A project report on

DIABETES MELLITUS PREDICTION USING MACHINE LEARNING ALGORITHM

Submitted in partial fulfillment for the award of the degree of

M.Tech (Software Engineering)

By

S V JEEVANANDHAM (18MIS0389)

SCHOOL OF INFORMATION TECHNOLOGY & ENGINEERING

April, 2023

DECLARATION

I here by declare that the thesis entitled "**DIABETES MELLITUS PREDICTION USING MACHINE LEARNING ALGORITHM**" submitted by me, for the award of the degree of M.Tech (Software Engineering) is a record of bonafide work carried out by me under the supervision of " **PROF - KARTHIKEYAN D** "

I further declare that the work reported in this thesis has not been submitted and will not be submitted, either in part or in full, for the award of any other degree or diploma in this institute or any other institute or university.

Place: Vellore

Date:

S V JEEVANANDHAM

Signature of the Candidate

CERTIFICATE

This is to certify that the thesis entitled " **DIABETES MELLITUS PREDICTION USING MACHINE LEARNING ALGORITHM** " submitted by **SV JEEVANANDHAM (18MIS0389)**, School of Information Technology & Engineering, Vellore Institute of Technology, Vellore for the award of the degree M.Tech (Software Engineering) is a record of bonafide work carried out by him/her under my supervision **PROF KARTHIKEYAN D**

The contents of this report have not been submitted and will not be submitted either in part or in full, for the award of any other degree or diploma in this institute or any other institute or university. The Project report fulfills the requirements and regulations of **VELLORE INSTITUTE OF TECHNOLOGY, VELLORE** and in my opinion meets the necessary standards for submission.

SIGNATURE OF THE GUIDE **SIGNATURE OF THE HOD**

INTERNAL EXAMINER **EXTERNAL EXAMINER**

ABSTRACT

Diabetes mellitus is fourth most high mortality rate diseases in the world and it is also a cause of kidney disease, blindness, and heart diseases.This study proposed to predict diabetes using data mining techniques. It is the data analysis process used to classify and predict the disease diabetes from famous dataset of 2080 patients, where data can be splitted into training and testing phases. Training dataset is used to categorize the data to develop the model whereas the performance of the classifier can be determined by using testing dataset. Accuracy is the performance measure for our analysis of prediction to find how many patients records are correctly classified. It can be observed by using three data mining algorithms like Bayesian classification, decision tree, k-nearest neighbor, Logistic Regression, Support vector classifier, Random forest algorithm. Therefore, data mining algorithms in data mining can be applied to our pima India dataset which makes valuable predictions and conclusions. Accuracy may be varied based on the conditions like the size of the dataset considered, number of attributes and type of attributes taken etc. In this project we give the accuracy of data mining algorithms Bayesian classification, knn, decision tree, Logistic Regression, SVC, Random forest. when applied on the datasets differing in size and number of attributes we show that output in graphical format. Also we compare the accuracy of these three algorithms and choose which is more suitable for diabetes prediction.

ACKNOWLEDGEMENT

It is my pleasure to express with deep sense of gratitude to **PROF KARTHIKEYAN** D, VELLORE, SCHOOL OF INFORMATION TECHNOLOGY & ENGINEERING , Vellore Institute of Technology,for his/her constant guidance, continual encouragement,understanding; more than all, he/she taught me patience in my endeavor. My association with him / her is not confined to academics only, but it is a great opportunity on my part of work with an intellectual and expert in the field of Machine Learning .

I would like to express my gratitude to DR.G.VISWANATHAN, Chancellor VELLORE INSTITUTE OF TECHNOLOGY, VELLORE, MR. SANKAR VISWANATHAN, DR. SEKAR VISWANATHAN, MR.G V SELVAM, Vice – Presidents VELLORE INSTITUTE OF TECHNOLOGY, VELLORE, DR. RAMBABU KODALI, Vice – Chancellor, DR. S. NARAYANAN, Pro- Vice Chancellor and Dr. S. Sumathy, Dean, School of Information Technology & Engineering (SITE), for providing with an environment to work in and for his inspiration during the tenure of the course.

In jubilant mood I express ingeniously my whole-hearted thanks to Dr.Shantharajah S.P, Program chair & HoD/Professor, all teaching staff and members working as limbs of our university for their not-self- centered enthusiasm coupled with timely encouragements showered on me with zeal, which prompted the acquirement of the requisite knowledge to finalize my course study successfully. I would like to thank my parents for their support.

