

PROFESSIONAL TRAINING REPORT

entitled

Multiple Disease Prediction using Machine Learning Techniques

Submitted in partial fulfillment of the requirements for the award of
Bachelor of Engineering degree in Computer Science and Engineering with
Specialization in Cyber Security

by

SANDIPA SAHA

41614083



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SCHOOL OF COMPUTING**

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited with Grade "A++" by NAAC

JEPPIAAR NAGAR, RAJIV GANDHISALAI,
CHENNAI – 600119

APRIL 2024



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

CATEGORY-I UNIVERSITY BY UGC

Accredited "A++" Grade by NAAC | 12B Status by UGC | Approved by AICTE

www.sathyabama.ac.in

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Professional Training Report is the bonafide work of **Ms. Sandipa Saha (41614083)** who carried out the project entitled "**Multiple Disease Prediction Using Machine Learning Techniques**" under my supervision from January 2024 to May 2024.

Internal Guide

Dr. D. Anu Disney, M.E., Ph.D.

Head of the Department

Dr. S. VIGNESHWARI, M.E., Ph.D.

Submitted for Viva voce Examination held on 07-05-24

Internal Examiner

External Examiner

DECLARATION

I, **SANDIPA SAHA (41614083)**, hereby declare that the Professional Training Report-II entitled "**Multiple Disease Prediction using Machine Learning Techniques**" done by me under the guidance of **Dr. D. Anu Disney, M.E., Ph.D.**, is submitted in partial fulfilment of the requirements for the award of Bachelor of Engineering degree in Computer Science and Engineering with specialization in ~~Cyber Security~~ **AI & ML**.

DATE: 07/05/2024

PLACE: Chennai

Sandipa Saha
SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management of SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T.Sasikala M.E., Ph.D., Dean, School of Computing, Dr. S.Vigneshwari M.E., Ph.D., Head of the Department of Computer Science and Engineering** for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Internal Guide **Dr. D. Anu Disney, M.E., Ph.D.,** for her valuable guidance, suggestions and constant encouragement which paved way for the successful completion of my phase-1 professional Training.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Computer Science and Engineering** who were helpful in many ways for the completion of the project.

COURSE CERTIFICATE



NPTEL Online Certification

(Funded by the MoE, Govt. of India)



This certificate is awarded to

SANDIPA SAHA

for successfully completing the course

Systems and Usable Security

with a consolidated score of **54** %

Online Assignments	22.33/25	Proctored Exam	31.5/75
--------------------	----------	----------------	---------

Total number of candidates certified in this course: **630**

Dr. Devendra Deshmukh,
Dean of Academic Affairs (DOAA),
Indian Institute of Technology Indore

Jan-Feb 2024

(4 week course)

Prof. Andrew Thangaraj
NPTEL, Coordinator
IIT Madras



Indian Institute of Technology Indore



Roll No: NPTEL24CS48S344100862

To verify the certificate



No. of credits recommended: 1 or 2

ABSTRACT

Diabetes is an illness caused because of high glucose level in a human body. Diabetes should not be ignored if it is untreated then Diabetes may cause some major issues in a person like: heart related problems, kidney problem, blood pressure, eye damage and it can also affect other organs of human body. To achieve this goal, in this paper we aim to develop a prediction system using machine learning to detect and classify the presence of diabetes in e-healthcare environment. A significant attention has been made to the accurate detection of diabetes which is a big challenge for the research community to develop a diagnosis system to detect diabetes in a successful way in the e- healthcare environment. To handle these issues, we have proposed diagnosis system using machine learning methods, such as preprocessing of data, feature selection, and classification for the detection of diabetes disease in e- healthcare environment. In this work we will use Machine Learning Classification and ensemble techniques on a dataset to predict diabetes which are K-Nearest Neighbour (KNN), Logistic Regression (LR), Decision Tree (DT), Support Vector Machine (SVM), Gradient Boosting (GB) and Random Forest (RF). The accuracy is different for every model when compared to other models. The Project work gives the accurate or higher accuracy model shows that the model can predict diabetes effectively.

TABLE OF CONTENTS

CHAPTER NO.	TITLE	PAGE NO.
	COURSE CERTIFICATE	v
	ABSTRACT	vi
	LIST OF FIGURES	viii
1	INTRODUCTION	1-5
	1.1 Diabetes	
	1.2 Types of Diabetes	
	1.3 Symptoms of Diabetes	
	1.4 Cause of Diabetes	
	1.5 Estimating Diabetes Data Set	
	1.6 Machine Learning Basics	
	1.7 Importance of Machine Learning	
2	AIM & SCOPE	6-10
	2.1 Aim	
	2.2 Scope	
	2.3 Attribute Information	
	2.4 Data Preparation	
	2.4.1 Imported Libraries	
	2.4.2 Libraries Used	
3	INTRODUCTION	11-20
	3.1 Dataset Description	
	3.1.1 Distribution of Diabetic Patient	
	3.2 Data Preprocessing	
	3.3 Apply Machine Learning	
	3.3.1 Support Vector Machine	
	3.3.2 K- Nearest Neighbour	
	3.3.3 Decision Tree	
	3.3.4 Logistic Regression	
	3.3.5 Random Forest	
	3.3.6 Gradient Boosting	
	3.4 Linear Regression Algorithm	
	3.5 Model Building	
4	EXPERIMENTAL RESULT	21-22
5	SUMMARY AND CONCLUSION	23
	REFERENCES	24
	APPENDIX	25-31
	A. Code and Screenshots	

LIST OF FIGURES

FIGURE NO.	FIGURE NAME	PAGE NO.
1.2.1	Types of Diabetes	2
1.5.1	Given Dataset	4
1.6.1	Concept of Machine Learning	4
1.7.1	Various field covered by Machine Learning	5
2.4.1.1	Imported Libraries	8
2.4.2.1	Example Implementation of Seaborn Library	10
3.1.1	Descriptive Dataset	11
3.1.1.1	Ratio of Diabetic and Non-Diabetic Patient	11
3.3.6.1	Overview of the process	17
3.4.1	Linear Regression Graph	18
3.4.2	Given Dataset	19
4.1	Accuracy Result of Machine Learning Methods	21
4.2	Feature Importance Plot for Random Forest Algorithm	21

