV =$\frac{ASV}{P}$

6: Repeat all the violating points are pruned

7: end if

8: end while

4-Averaged Perceptron AP algorithm

The algorithm of AP represents as follows:

"Algorithm: Perceptron Learning Algorithm"

P ← inputs with label 1;

N ← inputs with label 0;

Initialize w randomly;

while ! convergence do

Pick random x ϵ P ⋃ N;

if x ϵ P and w.x < 0 then

| w= w+x;

end

if x ϵ N and w.x ≥ 0 then

| w = w-x;

end

end

//the algorithm converges when all the inputs are classified correctly

5-Artificial Neural Network (ANN) algorithm

The algorithm of NN represents as follows:

1: procedure ANN (Input, Neurons, Repeat)

Create input database

2: Input ← Database with all possible variable combinations

Train ANNS

3: for Input=1 to End of input do

Change inputs for every run

4: for Neurons = 1 to 20 do

Increase neurons for every run

5: for Repeat=1 to 20 do

Repeat run 20 times

6: Train ANN

7: ANN-Storage ← save highest test R^2

8: end for

9: end for

10: ANN-Storage← Save best predicting ANN depending on inputs

11: end for

12: return ANN-Storage ► Library with best predicting ANN for every variable combination

13: end procedure

6-Boosted Decision Tree(BDT) algorithm:

The algorithm of BDT represents as follows:

INPUT: S, where S = set of classified instances

OUTPUT: Decision Tree

Require: $S \neq ø$, num attributes > 0

1: procedure BUILDTREE

2: repeat

3: maxGain ← 0

4: split A ← null

5: e ← Entropy (Attributes)

6: for all Attributes a in S do

7: gain ← InformationGain(a, e)

8: if gain > maxGain then

9: maxGain ← gain

10: splitA ← a

11: end if

12: end for

13: Partition (S, splitA)

14: until all partitions processed

15: end procedure

3.3.5. Training the Model

Search for and add the module called Train Model in canvas, just below the algorithm and Split Data modules. Connect its left input port to the algorithm module and its right port to Split Data's left output port. From its properties pane, select the Diabetic column. Doing so will set it as our model's target value. Run the experiment. If all goes well, all modules will have a green checkmark. Like data, you can visualize a trained model as well,

3.3.6. Scoring and Evaluating the Trained Model

Search for and add the module Score Model in canvas, just below the Train Model module. Connect its left input port to the Train Model module and its right port to Split Data's right output port. Run the experiment. Table (3.2) shows how visualizing scored data looks.

Table3.2. illustrate Visualizing scored data in Azure ML Studio

pregnancy_count	glucose_level	diastolic_blood_pressure	bmi	diabetes_pedigree_function	age	diabetic	Scored Labels	Scored Probabil-ities
0	123	72	36.3	0.258	52	1	0	0.5
2	112	66	25	0.307	24	0	0	0.125
0	117	80	45.2	0.089	24	0	0	0.125
7	114	64	27.4	0.732	34	1	0	0.5
3	130	64	23.1	0.314	22	0	0	0.125
4	114	65	21.9	0.432	37	0	0	0
2	108	62	25.3	0.881	22	0	0	0.125
0	95	85	37.4	0.247	24	1	0	0
7	97	76	40.9	0.871	32	1	0	0.25
13	76	60	32.8	0.18	41	0	0	0.125
7	168	88	38.2	0.787	40	1	1	1
2	84	50	30.4	0.968	21	0	0	0
4	156	75	48.3	0.238	32	1	1	0.75

As you can confirm from the table3.2, scoring is performed with the 25% output of Split Data module. And the no. of testing data was 181 after pre-processing operation. Values in the Diabetic column represent the actual values in the dataset, and values in the Scored Labels column are predicted values. Also note a correlation between the Score Labels and Scored Predictions columns. Where probability is sufficiently high, the actual and predicted values are the same. If actual and predicted values match for most rows, you can say our trained model is accurate. If not, try tweaking the

algorithm (or replacing it altogether) and running the experiment again. Next, search for and add the Score Model module in canvas, just below the Score Model module. Later drag and drop from the left list to the canvas Connect its left input port with the above module. Run the experiment and visualize Evaluate Model's output. You will see metrics such as accuracy, precision, f-score, etc., that can be used to determine correctness of this model. For all aforementioned metrics, values closer to 1 indicate a good model.

3.3.7. Programming Environment

The environment used in this project is Microsoft Azure Machine Learning Studio, it contributes as good tool for Machine Learning algorithms. the environment has well documented effective Machine Learning packages and are most widely used for data mining and predictive analysis strategies world-wide. Azure ML Studio was chosen because it has so many advantages. It does not need code and does not require that you be skilled in programming languages, which means that you will not suffer from programming errors or failure to implement the program. It quickly emerging Machine Learning algorithms into their system. Azure studio is accurate and does not require a large time from the beginning of the work to its end, it is quick to implement. Azure ML studio is unique in this advantage.

3.3.8. Deploying a Trained Model as a Web Service

At this time, we have a trained and tested model, and it is ready to be put to actual use. We'll provide end-user software apps the ability to programmatically forecast values by delivering them as a web service. we use the diabetes prediction model as a web service solution backend or with the Azure Stream service for real-time diabetes prediction.

Select the " Set Up Web Service ➤ Predictive Web Service option ". The present experiment would be transformed into a forecasting experiment as a result of this change.

After publishing the system as a web service, users can access it and enter their medical data to diagnose their health condition. As an example, for a person who uses this service to find out whether he has diabetes or not by entering the following information:

pregnancy_count = 6

glucose_level = 148

diastolic_blood_pressure = 72

bmi=35

diabetes_pedigree_function = 33.6,

age = 50,

Diabetic is the item we want to predict.

The result is

Score labels =1 (Diabeic)

Scored Probabilities = 0.571 (diabetic). (since it is more than 0.500 means that the patient is diabetic.

CHAPTER FOUR

Chapter Four

Implementation and Results

4.1. Introduction

In this chapter, will show the implementation and performance evaluation as graphs and discussion for the proposed system outlined in Chapter 3. The goal is twofold: first, to evaluate the performance of techniques with different performance metrics via a real diabetic dataset. Second, making a comparison among the results obtained with the used techniques.