It is indeed a pleasure to thank my friends who persuaded and encouraged me to take up and complete this task. At last but not least, I express my gratitude and appreciation to all those who have helped me directly or indirectly toward the successful completion of this project.

Place: Vellore S V JEEVANANDHAM

Date: **Name of the student :**

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

1.1 GENERAL

Diabetes is one of the deadliest diseases in the world. It is not only a disease but also creator of different kinds of diseases like heart attack, blindness etc. The normal identifying process is that patients need to visit a diagnostic center, consult their doctor, and sit tight for a day or more to get their reports. Cause of Diabetes vary depending on the genetic makeup, family history, ethnicity, health etc. Diabetes & pre-diabetes is diagnosed by blood test.

Diabetes mellitus is fourth most high mortality rate diseases in the world and it is also a cause of kidney disease, blindness, and heart diseases. This study proposed to predict diabetes using machine learning algorithms. It is the data analysis process used to classify and predict the disease diabetes from famous dataset of 2200 patients, where data can be splitted into training and testing phases. Training dataset is used to categorize the data to develop the model whereas the performance of the classifier can be determined by using testing dataset.

Accuracy is the performance measure for our analysis of prediction to find how many patients records are correctly classified. It can be observed by using the algorithms like Bayesian classification, decision tree, k-nearest neighbor, Logistic Regression, Support vector classifier, Random forest algorithm. Therefore, these algorithms can be applied to our pima India dataset which makes valuable predictions and conclusions. Accuracy may be varied based on the conditions like the size of the dataset considered, number of attributes and type of attributes taken etc. In this project we give the accuracy of algorithms like Bayesian classification, knn, decision tree, Logistic Regression, SVC, Random forest. when applied on the datasets differing in size and number of attributes we show that output in graphical format. Also we compare the accuracy of these three algorithms and choose which is more suitable for diabetes prediction.

1.2 EXISTING SYSTEM

In recent research works they analyzed the particular dataset using Clustering algorithms, Back Propagation algorithms they have only predicted that the person have diabetes disease or not. They have not go in depth of indicating blood sugar level and diagnosing diabetes. And they have failed to show more accuracy. The predicted accuracy using this algorithms are very low.

Sometimes it failed in prediction of diabetes for some persons. In this project we classify the severity of disease using PYTHON this will be helpful for doctors to diagnose the patients. We will fill the gap by applying Bayesian classification, knn, decision tree, Logistic Regression, SVC, Random forest and we can analyze the accuracy and gives the solution which algorithm is best for predicting diabetes mellitus.

1.3 OBJECTIVE

The objective of the project is to predict Diabetes mellitus using machine learning algorithms by collecting the dataset of patients and preprocessing the dataset to remove null values then splitting those dataset into training and testing phases.Then next building a model and fitting the dataset inside the machine learning algorithms which gives the confussion matrix. From the confussion matrix we calculate the accuracy of the prediction .

1.4 SCOPE OF THE PROJECT :

The Scope of the project is to identify whether the patient is having diabetes or not? And also used to identify the probability of diabetes in patients using machine learning algorithms and creating the user interface.

1.5 ADVANTAGE OF THE PROJECT :

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

CHAPTER 2

LITERATURE SURVEY :

SL. NO.	Name of technique	Domain	Resistance against attacks	Advantages of techniques	Disadvantages of using this technique
1	Improved K- means Cluster algorithm and Logistic regression algorithm for Predicting Type2 Diabetes Mellitus	Temporal Mining and Clustering	Clustering, Prediction, Classification and logistic regression algorithm.	The advantage of this method is that it reduces the bias associated with the random sampling method.The algorithm is Simple and easy to implement algorithm. The accuracy of our purposed model is 95.42%. We have applied two other diabetes atasets.Both experiments results show good performance.	The Hybrid prediction model has uses K-means cluster algorithm which has 92.38% accuracy but using our improved k-means algorithm and logistic regression algorithm we get more accuracy. In future, it can bring in hospital‘s real and latest patients data for continuous training and improving our proposed model.
2	Apriori algorithm, ANN, Random forest model for early prediction of diabetes	Association Rule Mining	Apriori algorith m, ANN, Random forest.	Data preprocessing is to be done to get the correct output. Using this model we get the best accuracy is 74.7%. And the AUROC curve value is 0.806.	Our future work is to select the unstructured data for dataset and these methods will be applied to other medical domains for prediction such as for different types of cancer, psoriasis and Parkinson‘s disease. Other attributes includes physical inactivity, family history of diabetes, and smoking habit, and diagnosis of diabetes.
3	Decision Tree, Naive Bayes, K- nearest neighbor‘s algorithm (k- NN), Classification via Clustering,Neural Network in Review on Prediction of Diabetes Mellitus using Data Mining Technique, Research Paper.	Association and Classification Domain.	In these tools classification measures with discovering the problem by distinguishin g the features of diseases between patients and diagnose or predict the diabetes in Medical Field	Applying Data mining methods and techniques will helps to predict the diabetes and also reduces the treatment cost. Data mining is a techniques used to extract useful information from existing large volume of data which enables you to gain more knowledge.	In this way data mining techniques are applied in medical data domain in order to predict diabetes theoretically and to find out efficient ways to treat them as well in manually.