CHAPTER 1

INTRODUCTION

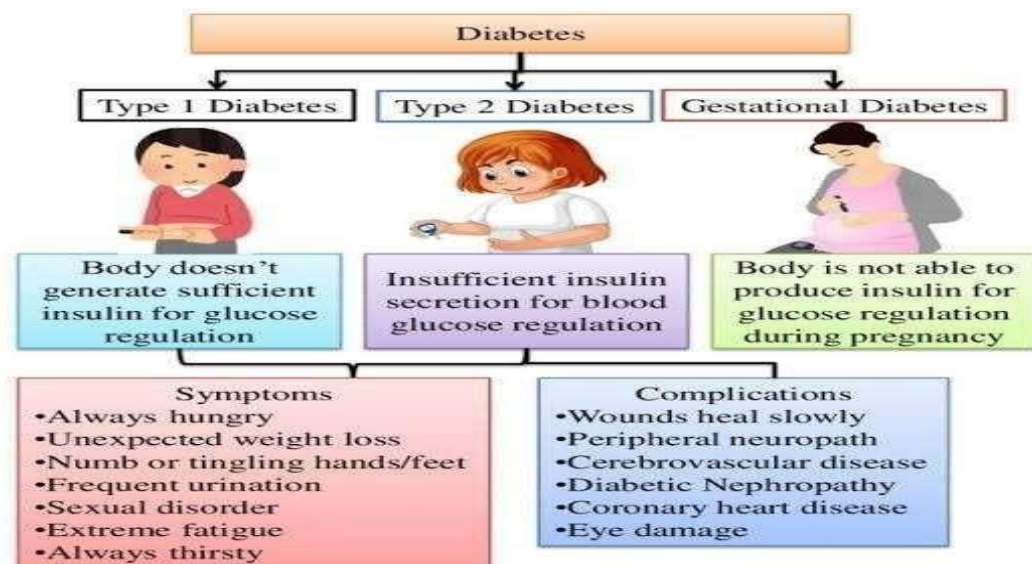
1.1 DIABETES

Diabetes is a noxious disease in the world. Diabetes is caused because of obesity high blood glucose levels, and so forth. It affects the hormone insulin, resulting in the abnormal metabolism of carbs, and improves the level of sugar in the blood. Diabetes occurs when the body does not make enough insulin. According to (WHO) World Health Organization about 422 million people suffer from diabetes, particularly from low- or middle-income countries. And this could be increased to 490 million by the year 2030. However, the prevalence of diabetes is found among various Countries like Canada, China, and India etc. The population of India is now more than 100 million so the actual number of diabetics in India is 40 million. Diabetes is a major cause of death in the world. Early prediction of diseases like diabetes can be controlled and save human life. To accomplish this, this work explores the prediction of diabetes by taking various attributes related to diabetes disease. For this purpose, we use the Pima Indian Diabetes Dataset, we apply various Machine Learning classification and ensemble Techniques to predict diabetes. Machine Learning is a method that is used to train computers or machines explicitly. Various Machine Learning Techniques provide efficient results to collect Knowledge by building various classification and ensemble models from collected datasets. Such collected data can be useful in predicting diabetes. Various techniques of Machine Learning can be capable of doing predictions; however, it is tough to choose the best technique. Thus, for this purpose, we apply popular classification and ensemble methods to the dataset for prediction.

1.2 Type of diabetes

Diabetes can be divided into two categories, type 1 diabetes, and type 2 diabetes.

- Patients with type 1 diabetes are normally younger, mostly less than 30 years old. The typical clinical symptoms are increased thirst and frequent urination, high blood glucose levels. This type of diabetes cannot be cured effectively with oral medications alone and the patients require insulin therapy.
- Type 2 diabetes occurs more commonly in middle-aged and elderly people, and is often associated with the occurrence of obesity, hypertension, dyslipidemia, arteriosclerosis, and other diseases.
- Gestational diabetes appears in pregnant women who suddenly develop high blood sugar. In two-thirds of the cases, it will reappear during subsequent pregnancies. There is a great chance that type 1 or type 2 diabetes will occur after a pregnancy affected by gestational diabetes.



Fig(1.2.1)-Types of Diabetes

1.3 Symptoms of Diabetes:

- Frequent Urination
- Increased thirst
- Tired/Sleepiness
- Weight loss
- Blurred vision
- Mood swings
- Confusion and difficulty concentrating
- Frequent Infections

1.4 Cause of Diabetes:

Genetic factors are the main cause of diabetes. It is caused by at least two mutant genes in chromosome 6, the chromosome that affects the response of the body to various antigens. Viral infection may also influence the occurrence of type 1 and type 2 diabetes. Studies have shown that infection with viruses such as rubella, Coxsackievirus, mumps, hepatitis B virus, and cytomegalovirus increases the risk of developing diabetes.

1.5 Estimating Diabetes Data Set Information:

In this dataset 422 million people worldwide have diabetes, the majority living in low-and middle-income countries, and 1.6 million deaths are directly attributed to diabetes each year. Both the number of cases and the prevalence of diabetes have been steadily increasing over the past few decades.

diabetes_data_upload - Excel

FileHomeInsertPage LayoutFormulasDataReviewViewHelp

Tell me what you want to do

Share

Calibri

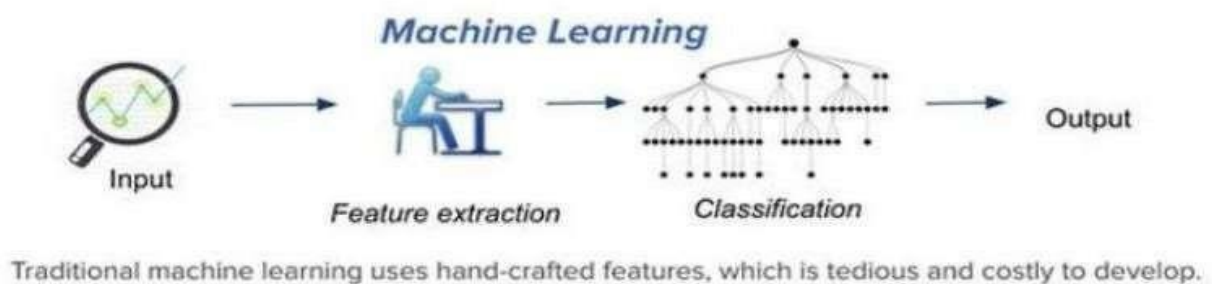
11

<

Fig(1.5.1)-Given Dataset

1.6 MACHINE LEARNING BASICS

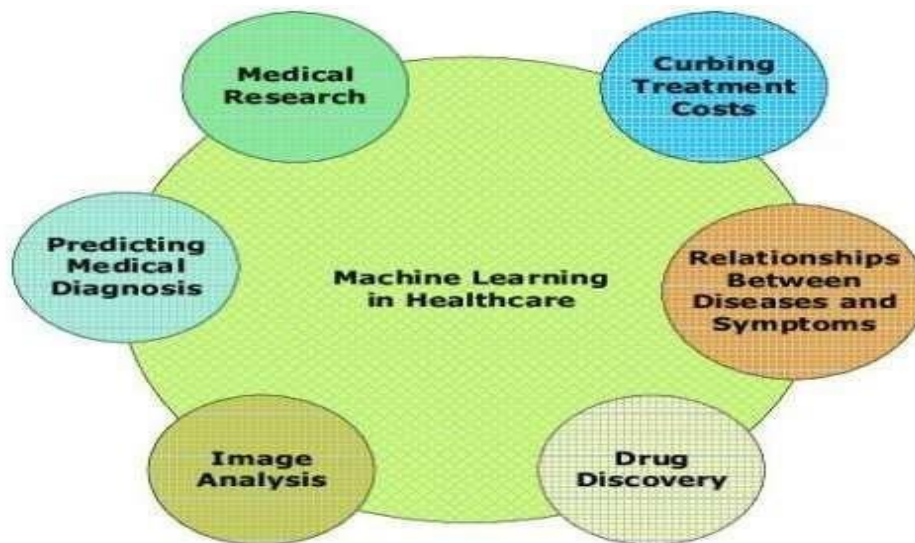
Machine learning is a method of data analysis that automates analytical model building. It is a branch of AI (artificial intelligence) based on the idea that system can learn from data, identify patterns, and make decisions with minimal human intervention.



Fig(1.6.1)-Concept of Machine Learning

1.7 Importance of machine learning

Some of the factors that made machine learning as most important one will be data mining and Bayesian analysis. Things like growing volumes and varieties of available data, computational processing that is cheaper and more powerful, and affordable data storage. While many machine learning algorithms have been around for a long time, the ability to automatically apply complex mathematical calculations to big data – over and over, faster, and faster is recent development.



Fig(1.7.1)- Various fields covered by Machine Learning

CHAPTER-2

AIM AND SCOPE OF PRESENT INVESTIGATION

2.1 AIM:

To predict the Diabetes in the given Diabetes dataset using features from the given Data set.

2.2 SCOPE:

There is need of regional studies prediction in India. The early intervention can reduce the prevalence of diabetes and hence the economic burden due to it. Diabetes Prediction is becoming the area of interest for researchers to train the program to identify the patient are diabetic or not by applying proper classifier on the dataset. Based on previous research work, it has been observed that the classification process is not much improved. Hence a system is required as Diabetes Prediction is important area in computers, to handle the issues identified based on previous research.