4.2. The Experiment Environment

ML applications can be created in a variety of ways. Someone, for example, can write code to implement a decision forest themselves, but this may cause a waste of time and it may be full of errors, hence it may be passive. Azure Machine-Learning Studio (AML) is one of the best ways for creating an end-to-end ML solution. AML Studio is a graphical interface for designing and implementing machine learning processes that use drag-and-drop functionality. AML Studio includes a large number of tried-and-true executions of tens of machine learning techniques for a wide range of problems.

4.3. Experimental Results

We will present results for prediction and efficiency of each algorithm depending on ($true\ positive\ (TP)$), ($false\ negative\ (FN)$), ($false\ positive\ (FP)$),($true\ negative\ (TN)$), $accuracy$, $precision$, $recall$, and AUC. Throughout all the experiments, 75% of the data was used for training and 25% for testing. The training set was 543 and the testing set was 181 (used for testing the model) after pre-processing operation on the dataset, where dataset was contained 768 rows

before pre-processing. The six ML models were applied to the PIMA diabetes dataset in the following experiments, and one attribute was chosen as "Label" and utilized for instance categorization.

4.3.1. Logistic Regression (LR)

The following findings of execution LR were acquired in this experiment and after the learning process. From the results, we can notice the following:

❖ The ROC (Receiver Operating Characteristic) curve depicts the trade-off between the True Positive Rate (the proportion of positive instances that are properly diagnosed) and the False Positive Rate (positive cases incorrectly classified). This curve is to the left and above the bright 45-degree line as shown in Figure 4.1. This shows that the classifier is more successful than guessing.

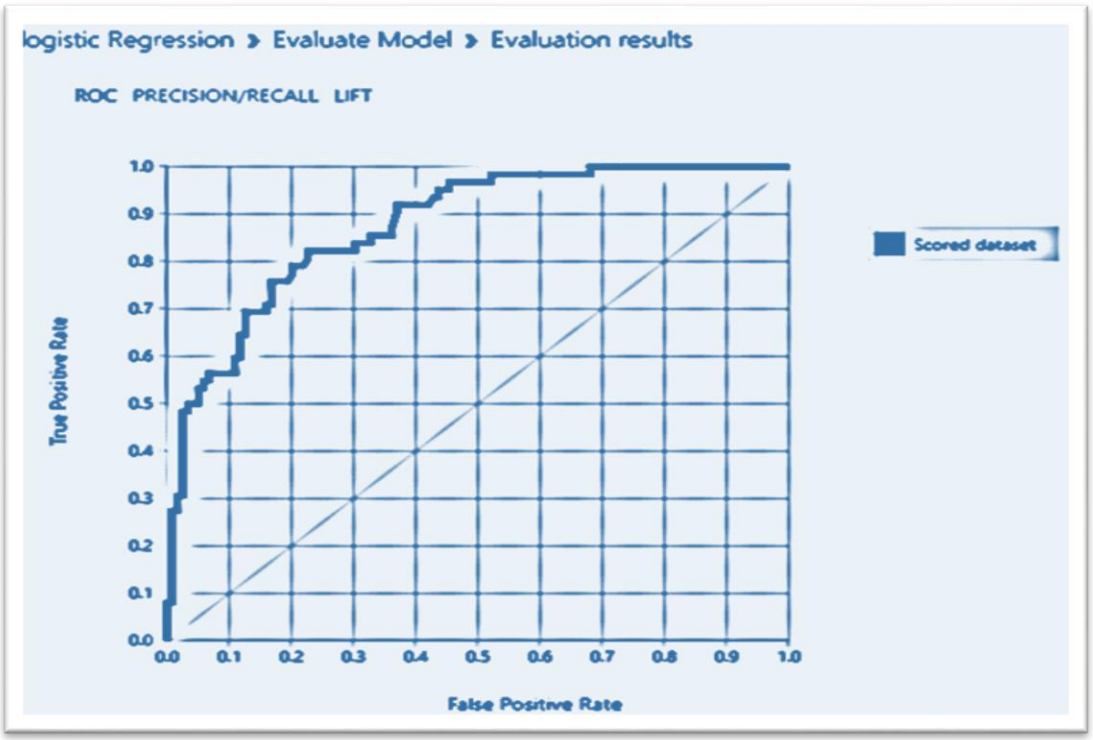


Figure 4.1. The ROC curve of Logistic Regression

❖ Table 4.1 shows the confusion matrix generated by this experiment.

Table 4.1. Confusion matrix Using LR algorithm

	Non-Diabetes (0)	Diabetes (1)
Non-Diabetes (0)	111	8
Diabetes (1)	27	35

The LR produced a reasonable confusion matrix, indicating that of the 25% of diabetic datasets used for testing, (35) were true positive, i.e., they were correctly predicted to be diabetic, and (27) were false negative, i.e., they were incorrectly predicted to be non-diabetic when they were actually diabetic. It also shows that 25% of non-diabetic datasets were used for testing. Eight of them were false positives, which means they were incorrectly projected to be diabetic, when they were actually non-diabetic. While (111) were true negatives, which means they were properly predicted to be non-diabetic.

- The **AUC** is greater than 0.87, suggesting that the classifier is practically good.
- The **Accuracy** is currently more than 80%.
- The **Recall** rate is presently close to 0.57.
- The **Precision** of 0.814 indicates that False Positives have been considerably decreased.
- The **F1** score is 0.667.

4.3.2. Decision Forest (DF):

The following findings of execution DF were acquired in this experiment and after the learning process. From the results, we can notice the following:

- The ROC curve is to the left and above the bright 45-degree line as shown in Figure 4.2. This shows that the classifier is more successful than guessing.