4	Decision Tree and Naïve Bayes in Diagnosis Of Diabetes Using Classification Mining Techniques, Research Paper.	Association and Classificati -on Domain	The data was divided into training set and test set by the cross validation technique and percentage split technique. In Fuzzy Ant Colony Optimization (ACO) was used dataset to find set of rules for the diabetes diagnosis.	Naïve Bayes are used to model actual diagnosis of diabetes for local and systematic treatment, along with presenting related work in the field.Experimental results show the effectiveness of the proposed model. The performance of the techniques was investigated for the diabetes diagnosis problem.	A developed model for diagnosis of diabetes will require more training data for creation and testing. In future it is not planned to gather the information from different locales over the world and make a more precise and general prescient model for diabetes conclusion.
5	Decision tree algorithm, Random forest, Naïve Bayesian, K nearest neighbor (KNN), Support vector machine in Analysis of diabetes mellitus for early prediction using optimal features selection, Research Paper.	Classification Domain	In this experiment is conducted through rapid miner data mining tool. The performance evaluation of the classification techniques is done through the various performance measure such as accuracy, sensitivity, specificity, and recall, precision.	In analysis, the accuracy of the classification technique is improved, for the predictive task will become faster. In this paper, proposed method improves the accuracy of the classification echniques, hence it is able to map the features effectively from low dimensions to high dimensions.	Some attributes are ignored because of correlation value is less. The attributes value are depend on the correlation value.
6	Methods used are: regression, parameter optimization and tree classification for Chronic disease prediction using administrative data and graph theory of type 2 diabetes	Regression domain	Using methods such as regression and decision tree, there predict the chronic disease for administrativ e data and tree classification	The main advantage is result of these is indicate that graph theory and social network analysis based methods can be effectivelyused for disease risk prediction purposes	It should be Useful for governments and health insure to Identify high.

7	Support vector machine for Machine learning and data mining methods in diabetes research	Classification domain	Support vector machine is more significantly used in big data technology to get accuracy more than 80%	Clinical, diagnostic data are plentiful due to low cost of their retrieval in contrast of other types of data.	It is significant work carried out in all of data mining research and biomarker identification and prediction diagnosis. And also depth exploration towards diagnosis is required
8	(C4.5, SVM, k-NN, PNN, BLR, MLR,PLS-DA, , k eans,Apriori) for Utilization of data mining techniques for prediction of diabetes disease survivability	Prediction domain	Three values (Accuracy, Specificity and Sensitivity) are calculated by Using formula and the prediction one is accuracy	With help of automatic design tools, they easily to reduce a waiting time at the experts.	For medical applications, it is difficult to choose the best algorithm to get accurate prediction
9	Multilayer perceptron, Bayesian classification,J rip,C4.5 for Prediction of diabetes disease using classification data mining techniques	Neural networks domain	Using the mentioned algorithms, they may measure confusion matrix ,distancevect or,accuracy, specificity,se nsitivity which will be used for predict the diabetes disease.	It have lower computation time Medical accuracy and accuracy rate also at the early stages.for this application C4.5 and Jrip give more accuracy above 85%	In future, it helps the doctors take preventive and actions on environment
10	Support Vector Machine (SVM), Naive ayes(NB), K Nearest Neighbor (KNN)and C4.5 Decision tree for measuring Performance analysis of machine learning techniques to predict diabetes mellitus(2019)	Knowledge extraction domain	(Accuracy, Specificity and Sensitivity)ar e calculated by using formula and the prediction one is accuracy	Our experimental results show that C4.5 decision tree achieved higher accuracy compared to other machine learning techniques.	In future, machine learning algorithms are used to provide efficient result to extract correct knowledge.