1. Data set characteristics are Multivariate.
2. Attribute characteristic' s are categorical, integer, real.
3. Associated task – Classification
4. No. of Instances – 520
5. No. of Attributes – 17

2.3 ATTRIBUTES INFORMATION:

Age:	20-65
Sex:	Male,Female (1,0)
Polyuria	Yes, No (1,0)
Polydipsia	Yes, No (1,0)
Sudden Weight Loss	Yes, No (1,0)
Weakness	Yes, No (1,0)
Polyphagia	Yes, No (1,0)
Genital thrush	Yes, No (1,0)
Visual Blurring	Yes, No (1 ,0)
Itching	Yes, No (1,0)
Irritability	Yes, No (1,0)
Delayed Healing	Yes, No (1,0)
Partial Paresis	Yes, No (1 ,0)
Muscle Stiffness	Yes, No (1,0)
Alopecia	Yes, No (1,0)
Obesity	Yes, No (1,0)

Class:

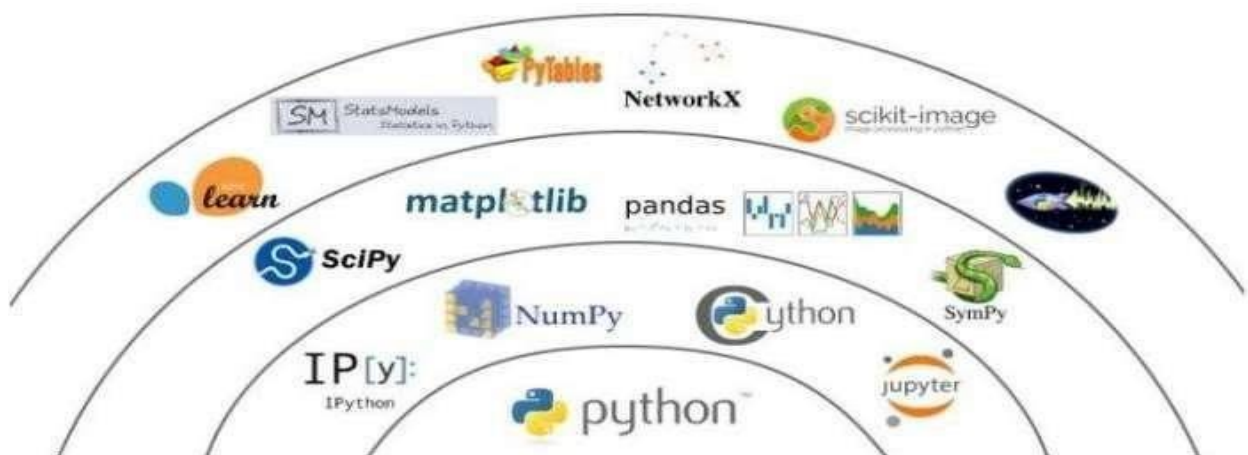
1. Positive
2. Negative.

2.4 DATA PREPARATION:

In this project as a backbone tool python is used to carry out machine learning concepts. With the help of a software called Anaconda Navigator, a Jupiter notebook is launched where it is already installed along with the navigator. Jupiter notebook is an open-source web application that allows to create and share documents and has live code and visualization.

2.4.1 IMPORTED LIBRARIES:

Libraries are collections of prewritten code that users can use to optimize tasks. In project as python is used for implementation tool, it has the most libraries as compared to other programming languages. More than of 60% machine learning developers use and goes for python as it is easy to learn. As python has comparatively large collection of libraries let us look at the libraries that came in handy for mammographic dataset.



Fig(2.4.1.1)-Imported Libraries

2.4.2 LIBRARIES USED:

1. Pandas

It is a widely-used data analysis and manipulation library for python. It provides a lot of functions and methods that expedite the data analysis and preprocessing steps. IT also provides fast, flexible, and expressive data structures working with relational or labeled or both easy and intuitive. Considered as nidamental high- level building block in performing practical, real-world data analysis in python has powerful tools like Data Frame and Series for analyzing.

2. NumPy

NumPy stands for Numerical Python, is a library consisting of multidimensional array objects and a collection of countless of routines for processing those arrays. Using this mathematical and logical operations on arrays can be performed. The difference in using NumPy from pandas is, it works on numerical data whereas pandas on tabular data.

3. Sklearn

Sklearn stands for Scikit-learn, a machine learning library. It is imported for various classification, regression and clustering algorithms including k- means, random forest, support vector machines, gradient boosting and DBSCAN. It is designed using libraries NumPy and SciPy. From the sklearnlibrary and from the tree inside the library Decision Tree Classifier. It is a class capable of performing multi-class classifier on a dataset. When compared with other classifiers, Decision Tree Classifier takes input as two arrays:

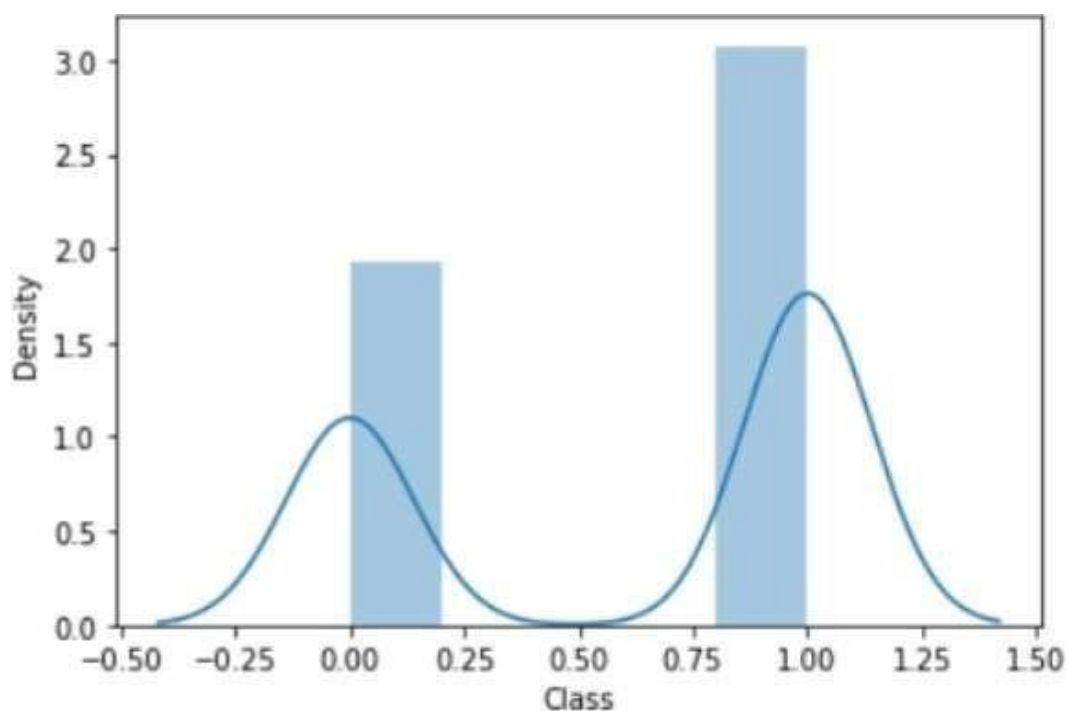
- an array X, a parse or dense, of shape(n_samples,n_features) holding training samples
- an array Y of integer values, shape(n_samples), holding class labels for training sample.

4. Math

Math is a built-in module that you we can use for mathematical tasks. It has set of methods and constants. It is a standard module in python is always available.

5. Seaborn

Seaborn is a library built on top of matplotlib. It used for data visualizationand exploratory data analysis. They work easily with data frames and pandas library. The graphs created can also be customized easily. It provides default styles and color palettes to make statistical plots more attractive. Also closely integrated to the data structures from pandas.



Fig(2.4.2.1)-Example Implementation of Seaborn Library

CHAPTER 3

PROPOSED METHODOLOGY

In this section we shall learn about the various classifiers used in machine learning to predict diabetes. We shall also explain our proposed methodology to improve the accuracy. Different methods were used in this paper. The different methods used are defined below. The output is the accuracy metrics of the machine learning models. Then, the model can be used in prediction.