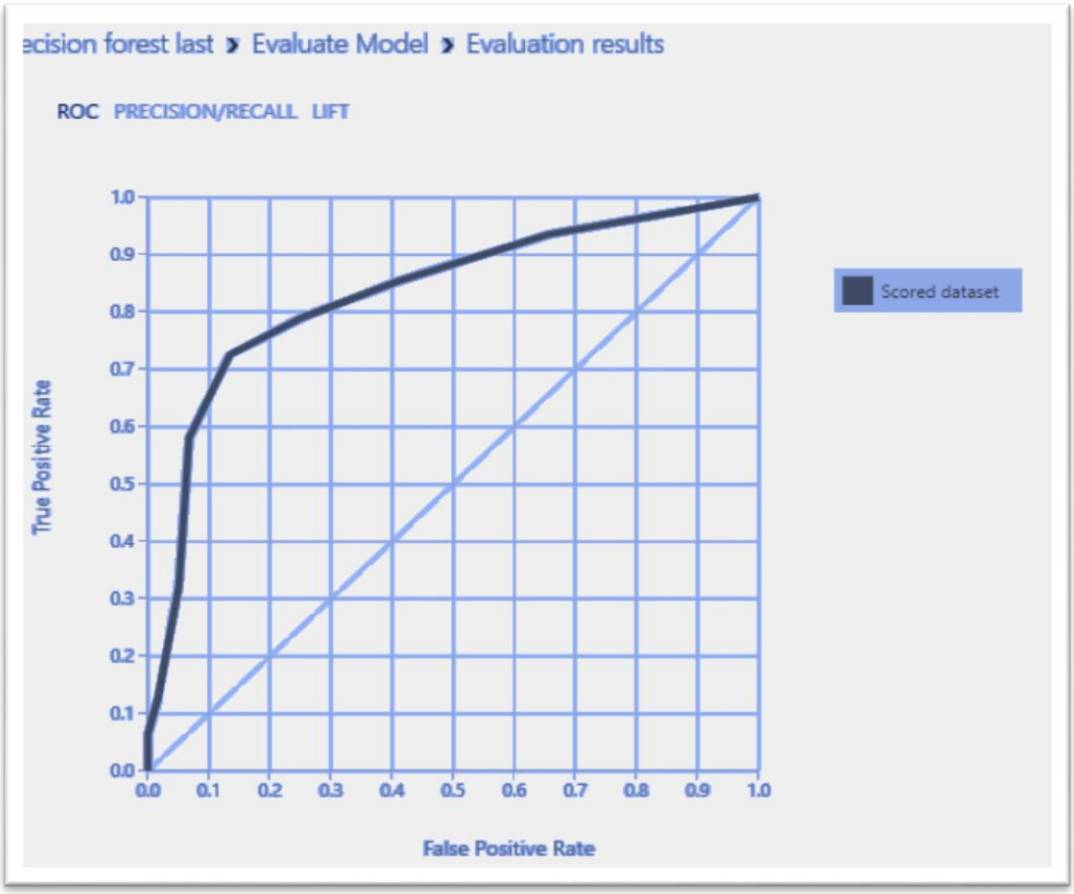


Figure 4.2. The ROC curve of Decision Forest

❖ Table 4.2 shows the confusion matrix generated by this experiment.

Table 4.2. Confusion matrix Using DF algorithm

	Non-Diabetes (0)	Diabetes (1)
Non-Diabetes (0)	111	8
Diabetes (1)	26	36

The DF produced a reasonable confusion matrix, indicating that (36) were correctly predicted to be diabetic, and (26) were incorrectly predicted to be non-diabetic when they were actually diabetic. It also shows that (8) of them were incorrectly projected to be diabetic, when they were actually non-diabetic. While (111) were properly predicted to be non-diabetic.

- ❖ The **AUC** is greater than 0.83, suggesting that the classifier is practically good.
- ❖ The **Accuracy** is currently more than 81%.
- ❖ The **Recall** rate is presently close to 0.58.
- ❖ The **Precision** of 0.818 indicates that False Positives have been considerably decreased.
- ❖ The **F1** score is 0.679.

4.3.3. Averaged Perceptron (AP)

The following findings of execution AP were acquired in this experiment and after the learning process. From the results, we can notice the following:

- ❖ The ROC curve is to the left and above the bright 45-degree line as shown in Figure 4.3. This shows that the classifier is more successful than guessing.

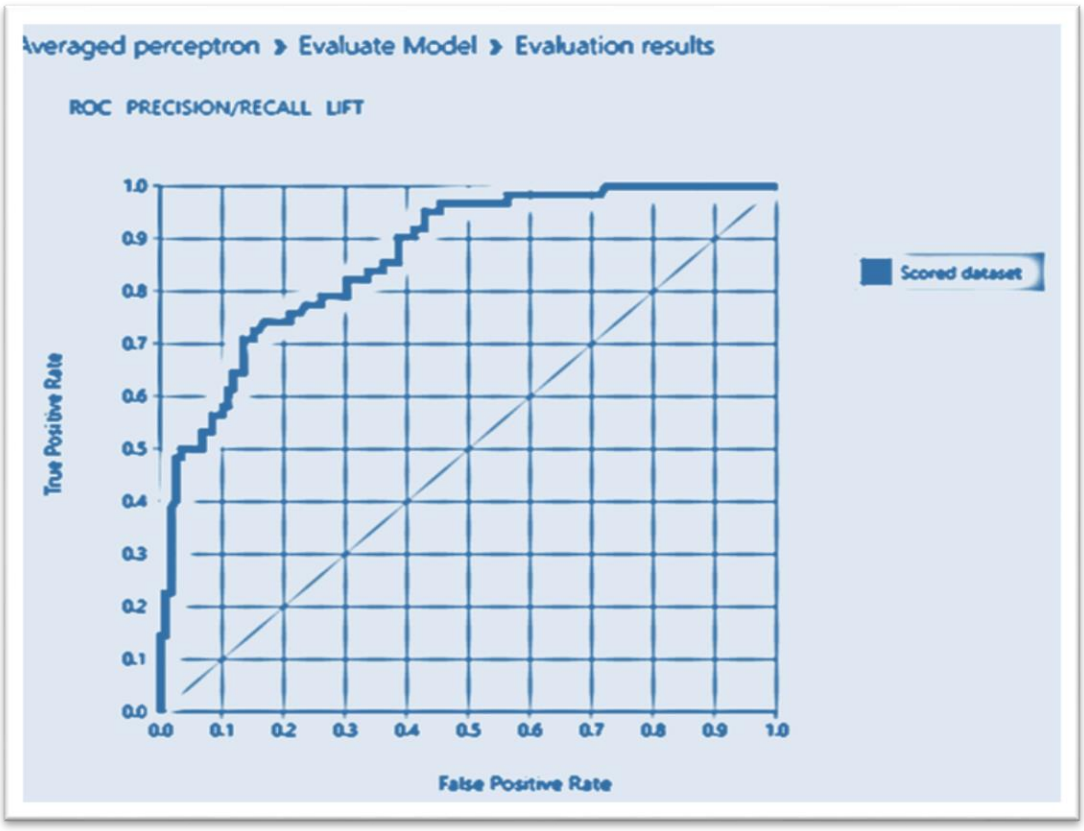


Figure 4.3. The ROC curve of Averaged Perceptron

- Table 4.3 shows the confusion matrix generated by this experiment.