CHAPTER 3

SYSTEM ARCHITECTURE

3.1. OVERALL ARCHITECTURE DIAGRAM

OUR PROPOSED APPROACH :

The following process is used to better understand the problem and give us better insight of whole process.

Our project approach has six phases:

Data Understanding Phase

- ✓ Collect the data
- ✓ Assess and analyse the data

Data Preparation Phase

- ✓ Clean the data i.e. remove any missing values or outliers etc.
- ✓ Transform the data
- ✓ Select specific data for analysis

Modeling

- ✓ Select appropriate modeling technique

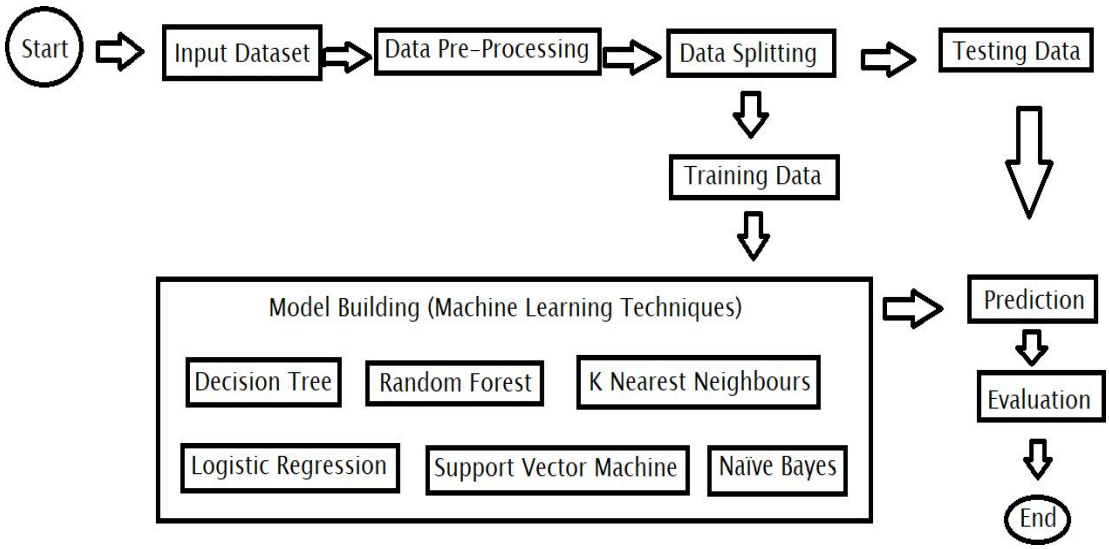
Evaluation

- ✓ Evaluate the model
- ✓ Calculate the accuracy and success rate of the model

Deployment

- ✓ Plan deployment
- ✓ Monitor Deployment
- ✓ Generate reports to test success of the model

ARCHITECTURE DIAGRAM :



CHAPTER 4

DATA UNDERSTANDING

Data Source

The data set comes from the open source standard test data set website Kaggle repository. The data set was obtained by direct questionnaires from 2200 patients at the Sylhet Diabetes Hospital in Sylhet, Bangladesh, and was approved by doctors. The data set is divided into 17 attributes including age, gender, polyuria, Polydipsia, Sudden weight loss,Weakness, Polyphagia, Genital thrush, Visual blurring, Itching, Irritability, Delayed healing, Partial paresis, Muscle stiffness, Alopecia, Obesity and Output- class

Dataset: https://www.kaggle.com/shikhnu/diabetes-risk-prediction-dataset

About Data :

This dataset contains the sign and symptom data of newly diabetic or would be diabetic patient.
Our dataset has data of patients Out of those patients 63.1% males and 36.9% Females
Features of the dataset

The dataset consist of total 16 features and one target variable named class.

1. Age: Age in years ranging from (20years to 65 years)
2. Gender: Male / Female
3. Polyuria: Yes / No
4. Polydipsia: Yes/ No
5. Sudden weight loss: Yes/ No
6. Weakness: Yes/ No
7. Polyphagia: Yes/ No
8. Genital Thrush: Yes/ No
9. Visual blurring: Yes/ No
10. Itching: Yes/ No
11. Irritability: Yes/No
12. Delayed healing: Yes/ No
13. Partial Paresis: Yes/ No
14. Muscle stiffness: yes/ No
15. Alopecia: Yes/ No
16. Obesity: Yes/ No
17. Output- Class: Positive / Negative

Some of the binary (Yes or No) features include:

Polyuria : Polyuria is urine output of > 3 L/day; it must be distinguished from urinary frequency, which is the need to urinate many times during the day or night but in normal or less-than-normal volumes.