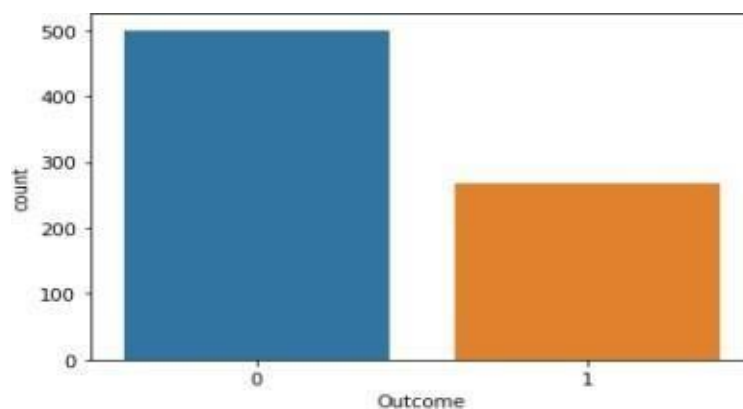
3.1 Dataset Description

S No.	Attributes
1	Pregnancy
2	Glucose
3	Blood Pressure
4	Skin thickness
5	Insulin
6	BMI(Body Mass Index)
7	Diabetes Pedigree Function
8	Age

Fig(3.1.1)-Descriptive Dataset

3.1.1 Distribution of Diabetic Patient

We made a model to predict diabetes however the dataset was slightly imbalanced having around 500 classes labeled as 0 means negative meansno diabetes and 268 labeled as 1 means positive means diabetic.



Fig(3.1.1.1)-Ratio of Diabetic and Non-Diabetic Patient

3.2 Data preprocessing

Data preprocessing is most important process. Mostly healthcare related data contains missing value and other impurities that can cause effectiveness of data. To improve quality and effectiveness obtained after mining process, Data preprocessing is done. To use Machine Learning Techniques on the dataset effectively this process is essential for accurate result and successful prediction. For Pima Indian diabetes dataset we need to perform preprocessing in two steps.

1) Missing Values removal

Remove all the instances that have zero (0) as worth. Having zero as worth is not possible. Therefore, this instance is eliminated. Through eliminating irrelevant features/instances we make feature subset and this process is called features subset selection, which reduces dimensionality of data and help to work faster.

2) Splitting of data

After cleaning the data, data is normalized in training and testing the model. When data is splitted then we train algorithm on the training data set and keep test data set aside. This training process will produce the training model based on logic and algorithms and values of the feature in training data. Basically, aim of normalization is to bring all the attributes under same scale.

3.3 Apply Machine Learning

3.3.1 Support Vector Machine

Support Vector Machine also known as SVM is a supervised machine learning algorithm. SVM is most popular classification technique. SVM creates a hyperplane that separate two classes. It can create a hyperplane

or set of hyperplanes in high dimensional space. This hyper plane can be used for classification or regression also. SVM differentiates instances in specific classes and can also classify the entities which are not supported by data. Separation is done by through hyperplane performs the separation to the closest training point of any class.

Algorithm-

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

3.3.2 K-Nearest Neighbour

KNN is also a supervised machine learning algorithm. KNN helps to solve both the classification and regression problems. KNN is lazy prediction technique. KNN assumes that similar things are near to each other. Many times, data points which are similar are very near to each other. KNN helps to group new work based on similarity measure. KNN algorithm record all the records and classify them according to their similarity measure. For finding the distance between the points uses tree like structure. To make a prediction for a new data point, the algorithm finds the closest data points in the training data set — it is nearest neighbors. Here K= Number of nearby neighbors, it is always a positive integer. Neighbor's value is chosen from set of class. Closeness is mainly defined in terms of Euclidean distance. The Euclidean distance between two points P and Q i.e. P ($p_1, p_2, \dots, p_n$) and Q ($q_1, q_2, \dots, q_n$) is defined by the following equation:-

$$d(P, Q) = \sum_{i=0}^n (P_i - Q_i)^2$$

Algorithm-

- Take a sample dataset of columns and rows named as Pima Indian Diabetes data set.
- Take a test dataset of attributes and rows.
- Find the Euclidean distance by the help of formula-
Euclidean Distance =

$$\sqrt{\sum_{i=1}^y \sum_{j=1}^m \sum_{l=1}^{n-1} ((R_{(j,l)} - P_{(i,l)})^2}$$

- Then with the help of these minimum distance and Euclidean distance find out the nth column of each.
- Then, decide a random value of K. is the no. of nearest neighbours.
- Find out the same output values.

If the values are same, then the patient is diabetic, otherwise not.

3.3.3 Decision Tree-

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

Algorithm

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

3.3.4 Logistic Regression

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

Sigmoid function $P = \frac{1}{1+e^{-(a+bx)}}$ Here P = probability, a and b = parameter of Model.

3.3.5 Random Forest

It is type of ensemble learning method and used for classification and regression tasks. The accuracy it gives is grater then compared to other models. This method can easily handle large datasets. Random Forest is developed by Leo Bremen.

It is popular ensemble Learning Method. Random Forest Improve Performance of Decision Tree by reducing variance. It operates by constructing a multitude of decision trees at training time and outputs the class that is the mode of the classes or classification or mean prediction (regression) of the individual tree.

Algorithm-

- The first step is to select the “R” features from the total features “m” where $R \ll M$.
- Among the “R” features, the node using the best split point.
- Split the node into sub nodes using the best split.
- Repeat a to c steps until “l” number of nodes has been reached.
- Built forest by repeating steps a to do for “a” number of times to create “n” number.

The random forest finds the best split using the Gin-Index Cost Function which is given by:

$$Gini = \sum_{k=1}^n p_k * (1 - p_k) \quad \text{Where } k = \text{Each class} \quad \text{and } p = \text{proportion of training instances}$$

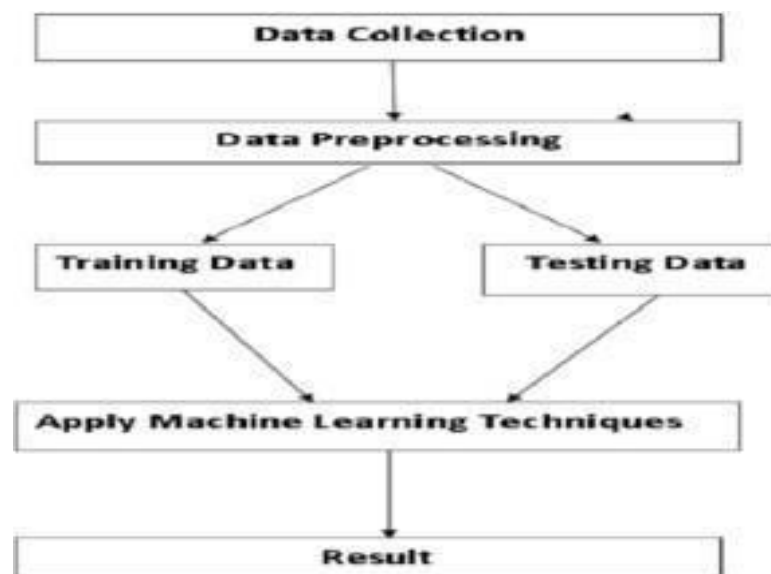
The first step is to need the take a glance at choices and use the foundations of each indiscriminately created decision tree to predict the result and stores the anticipated outcome at intervals the target place. Secondly, calculate the votes for each predicted target and ultimately, admit the high voted predicted target because of the ultimate prediction from the random forest formula. Some of the options of Random Forest does correct predictions result for a spread of applications are offered.

3.3.6 Gradient Boosting

Gradient Boosting is most powerful ensemble technique used for prediction and it is a classification technique. It combines weak learner together to make strong learner models for prediction. It uses Decision Tree model. It classifies complex data sets and it is very effective and popular method. In gradient boosting model performance improve over iterations.