Table 4.3. Confusion matrix Using AP algorithm

	Non-Diabetes (0)	Diabetes (1)
Non-Diabetes (0)	106	13
Diabetes (1)	26	36

The AP produced a reasonable confusion matrix, indicating that (36) were correctly predicted to be diabetic, and (26) were incorrectly predicted to be non-diabetic when they were actually diabetic. It also shows that (13) of them were incorrectly projected to be diabetic when they were actually non-diabetic. While (106) were properly predicted to be non-diabetic.

- The **AUC** is greater than 0.86, suggesting that the classifier is practically good.
- The **Accuracy** is currently more than 78%.
- The **Recall** rate is presently close to 0.58.
- The **Precision** of 0.735 indicates that False Positives have been considerably decreased.
- The **F1** score is 0.649.

4.3.4. Neural Network (NN)

The following findings of execution NN were acquired in this experiment and after the learning process. From the results, we can notice the following:

- The ROC curve is to the left and above the bright 45-degree line as shown in Figure 4.4. This shows that the classifier is more successful than guessing.

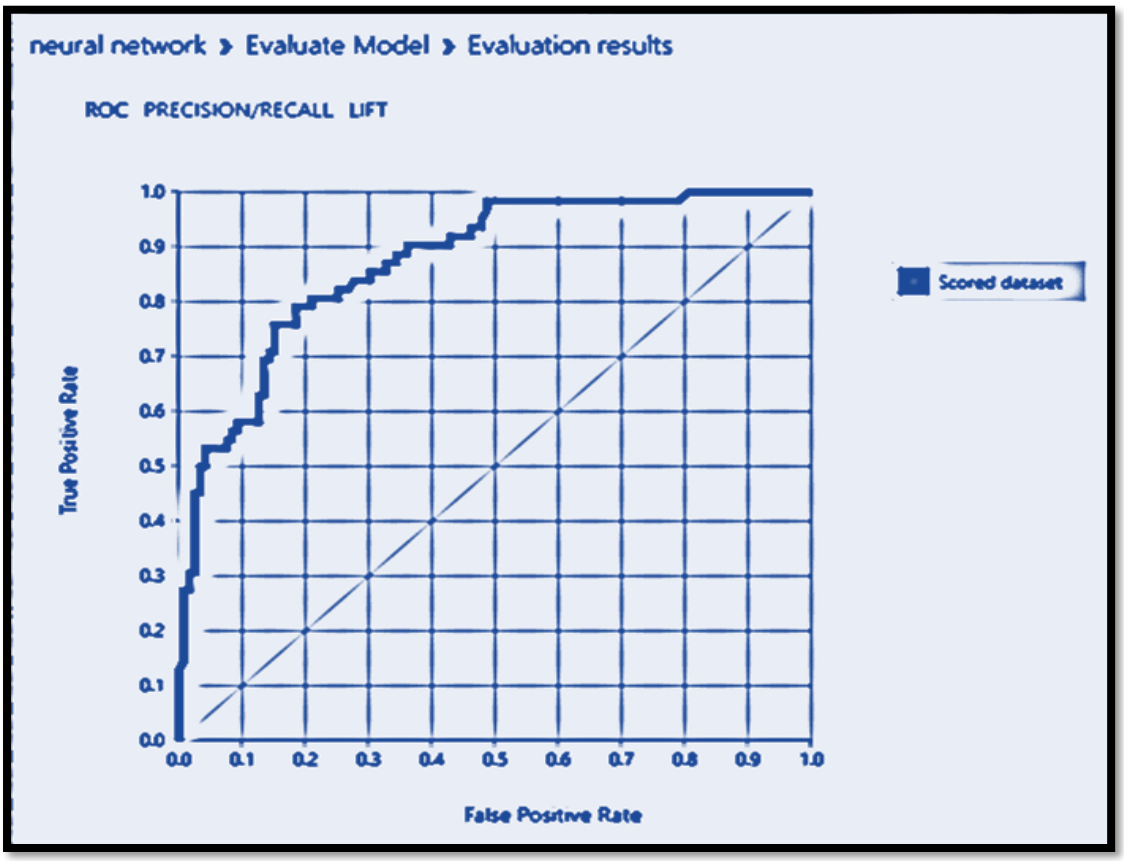


Figure 4.4. The ROC curve of Neural Network

❖ Table 4.4 shows the confusion matrix generated by this experiment.

Table 4.4. Confusion matrix Using NN algorithm

	Non-Diabetes (0)	Diabetes (1)
Non-Diabetes (0)	106	13
Diabetes (1)	26	36

The NN produced a reasonable confusion matrix, indicating that (36) were correctly predicted to be diabetic, and (26) were incorrectly predicted to be non-diabetic when they were actually diabetic. It also shows that (13) of them were incorrectly projected to be diabetic when they were actually non-diabetic. While (106) were properly predicted to be non-diabetic.

- The **AUC** is greater than 0.87, suggesting that the classifier is practically good.
- The **Accuracy** is currently more than 78%.
- The **Recall** rate is presently close to 0.58.
- The **Precision** of 0.735 indicates that False Positives have been considerably decreased.
- The **F1** score is 0.649.

4.3.5. Support Vector Machine (SVM)

The following findings of execution SVM were acquired in this experiment and after the learning process. From the results, we can notice the following:

- The ROC curve is to the left and above the bright 45-degree line as shown in Figure 4.5. This shows that the classifier is more successful than guessing.