Polydipsia : Polydipsia is a medical name for the feeling of extreme thirstiness. Polydipsia is often linked to urinary conditions that cause you to urinate a lot.

Polyphagia : Polyphagia is the medical term used to describe excessive hunger or increased appetite and is one of the 3 main signs of diabetes.

Genital Thrush : Thrush (or candidiasis) is a common condition caused by a type of yeast called Candida.

Visual Blurring : Lack of sharpness of vision with, as a result, the inability to see fine detail.

Irritability : Irritability is the excitatory ability that living organisms have to respond to changes in their environment. The term is used for both the physiological. reaction to stimuli and for the pathological, abnormal or excessive sensitivity to stimuli.

Partial Paresis : Paresis is the medical term for weakened muscle movement. It may also sometimes see it referred to as ―mild paralysis‖ or ―partial paralysis.

Alopecia : Alopecia is the term used for loss of hair, either diffuse or patchy, due to a structural or functional defect in the follicle or to a change in the hair itself.

Target Feature: Likelihood of Diabetes (Positive or Negative)

SAMPLE DATA :

Age	Gender	Polyuria	Polydipsia	sudden weight loss	weakness	Polyphagia	Genital thrush	visual blurring	Itching	Irritability	delayed healing	partial paresis	muscle stiffness	Alopecia	Obesity	class
40	Male	No	Yes	No	Yes	No	No	No	Yes	No	Yes	No	Yes	Yes	Yes	Positive
58	Male	No	No	No	Yes	No	No	Yes	No	No	No	Yes	No	Yes	No	Positive
41	Male	Yes	No	No	Yes	Yes	No	No	Yes	No	Yes	No	Yes	Yes	No	Positive
45	Male	No	No	Yes	Yes	Yes	Yes	No	Yes	No	Yes	No	No	No	No	Positive
60	Male	Yes	Yes	Yes	Yes	Yes	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Positive
55	Male	Yes	Yes	No	Yes	Yes	No	Yes	Yes	No	Yes	No	Yes	Yes	Yes	Positive
57	Male	Yes	Yes	No	Yes	Yes	Yes	No	No	No	Yes	Yes	No	No	No	Positive
66	Male	Yes	Yes	Yes	Yes	No	No	Yes	Yes	Yes	No	Yes	Yes	No	No	Positive
67	Male	Yes	Yes	No	Yes	Yes	Yes	No	Yes	Yes	No	Yes	Yes	No	Yes	Positive
70	Male	No	Yes	Yes	Yes	Yes	No	Yes	Yes	Yes	No	No	No	Yes	No	Positive
44	Male	Yes	Yes	No	Yes	No	Yes	No	No	Yes	Yes	No	Yes	Yes	No	Positive
38	Male	Yes	Yes	No	No	Yes	Yes	No	Yes	No	Yes	No	Yes	No	No	Positive
35	Male	Yes	No	No	No	Yes	Yes	No	No	Yes	Yes	No	No	Yes	No	Positive
61	Male	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No	No	No	No	Yes	Yes	Positive
60	Male	Yes	Yes	No	Yes	Yes	No	Yes	Yes	No	Yes	Yes	No	No	No	Positive
58	Male	Yes	Yes	No	Yes	Yes	No	No	No	No	Yes	Yes	Yes	No	No	Positive
54	Male	Yes	Yes	Yes	Yes	No	Yes	No	No	No	Yes	No	Yes	No	No	Positive
67	Male	No	Yes	No	Yes	Yes	No	Yes	No	Yes	Yes	Yes	Yes	Yes	Yes	Positive
66	Male	Yes	Yes	No	Yes	Yes	No	Yes	No	No	No	Yes	Yes	No	No	Positive
43	Male	Yes	Yes	Yes	Yes	No	Yes	No	No	No	No	No	No	No	No	Positive