Algorithm-

- Consider a sample of target values as P
- Estimate the error in target values.
- Update and adjust the weights to reduce error M .
- $P[x] = p[x] + \alpha M[x]$
- Model Learners are analysed and calculated by loss function F
- Repeat steps till desired & target result P .

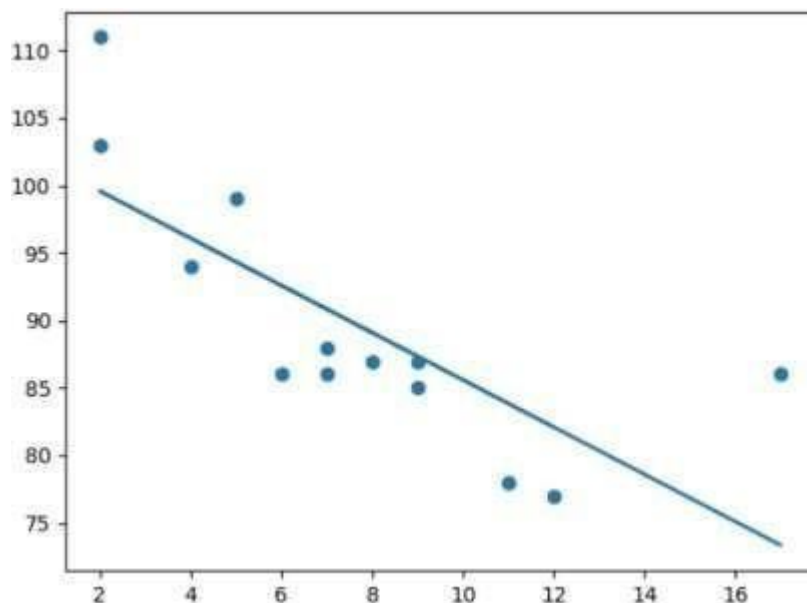


Fig(3.3.6.1)-Overview of the process

3.4 LINEAR REGRESSION ALGORITHM

Linear Regression is a machine learning algorithm based on supervised learning. It performs a regression task. Regression models a target prediction value based on independent variables. It is mostly used for finding out the relationship between variables and forecasting. Different regression models differ based on – the kind of relationship between dependent and independent variables, they are considering and the number of independent variables being used.

$$J = \frac{1}{n} \sum_{i=1}^n (pre_i - y_i)^2$$



Fig(3.4.1)-Linear Regression Graph

Implementation of linear Regression –

1. Import the packages and classes you need.
2. Provide data to work with and eventually do appropriate transformations.
3. Create a regression model and fit it with existing data.

4. Check the results of model fitting to know whether the model is satisfactory.
5. Apply the model for predictions

	Age	Gender	Polyuria	Polydipsia	sudden weight loss	weakness	Polyphagia	Genital thrush	visual blurring	itching	Irritability	delayed healing	partial paresis	muscle stiffness	Alopecia	Obesity	class
0	40	1	0	1	0	1	0	0	0	1	0	1	0	1	1	1	1
1	58	1	0	0	0	1	0	0	1	0	0	0	1	0	1	0	1
2	41	1	1	0	0	1	1	0	0	1	0	1	0	1	1	0	1
3	45	1	0	0	1	1	1	1	0	1	0	1	0	0	0	0	1
4	60	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1
15	39	0	1	1	1	0	1	0	0	1	0	1	1	0	0	0	1
16	48	0	1	1	1	1	1	0	0	1	1	1	1	0	0	0	1
17	58	0	1	1	1	1	1	0	1	0	0	0	1	1	0	1	1
18	32	0	0	0	0	1	0	0	1	1	0	1	0	0	1	0	0
19	42	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fig(3.4.2)-Given Dataset

3.5 MODEL BUILDING

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

Procedure of Proposed Methodology

Step1: Import required libraries, Import diabetes

dataset. Step2: Pre-process data to remove missing data.

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

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

Step5: Build the classifier model for the mentioned machine learning algorithm based on training set.

Step6: Test the Classifier model for the mentioned machine learning algorithm based on test set.

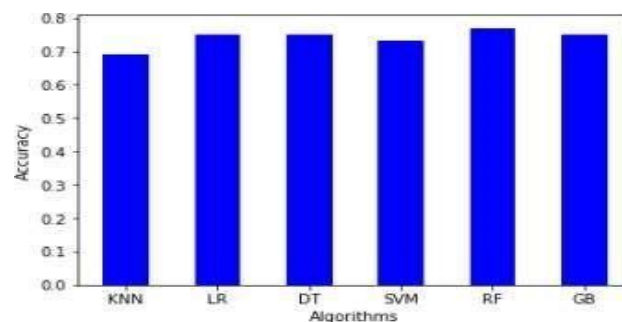
Step7: Perform Comparison Evaluation of the experimental performance results obtained for each classifier.

Step8: After analyzing based on various measures conclude the best performing algorithm.

CHAPTER 4

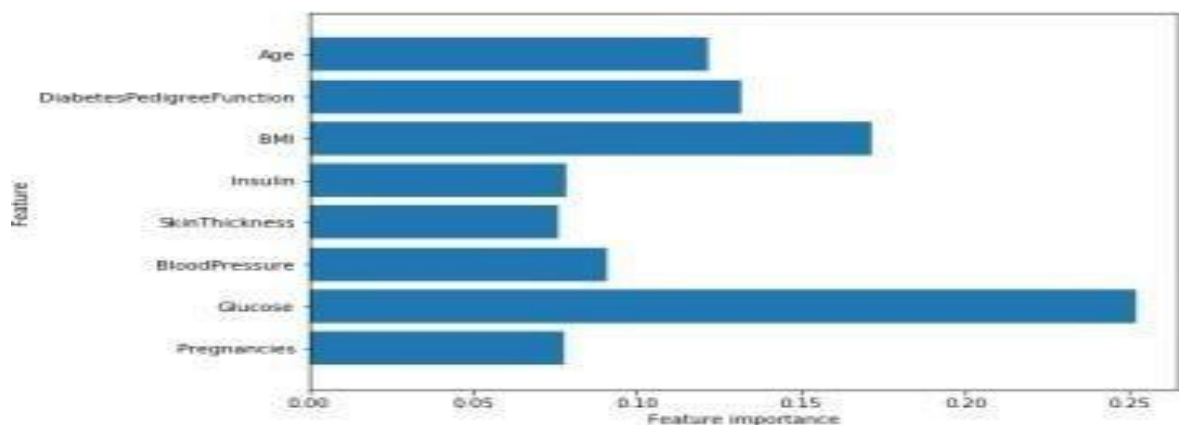
EXPERIMENTAL RESULTS

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



Fig(4.1)-Accuracy Result of Machine Learning Methods

Here feature played important role in prediction is presented for random forest algorithm. The sum of the importance of each feature playing major role for diabetes have been plotted, where X-axis represents the importance of each feature and Y-Axis the names of the features.