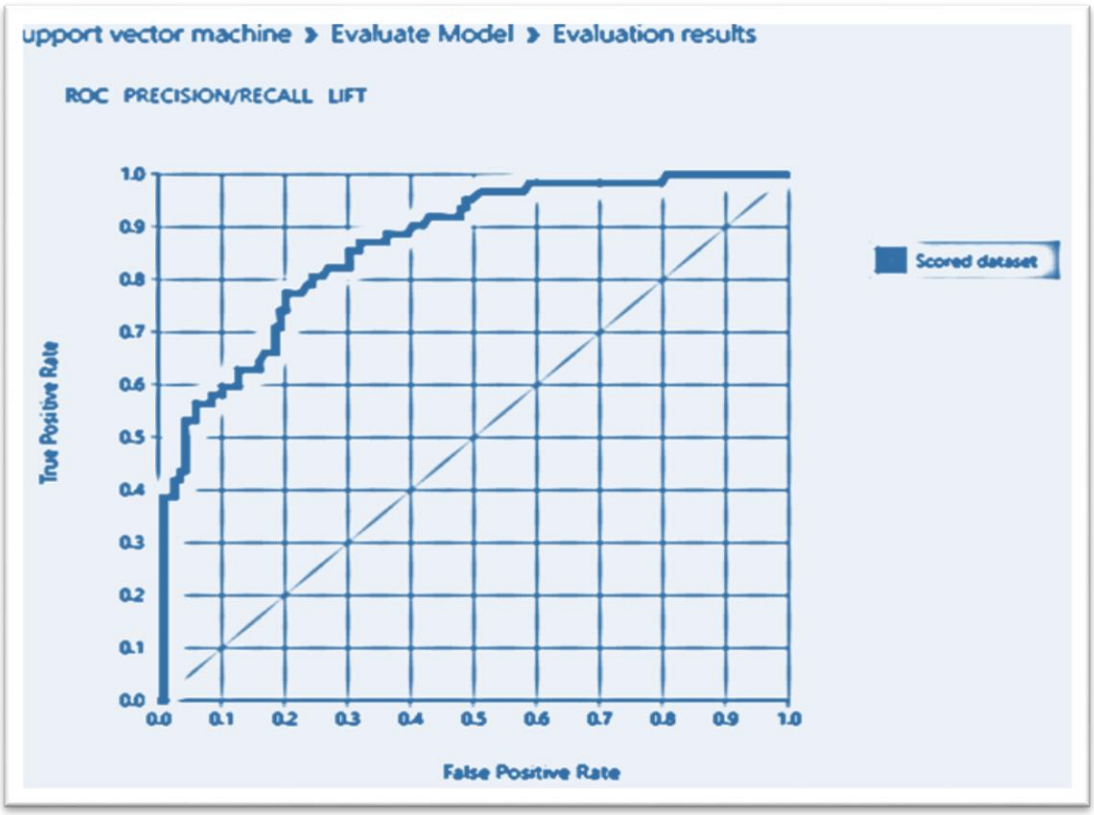


Figure 4.5. The ROC curve of Support Vector Machine

- Table 4.5 shows the confusion matrix generated by this experiment.

Table 4.5. Confusion matrix Using NN algorithm

	Non-Diabetes (0)	Diabetes (1)
Non-Diabetes (0)	107	12
Diabetes (1)	26	36

The SVM produced a reasonable confusion matrix, indicating that (36) were correctly predicted to be diabetic, and (26) were incorrectly predicted to be non-diabetic when they were actually diabetic. It also shows that (12) of them were incorrectly projected to be diabetic, when they were actually non-diabetic. While (107) were properly predicted to be non-diabetic.

- The **AUC** is greater than 0.86, suggesting that the classifier is practically good.
- The **Accuracy** is currently more than 79%.
- The **Recall** rate is presently close to 0.58.
- The **Precision** of 0.735 indicates that False Positives have been considerably decreased.
- The **F1** score is 0.655.

4.3.6. Boosted Decision Tree (BDT)

The following findings of execution BDT were acquired in this experiment and after the learning process. From the results, we can notice the following:

- The ROC curve is to the left and above the bright 45-degree line as shown in Figure 4.6. This shows that the classifier is more successful than guessing.