Fig(4.2)-Feature Importance Plot for Random Forest Algorithm

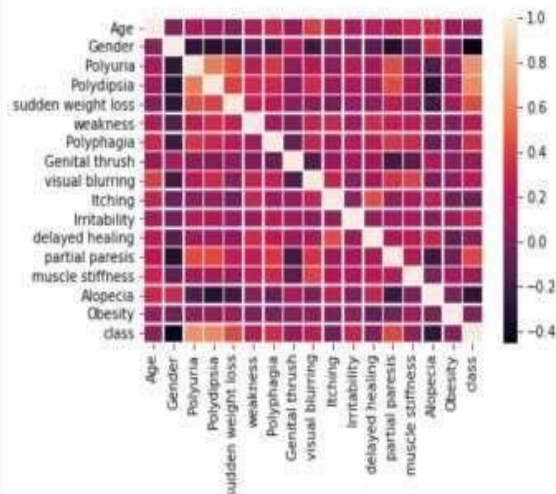
CORRELATION MATRIX

```
df.corr()
```

	Age	Gender	Polyuria	Polydipsia	sudden weight loss	weakness	Polyphagia	Genital thrush	visual blurring	itching	Irritability	delayed healing	partial paresis	muscle stiffness
Age	1.000000	0.062672	0.199781	0.137382	0.064808	0.224506	0.315577	0.096519	0.402729	0.296558	0.201625	0.257501	0.232742	0.307
Gender	0.062672	1.000000	-0.268894	-0.312262	-0.281840	-0.124490	-0.219968	0.208061	-0.208082	-0.052496	-0.013735	-0.101978	-0.332288	-0.090
Polyuria	0.199781	-0.268894	1.000000	0.598009	0.447207	0.263000	0.373873	0.007273	0.235095	0.088289	0.237740	0.149873	0.441664	0.152
Polydipsia	0.137382	-0.312262	0.598009	1.000000	0.406805	0.332453	0.316839	0.028001	0.331250	0.128716	0.203446	0.115691	0.442249	0.180
sudden weight loss	0.064808	-0.281840	0.447207	0.406805	1.000000	0.282884	0.243511	0.089858	0.088754	-0.004518	0.140340	0.088140	0.264014	0.109
weakness	0.224506	-0.124490	0.263000	0.332453	0.282884	1.000000	0.180266	0.027780	0.301043	0.309440	0.146608	0.335507	0.272082	0.263
Polyphagia	0.315577	-0.219968	0.373873	0.316839	0.243511	0.180266	1.000000	-0.063712	0.293545	0.144390	0.259466	0.263080	0.373586	0.320
Genital thrush	0.096519	0.208061	0.007273	0.028001	0.089858	0.027780	-0.063712	1.000000	-0.148408	0.125330	0.160551	0.138111	-0.195612	-0.100

```
import seaborn as sns
sns.heatmap(df.corr(),linewidths=0.5)
```

<AxesSubplot:>



CHAPTER 5

SUMMARY AND CONCLUSION

Diabetes mellitus is a disease, which can cause many complications. How to exactly predict and diagnose this disease by using machine learning is worthy studying. The main aim of this project was to design and implement Diabetes Prediction Using Machine Learning Methods and Performance Analysis of that methods and it has been achieved successfully. The proposed approach uses various classification and ensemble learning method in which SVM, Knn, Random Forest, Decision Tree, Logistic Regression and Gradient Boosting classifiers are used. The Experimental results can be asst health care to take early prediction and make early decision to cure diabetes and save humans life.



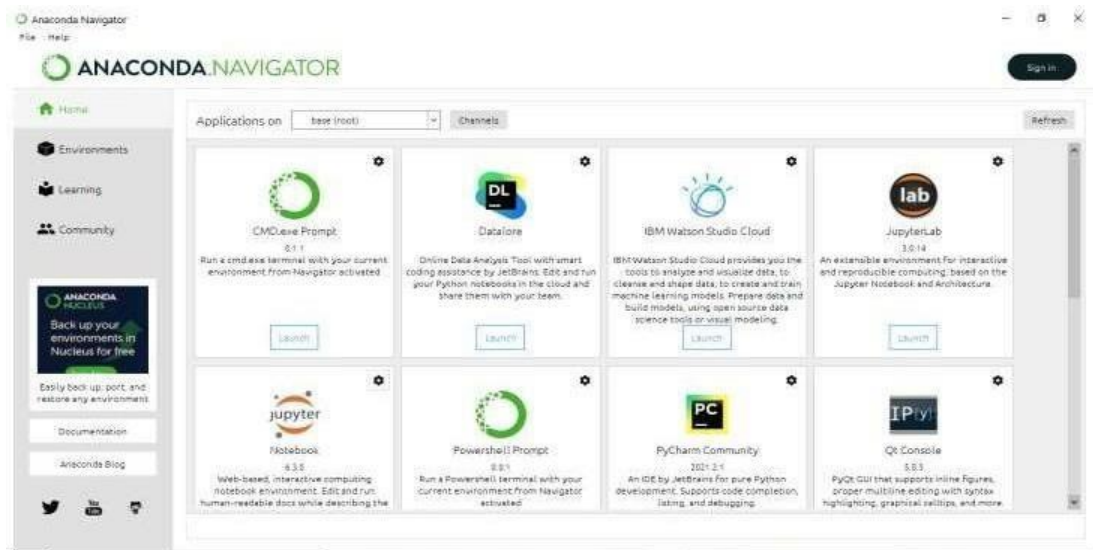
REFERENCES

- [1] Debadri Dutta, Debpriyo Paul, Parthajeet Ghosh, "Analyzing Feature Importance's for Diabetes Prediction using Machine Learning". IEEE, pp 942-928, 2018.
- [2] Md. Faisal Faruque, Asaduzzaman, Iqbal H. Sarker, "Performance Analysis of Machine Learning Techniques to Predict Diabetes Mellitus". International Conference on Electrical, Computer and Communication Engineering (ECCE), 7-9 February, 2019.
- [3] Tejas N. Joshi, Prof. Pramila M. Chawan, "Diabetes Prediction Using Machine Learning Techniques".Int. Journal of Engineering Research and Application, Vol. 8, Issue 1, (Part -II) January 2018, pp.-09-13.
- [4] A.K., Dewangan, and P., Agrawal, "Classification of Diabetes Mellitus Using Machine Learning Techniques," International Journal of Engineering and Applied Sciences, vol. 2, 2015.

APPENDIX

WORKING ENVIRONMENT:

ANACONDA NAVIGATOR is desktop GUI used to launch applications and manage packages in one place. Outlook.



CODING ENVIRONMENT:

Jupyter notebook from the anaconda navigator is launched along with all the preinstalled packages for python.



SCREENSHOTS

The first screenshot shows the Jupyter Notebook interface with the title 'diabetes'. The notebook is running on a local host at localhost:8889. The interface includes a menu bar (File, Edit, View, Insert, Cell, Kernel, Widgets, Help) and a toolbar with icons for file operations, running cells, and markdown. The notebook content is divided into two sections: 'Step 0: Import libraries and Dataset' and 'Step 1: Descriptive Statistics'.

Step 0: Import libraries and Dataset

```
In [1]: # Importing Libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

import warnings
warnings.filterwarnings('ignore')

In [2]: # Importing dataset
dataset = pd.read_csv('diabetes.csv')
```

Step 1: Descriptive Statistics

```
In [3]: # Preview data
dataset.head()
```

Out[3]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	6	148	72	35	0	33.6	0.627	50	1
1	1	85	66	29	0	26.6	0.351	31	0

The second screenshot shows the same Jupyter Notebook interface, but with the 'Step 1: Descriptive Statistics' section expanded. It includes the same code for importing the dataset and previewing the data, followed by additional code for checking the dataset dimensions and features data-type.

```
In [2]: # Importing dataset
dataset = pd.read_csv('diabetes.csv')

In [3]: # Preview data
dataset.head()
```

Out[3]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	6	148	72	35	0	33.6	0.627	50	1
1	1	85	66	29	0	26.6	0.351	31	0
2	8	183	64	0	0	23.3	0.672	32	1
3	1	89	66	23	94	28.1	0.167	21	0
4	0	137	40	35	168	43.1	2.288	33	1

```
In [4]: # Dataset dimensions - (rows, columns)
dataset.shape

Out[4]: (768, 9)

In [5]: # Features data-type
dataset.info()
```

Desktop/ml/ x diabetes - Jupyter Notebook x +

localhost:8889/notebooks/Desktop/ml/diabetes.ipynb

jupyter diabetes Last Checkpoint: Last Saturday at 10:32 AM (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 (ipykernel)