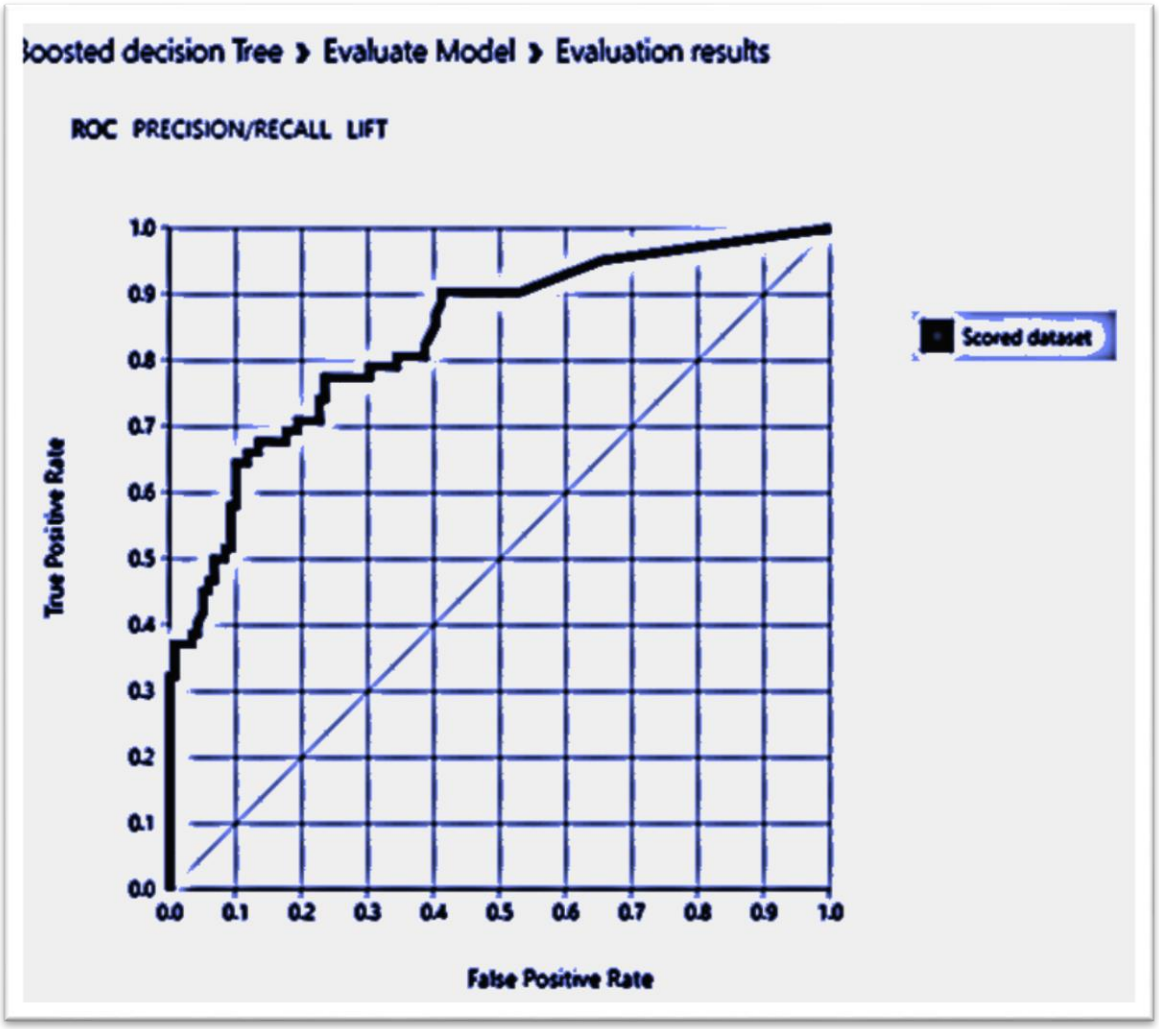


Figure 4.6. The ROC curve of Boosted Decision Tree

❖ Table 4.6 shows the confusion matrix generated by this experiment.

Table 4.6. Confusion matrix Using BDT algorithm

	Non-Diabetes (0)	Diabetes (1)
Non-Diabetes (0)	107	12
Diabetes (1)	25	37

The BDT produced a reasonable confusion matrix, indicating that (37) were correctly predicted to be diabetic, and (25) were incorrectly predicted to be non-diabetic when they were actually diabetic. It also shows that (12) of them were incorrectly projected

to be diabetic when they were actually non-diabetic. While (107) were properly predicted to be non-diabetic.

- The **AUC** is greater than 0.83, suggesting that the classifier is practically good.
- The **Accuracy** is currently more than 79%.
- The **Recall** rate is presently close to 0.6.
- The **Precision** of 0.755 indicates that False Positives have been considerably decreased.
- The **F1** score is 0.667.

4.4. Comparing the Performance of ML Algorithms

In this section, a comparison between the performances of the six algorithms that were used in the prediction process for the diagnosis of diabetes is made. The six algorithms are, namely, DF, BDT, LR, SVM, NN, and AP. The highest accuracy recorded was 81.2 and 80.7 for each of the DF and LR algorithms, respectively. The algorithms that scored the highest AUC are LR, equal to 87.5%, NN, equal to 87.3%, and AP, equal to 87.6%. And the algorithm that scored the highest precision was DF, which was 81.8%, as shown in the statistical chart (4.7) and table (4.7).

The table below clear most of the performance metrics

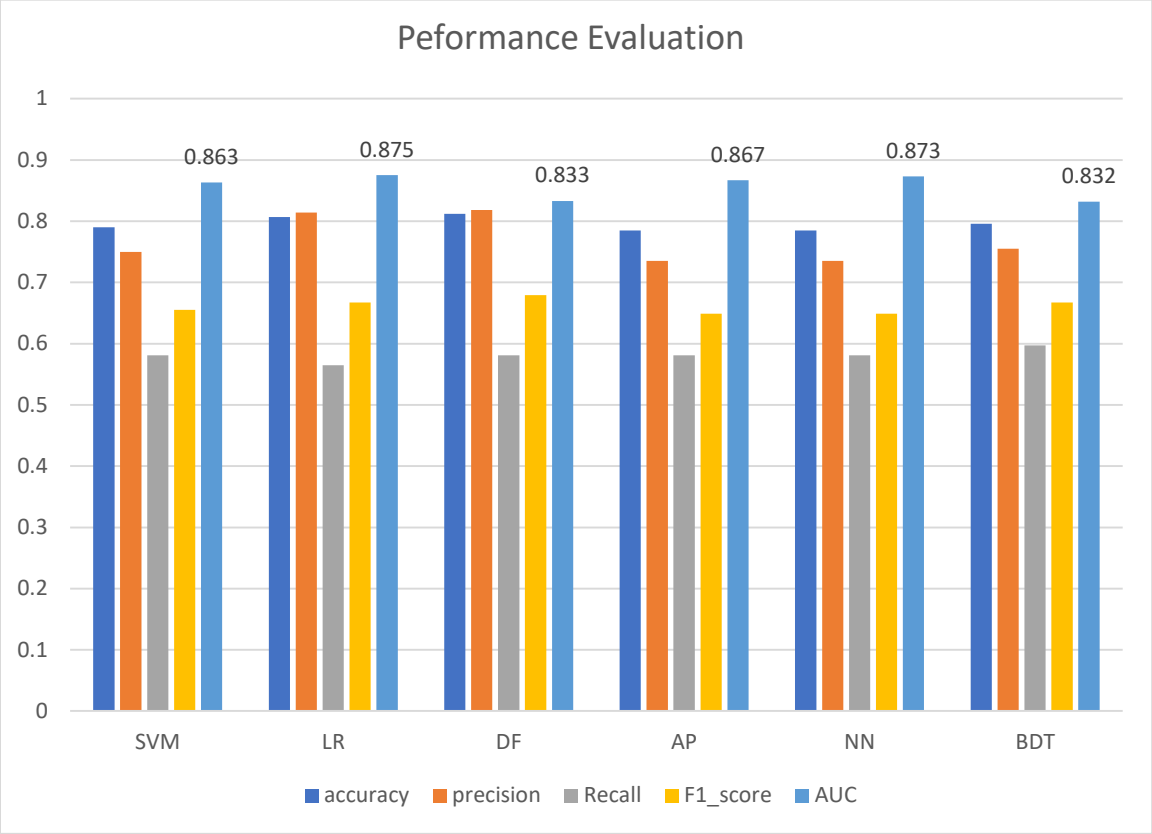


Figure (4.7) statistical chart that shows the performance evaluation of algorithms