4 0 137 40 35 168 43.1 2.288 33 1

```
In [4]: # Dataset dimensions - (rows, columns)
dataset.shape

Out[4]: (768, 9)
```

```
In [5]: # Features data-type
dataset.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 768 entries, 0 to 767
Data columns (total 9 columns):
#   Column                Non-Null Count  Dtype
---  ---
0   Pregnancies            768 non-null    int64
1   Glucose                768 non-null    int64
2   BloodPressure          768 non-null    int64
3   SkinThickness          768 non-null    int64
4   Insulin                768 non-null    int64
5   BMI                   768 non-null    float64
6   DiabetesPedigreeFunction 768 non-null    float64
7   Age                   768 non-null    int64
8   Outcome                768 non-null    int64
dtypes: float64(2), int64(7)
memory usage: 54.1 KB
```

Type here to search

Desktop/ml/ x diabetes - Jupyter Notebook x +

localhost:8889/notebooks/Desktop/ml/diabetes.ipynb

jupyter diabetes Last Checkpoint: Last Saturday at 10:32 AM (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 (ipykernel)

```
In [6]: # Statistical summary
dataset.describe().T

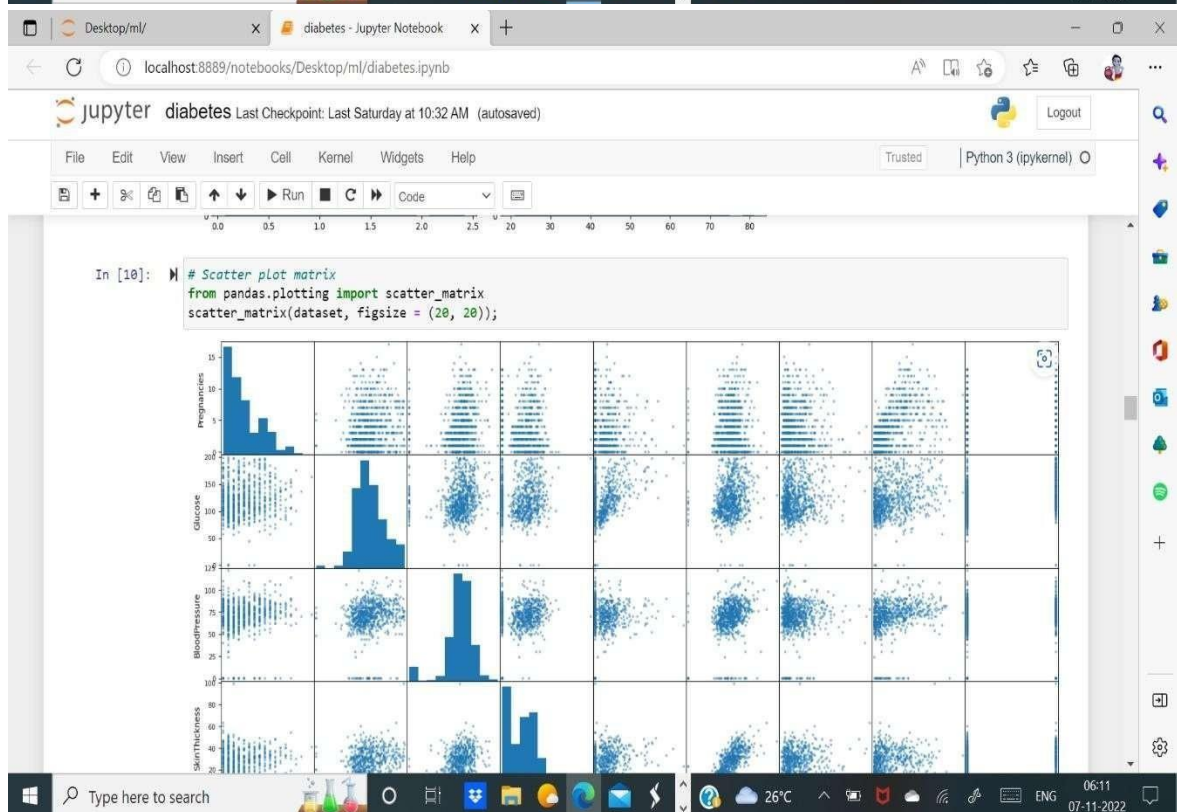
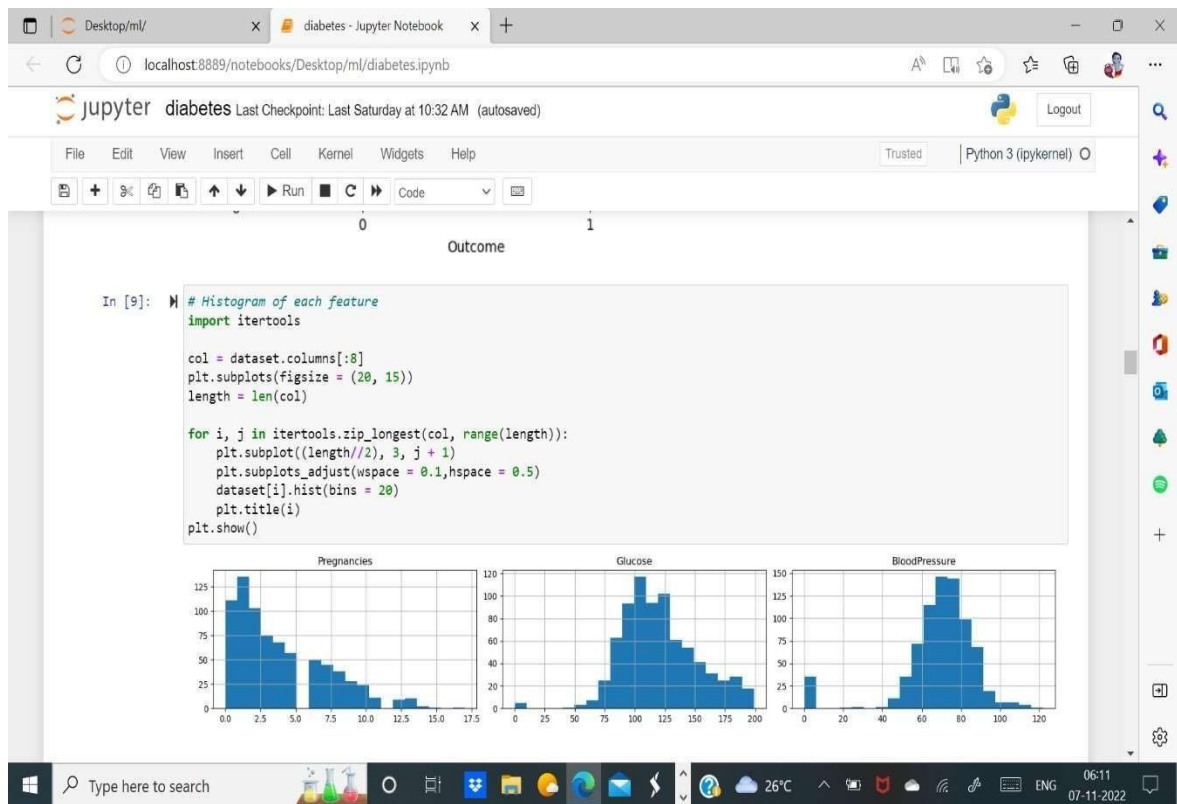
Out[6]:
```

	count	mean	std	min	25%	50%	75%	max
Pregnancies	768.0	3.845052	3.369578	0.000	1.00000	3.00000	6.00000	17.00
Glucose	768.0	120.894531	31.972618	0.000	99.00000	117.00000	140.25000	199.00
BloodPressure	768.0	69.105469	19.355807	0.000	62.00000	72.00000	80.00000	122.00
SkinThickness	768.0	20.536458	15.952218	0.000	0.00000	23.00000	32.00000	99.00
Insulin	768.0	79.799479	115.244002	0.000	0.00000	30.50000	127.25000	846.00
BMI	768.0	31.992578	7.884160	0.000	27.30000	32.00000	36.60000	67.10
DiabetesPedigreeFunction	768.0	0.471876	0.331329	0.078	0.24375	0.3725	0.62625	2.42
Age	768.0	33.240885	11.760232	21.000	24.00000	29.00000	41.00000	81.00
Outcome	768.0	0.348958	0.476951	0.000	0.00000	0.00000	1.00000	1.00

```
In [7]: # Count of null values
dataset.isnull().sum()

Out[7]: Pregnancies      0
Glucose                0
BloodPressure          0
```