Table (4.7) shows the evaluation of performance to the algorithms

Algorithm	Accuracy	True-Positive	False-Negative	Precision	False-Positive	True-Negative	Recall	F1_score	AUC
SVM	79 %	36	12	0.75	12	107	0.581	0.655	0.863
LR	80.7%	35	27	0.814	8	111	0.565	0.667	0.875
DF	81.2 %	36	26	0.818	8	111	0.581	0.679	0.833
AP	78.5 %	36	26	0.735	13	106	0.581	0.649	0.867
ANN	78.5 %	36	26	0.735	13	106	0.581	0.649	0.873
BDT	79.6 %	37	25	0.755	12	107	0.597	0.667	0.832

4.5. The Result of Prediction using Web Service

This section displays the results of applying the proposed system after converting it into a web service and based on the Decision Forest ML algorithm, it is chosen since it get most accuracy . The result as follows:

Case (1):

The input values of features are:

pregnancy_count = 1

glucose_level = 122

diastolic_blood_pressure = 90

bmi=49.7

diabetes_pedigree_function =0.325

age = 31

outcome =

The result is Scored labels = 1 (diabetic).

Scored probabilities = 0.875 ► means that the patient is diabetic since the value ≥ 0.500

When the scored labels get to (1), it means that a diabetes is found as there are only two cases, infected 1 and uninfected 0, and the scored probability equals 0.875, which is greater than 0.500, then the diabetes is found.

Case (2):

The input values of features are:

pregnancy_count = 2

glucose_level = 92

diastolic_blood_pressure = 62

bmi=31.6

diabetes_pedigree_function =0.13

age = 24

outcome =

The result is Scored labels = 0 (non-diabetic).

Scored probabilities = 0.125 absence diabetes

When the scored labels get to (0), it means that diabetes is not found as there are only two cases, infected 1 and uninfected 0, and the scored probability equals 0.125. Surely, it's a number less than 0.500 as we mentioned earlier, it means diabetes-free.

Chapter Five

CHAPTER FIVE

CONCLUSIONS AND FUTURE WORKS SUGGESTIONS

5.1. Introduction

The conclusions of this project would be derived from the outcomes of the execution of the proposed system as documented in the previous chapters. This chapter is split into two sections: the conclusions and the future work suggestions.

5.2. Conclusions

ML aids in the creation of estimation methods for diabetes and associated consequences:

1-In the proposed system was used to make a system to predict the diagnosis of diabetes. Six machine learning algorithms were used. The data has been divided for training the model and testing by 75-25% because we found it to be the best percentage that gives the best results. The best accuracy was achieved by the decision forest algorithm, where the accuracy reached more than 81%, which supports the use of machine learning methods to predict the occurrence of diabetes.

2-the research will help to bring in an element of personalized care in the management of diabetes. Patients are now being empowered to manage their own health and physicians can provide a timely and targeted intervention through technical platforms.

3-These advances save time and cost because data can be collected remotely and virtual management is replacing the routine visits to a clinic.

4-The proposed system predicts the diabetes illness with accuracy and the diabetic do not need diabetes checked.

5-It can predict the case of diabetes in its beginning, which contributes to reducing the incidence of complications.

5.3. Suggestions for Future Works

1- The project aim to research more features that can be used as potential predictors. In this way, this work can be expanded to aid in fully automating diabetes prediction

2-the work cannot predict the type of diabetes, so in future, so the project aim to predicting the type of diabetes and exploring the proportion of each indicator, which may improve the accuracy of predicting diabetes

3- in future proposal is to link the system with the IoT network to fetch information quickly and in real-time in order to predict the patient's condition more efficiently, accurately, and quickly.

4- Because of the lack of time, the system not complete what they wanted to increase the accuracy of the algorithms, so the suggestion is to use two datasets and joining them together to increase the number of the training set to get a more accurate result.

Reference

References

[1] Zou, Q., Qu, K., Luo, Y., Yin, D., Ju, Y., & Tang, H. (2018). Predicting diabetes mellitus with machine learning techniques. *Frontiers in genetics*, *9*, 515.

[2] Ismail, L., & Materwala, H. (2021). IDMPF: intelligent diabetes mellitus prediction framework using machine learning. Applied Computing and Informatics.

[3] Rao, S. K. (2018). Health care System: Stream Machine Learning Classifier for features Prediction in Diabetes Therapy. International Journal of Applied Engineering Research, 13(1), 59-65.

[4] Sarwar, M. A., Kamal, N., Hamid, W., & Shah, M. A. (2018, September). Prediction of diabetes using machine learning algorithms in healthcare. In 2018 24th International Conference on Automation and Computing (ICAC) (pp. 1-6). IEEE.

[5] Forsman, R., & Jönsson, J. (2019). Artificial intelligence and Machine learning: a diabetic readmission study.

[6] Khedkar, V. N., & Patel, S. (2021). Diabetes Prediction using Machine learning: A Bibliometric Analysis. Diabetes.

[7] Joshi, T. N., & Chawan, P. P. M. (2018). Diabetes prediction using machine learning techniques. Ijera, 8(1), 9-13.

[8] Dey, S.K., Hossain, A. and Rahman, M.M., 2018, December. Implementation of a web application to predict diabetes disease: an approach using machine learning algorithm. In *2018 21st international conference of computer and information technology (ICCIT)* (pp. 1-5). IEEE.

[9] Islam, M. A., & Jahan, N. (2017). Prediction of onset diabetes using machine learning techniques. International Journal of Computer Applications, 180(5), 7-11.

[10] Reid, C. (2019). Diabetes Diagnosis and Readmission Risks Predictive Modelling: USA (Doctoral dissertation, Dublin, National College of Ireland).

[11] Wiley, M. T. (2011). Machine learning for diabetes decision support (Doctoral dissertation, Ohio University).

[12] Pathak, N. and Bhandari, A., 2018. IoT, AI, and Blockchain for .NET. Building a Next-Generation Application from the Ground Up. Apress

[13] Joshi, R., & Alehegn, M. (2017). Analysis and prediction of diabetes diseases using machine learning algorithm: Ensemble approach. International Research Journal of Engineering and Technology, 4(10), 426-435

[14] Zhang, X. (2019). Machine learning algorithm cheat sheet for Azure Machine Learning Studio. Microsoft Azure, https://docs. microsoft. com/en-us/azure/machine-learning/studio/algorithm-cheat-sheet.

[15] Bhavsar, H., & Ganatra, A. (2012). A comparative study of training algorithms for supervised machine learning. International Journal of Soft Computing and Engineering (IJSCE), 2(4), 2231-2307.

[16] Osisanwo, F. Y., Akinsola, J. E. T., Awodele, O., Hinmikaiye, J. O., Olakanmi, O., & Akinjobi, J. (2017). Supervised machine learning algorithms: classification and comparison. International Journal of Computer Trends and Technology (IJCTT), 48(3), 128-138.

[17] Sridhar, K. K. (2020). Using Machine Learning to Predict Readmissions of Diabetes Patients.

[18] Kotsiantis, S. B., Zaharakis, I., & Pintelas, P. (2007). Supervised machine learning: A review of classification techniques. Emerging artificial intelligence applications in computer engineering, 160(1), 3-24.

[19] Rokach, L. (2016). Decision forest: Twenty years of research. Information Fusion, 27, 111.

[20] Uddin, S., Khan, A., Hossain, M. E., & Moni, M. A. (2019). Comparing different supervised machine learning algorithms for disease prediction. BMC medical informatics and decision making, 19(1), 1-16.

[21] Mahmud, S. H., Hossin, M. A., Ahmed, M. R., Noori, S. R. H., & Sarkar, M. N. I. (2018, August). Machine learning based unified framework for diabetes prediction. In Proceedings of the 2018 International Conference on Big Data Engineering and Technology (pp. 46-50).

[22] Jagirdar, N. M. (2018). Online Machine Learning Algorithms Review and Comparison in Healthcare.

[23] ghous, h. (2020). *analysis of different machine learning tools for diabetes prediction* (doctoral dissertation, university of Miskolc).

[24] Sowah, R. A., Bampoe-Addo, A. A., Armoo, S. K., Saalia, F. K., Gatsi, F., & Sarkodie-Mensah, B. (2020). Design and development of diabetes management system using machine learning. *International journal of telemedicine and applications*, *2020*.

[25] Barnes, J. (2015). Azure machine learning. *Microsoft Azure Essentials. 1st ed, Microsoft*.
[26] Mund, S. (2015). *Microsoft azure machine learning*. Packt Publishing Ltd.
[27] Tomar, D., & Agarwal, S. (2013). A survey on Data Mining approaches for Healthcare. International Journal of Bio-Science and Bio-Technology, 5(5), 241-266.

الخلاصة

مرض السكري من الأمراض الخطيرة والصامتة وهو من ضمن العشرة اسباب التي قد تسبب الموت المفاجئ. يمكن أن تحدث الاصابة به في أي وقت وقد يسبب إصابة جسيمة بأعضاء الجسم أو يتلفها بالكامل. لذلك يجب التحقيق من هذا المرض في بداية ظهوره وقبل أن يصعب علاجه.
مع التقدم السريع في التعلم الآلي (ML) ، عززت هذه الأساليب كفاءة عمليات اتخاذ القرار في مجموعة واسعة من التطبيقات ، بما في ذلك التشخيص الطبي. في هذا المشروع، اخترنا مجال تطبيق طبي واستخدمنا ستة خوارزميات للتعلم الخاضع للإشراف ومن ثم تم اختيار الموديل الأكثر كفاءة لبناء نموذج تنبؤ عالي الدقة لمرض السكري لدى البشر في مرحلة مبكرة، قبل أن يتطور إلى درجة المرض أو الوفاة وبعد ذلك نشر هذا النموذج في صفحة الويب. يمكن للنموذج المقترح استخلاص المعرفة المخفية من البيانات المتعلقة بمرض السكري التي تم جمعها من مستودع التعلم الآلي Kaggle. ستفيد هذه الدراسة أيضًا القطاع الصحي من خلال تزويد المستخدمين بأداة عبر الإنترنت (أي صفحة ويب) تسمح لهم بإدخال البيانات وتلقي النتائج التي تتنبأ بما إذا كان الشخص مصابًا بالسكري أم لا. نتيجة لذلك، فإن المعرفة المسبقة والمراقبة المستمرة لحالتهم الصحية لمرضى السكري ستقلل من مخاطر المضاعفات والمراضة والوفاة الناجمة عن هذا المرض. تتمثل المساهمة الرئيسية للنظام المقترح في تحسين جودة الرعاية الصحية، وتقليل الاستشفاء، وخفض النفقات المرتفعة للرعاية الصحية والأدوية.

في هذا المشروع استخدمنا Microsoft Azure Machine Learning Studio لنمذجة خوارزميات ML. هذا الاستوديو عبارة عن بيئة تأليف بصرية قوية للغاية تعتمد على السحب والإفلات. ليست هناك حاجة لأي أنظمة أو تطبيقات معينة لأنها بيئة أساسية قائمة على المستعرض. بعد إجراء العديد من تجارب خوارزميات التصنيف يتم اختيار الخوارزمية التي تعطي افضل النتائج لاستخدامها بعد ذلك لبناء النظام المقترح، يتم قياس العديد من مؤشرات الأداء ، بما في ذلك (f1-score, Accuracy, Precision, Recall)للمقارنة بين أداء الخوارزميات. بناءً على مخرجات التصنيف، تم تحديد أن Decision Forest هي الخوارزمية الافضل وتنتج نتائج أفضل من مناهج Machine Learning الاخرى .

وزارة التعليم العالي
والبحث العلمي
جامعة بابل
كلية العلوم للبنات

نظام تنبؤي للرعاية الصحية لتشخيص مرض السكري بالاعتماد على تقنيات التعلم الآلي وخدمة ويب

مشروع
مقدم الى مجلس كلية العلوم للبنات في جامعة بابل استيفاء جزءا
لمتطلبات نيل درجة الدبلوم العالي في علوم الحاسوب

من قبل
رشـــا علي عبد الرحيم

الغرابي أشراف محمد كاظم علي أ.م.د.

1443 **2022**