Type here to search



Desktop/ml/ x diabetes - Jupyter Notebook x +

localhost:8889/notebooks/Desktop/ml/diabetes.ipynb

Jupyter diabetes Last Checkpoint: Last Saturday at 10:32 AM (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 (ipykernel)

Step 3: Data Preprocessing

```
In [13]: dataset_new = dataset
```

```
In [14]: # Replacing zero values with NaN
dataset_new[["Glucose", "BloodPressure", "SkinThickness", "Insulin", "BMI"]] = dataset_new[["Glucose", "BloodPressure", "SkinThickness", "Insulin", "BMI"]].replace(0, np.NaN)
```

```
In [15]: # Count of NaN
dataset_new.isnull().sum()
```

```
Out[15]: Pregnancies      0
Glucose      5
BloodPressure 35
SkinThickness 227
Insulin      374
BMI          11
DiabetesPedigreeFunction 0
Age          0
Outcome      0
dtype: int64
```

```
In [16]: # Replacing NaN with mean values
dataset_new["Glucose"].fillna(dataset_new["Glucose"].mean(), inplace = True)
dataset_new["BloodPressure"].fillna(dataset_new["BloodPressure"].mean(), inplace = True)
```

Type here to search

Desktop/ml/ x diabetes - Jupyter Notebook x +

localhost:8889/notebooks/Desktop/ml/diabetes.ipynb

Jupyter diabetes Last Checkpoint: Last Saturday at 10:32 AM (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 (ipykernel)

```
In [18]: # Feature scaling using MinMaxScaler
from sklearn.preprocessing import MinMaxScaler
sc = MinMaxScaler(feature_range = (0, 1))
dataset_scaled = sc.fit_transform(dataset_new)
```

```
In [19]: dataset_scaled = pd.DataFrame(dataset_scaled)
```

```
In [20]: # Selecting features - [Glucose, Insulin, BMI, Age]
X = dataset_scaled.iloc[:, [1, 4, 5, 7]].values
Y = dataset_scaled.iloc[:, 8].values
```

```
In [21]: # Splitting X and Y
from sklearn.model_selection import train_test_split
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size = 0.20, random_state = 42, stratify = dataset_new['Outcome'])
```

```
In [22]: # Checking dimensions
print("X_train shape:", X_train.shape)
print("X_test shape:", X_test.shape)
print("Y_train shape:", Y_train.shape)
print("Y_test shape:", Y_test.shape)
```

```
X_train shape: (614, 4)
X_test shape: (154, 4)
Y_train shape: (614,)
Y_test shape: (154,)
```

Type here to search

Desktop/ml/ x diabetes - Jupyter Notebook x +

localhost:8889/notebooks/Desktop/ml/diabetes.ipynb

Jupyter diabetes Last Checkpoint: Last Saturday at 10:32 AM (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 (ipykernel)

Step 4: Data Modelling

```
In [23]: # Logistic Regression Algorithm
from sklearn.linear_model import LogisticRegression
logreg = LogisticRegression(random_state = 42)
logreg.fit(X_train, Y_train)

Out[23]: LogisticRegression(random_state=42)
```

```
In [24]: # Plotting a graph for n_neighbors
from sklearn import metrics
from sklearn.neighbors import KNeighborsClassifier

X_axis = list(range(1, 31))
acc = pd.Series()
x = range(1,31)

for i in list(range(1, 31)):
    knn_model = KNeighborsClassifier(n_neighbors = i)
    knn_model.fit(X_train, Y_train)
    prediction = knn_model.predict(X_test)
    acc = acc.append(pd.Series(metrics.accuracy_score(prediction, Y_test)))

plt.plot(X_axis, acc)
plt.xticks(x)
plt.title("Finding best value for n_estimators")
plt.xlabel("n_estimators")
plt.ylabel("Accuracy")
plt.grid()
```

Type here to search 26°C 06:12 07-11-2022

Desktop/ml/ x diabetes - Jupyter Notebook x +

localhost:8889/notebooks/Desktop/ml/diabetes.ipynb

Jupyter diabetes Last Checkpoint: Last Saturday at 10:32 AM (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 (ipykernel)

```
In [25]: # K nearest neighbors Algorithm
from sklearn.neighbors import KNeighborsClassifier
knn = KNeighborsClassifier(n_neighbors = 24, metric = 'minkowski', p = 2)
knn.fit(X_train, Y_train)

Out[25]: KNeighborsClassifier(n_neighbors=24)
```

```
In [26]: # Support Vector Classifier Algorithm
from sklearn.svm import SVC
svc = SVC(kernel = 'linear', random_state = 42)
svc.fit(X_train, Y_train)

Out[26]: SVC(kernel='linear', random_state=42)
```

```
In [27]: # Naive Bayes Algorithm
from sklearn.naive_bayes import GaussianNB
nb = GaussianNB()
nb.fit(X_train, Y_train)

Out[27]: GaussianNB()
```

```
In [28]: # Decision tree Algorithm
from sklearn.tree import DecisionTreeClassifier
dectree = DecisionTreeClassifier(criterion = 'entropy', random_state = 42)
dectree.fit(X_train, Y_train)

Out[28]: DecisionTreeClassifier(criterion='entropy', random_state=42)
```

Type here to search 26°C 06:12 07-11-2022

Desktop/ml/ diabetes - Jupyter Notebook

localhost:8889/notebooks/Desktop/ml/diabetes.ipynb

jupyter diabetes Last Checkpoint: Last Saturday at 10:32 AM (autosaved)

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 (ipykernel)

Step 5: Model Evaluation

```
In [31]: # Evaluating using accuracy_score metric
from sklearn.metrics import accuracy_score
accuracy_logreg = accuracy_score(Y_test, Y_pred_logreg)
accuracy_knn = accuracy_score(Y_test, Y_pred_knn)
accuracy_svc = accuracy_score(Y_test, Y_pred_svc)
accuracy_nb = accuracy_score(Y_test, Y_pred_nb)
accuracy_dectree = accuracy_score(Y_test, Y_pred_dectree)
accuracy_ranfor = accuracy_score(Y_test, Y_pred_ranfor)
```

```
In [32]: # Accuracy on test set
print("Logistic Regression: " + str(accuracy_logreg * 100))
print("K Nearest neighbors: " + str(accuracy_knn * 100))
print("Support Vector Classifier: " + str(accuracy_svc * 100))
print("Naive Bayes: " + str(accuracy_nb * 100))
print("Decision tree: " + str(accuracy_dectree * 100))
print("Random Forest: " + str(accuracy_ranfor * 100))
```

Logistic Regression: 72.07792207792207
 K Nearest neighbors: 78.57142857142857
 Support Vector Classifier: 73.37662337662337
 Naive Bayes: 71.42857142857143
 Decision tree: 68.18181818181817
 Random Forest: 75.97402597402598

Type here to search 26°C 06:12 07-11-2022

Desktop/ml/ diabetes - Jupyter Notebook

localhost:8889/notebooks/Desktop/ml/diabetes.ipynb

jupyter diabetes Last Checkpoint: Last Saturday at 10:32 AM (autosaved)

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 (ipykernel)

```
In [35]: # Heatmap of Confusion matrix
sns.heatmap(pd.DataFrame(cm), annot=True)
```

Out[35]: <AxesSubplot>

	0	1
0	87	13
1	20	34

Type here to search 26°C 06:13 07-11-2022