62	Male	Yes	Yes	No	Yes	Yes	No	Yes	No	Yes	No	Yes	Yes	No	No	Positive
54	Male	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No	Yes	No	Yes	Yes	No	Positive
39	Male	Yes	No	Yes	No	No	Yes	No	Yes	Yes	No	No	No	Yes	No	Positive
48	Male	No	Yes	Yes	Yes	No	No	Yes	Yes	Yes	Yes	No	No	No	No	Positive
58	Male	Yes	Yes	Yes	Yes	Yes	No	Yes	No	No	Yes	Yes	Yes	No	Yes	Positive
32	Male	No	No	No	No	No	Yes	No	No	Yes	Yes	No	No	No	Yes	Positive
42	Male	No	No	No	Yes	Yes	No	No	No	Yes	No	No	Yes	No	No	Positive
52	Male	Yes	Yes	Yes	Yes	Yes	No	Yes	Yes	No	Yes	Yes	Yes	No	No	Positive
38	Male	No	Yes	No	No	No	Yes	No	No	No	No	No	No	Yes	No	Positive
53	Male	Yes	Yes	Yes	Yes	Yes	No	Yes	No	Yes	No	Yes	No	Yes	No	Positive
57	Male	Yes	Yes	Yes	Yes	Yes	No	Yes	No	No	No	Yes	No	No	No	Positive
41	Male	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No	No	No	No	Yes	Positive
37	Male	No	No	No	Yes	No	No	No	No	No	Yes	No	No	Yes	No	Positive
54	Male	Yes	Yes	Yes	Yes	No	No	Yes	Yes	Yes	Yes	Yes	Yes	No	No	Positive
49	Male	Yes	Yes	No	Yes	No	No	Yes	Yes	No	No	No	No	No	No	Positive
48	Male	Yes	Yes	No	Yes	No	Yes	Yes	Yes	No	No	No	No	No	No	Positive
60	Male	Yes	No	No	No	No	No	Yes	No	No	No	Yes	No	Yes	No	Positive
63	Male	Yes	Yes	Yes	Yes	Yes	No	Yes	No	No	No	No	Yes	Yes	Yes	Positive
35	Male	Yes	No	Yes	No	No	No	No	No	No	No	Yes	No	Yes	No	Positive
30	Female	Yes	No	Yes	Yes	Yes	No	No	No	No	Yes	No	No	No	No	Positive
53	Female	No	Yes	Yes	No	Yes	Yes	Yes	Yes	Yes	Yes	No	Yes	No	No	Positive
50	Female	Yes	Yes	Yes	No	Yes	No	No	No	No	Yes	Yes	No	No	No	Positive
50	Female	No	Yes	No	Yes	Yes	No	Yes	Yes	Yes	Yes	Yes	Yes	No	No	Positive
35	Female	Yes	Yes	Yes	Yes	Yes	No	No	Yes	No	Yes	Yes	Yes	No	No	Positive
40	Female	Yes	Yes	Yes	Yes	No	No	Yes	Yes	No	No	Yes	Yes	No	No	Positive
48	Female	No	No	Yes	Yes	No	No	Yes	Yes	No	Yes	Yes	No	No	No	Positive
60	Female	Yes	Yes	Yes	Yes	Yes	No	Yes	Yes	No	Yes	Yes	Yes	No	Yes	Positive
60	Female	Yes	Yes	Yes	Yes	Yes	No	Yes	Yes	No	No	Yes	Yes	No	No	Positive
35	Female	Yes	Yes	Yes	Yes	Yes	No	Yes	Yes	No	Yes	No	Yes	No	Yes	Positive
46	Female	Yes	No	Yes	Yes	No	No	Yes	No	No	Yes	Yes	Yes	No	No	Positive
36	Female	Yes	Yes	Yes	No	Yes	No	Yes	No	Yes	Yes	Yes	Yes	No	No	Positive
50	Female	Yes	Yes	Yes	Yes	Yes	No	Yes	No	No	No	Yes	No	No	No	Positive
60	Female	No	Yes	Yes	Yes	Yes	Yes	Yes	No	Yes	Yes	Yes	Yes	No	No	Positive
50	Female	No	Yes	Yes	Yes	No	No	Yes	No	No	Yes	No	No	No	No	Positive
51	Female	No	No	No	No	No	No	Yes	No	No	No	Yes	Yes	No	No	Positive
38	Female	No	No	No	No	No	No	No	No	No	No	No	No	No	Yes	Positive
66	Female	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No	Yes	Yes	Yes	No	No	Positive
53	Female	No	No	No	No	Yes	No	Yes	No	No	No	Yes	No	No	No	Positive
59	Female	No	No	No	No	No	No	No	No	No	No	No	No	No	No	Positive
39	Female	Yes	Yes	No	Yes	Yes	No	Yes	No	Yes	Yes	Yes	Yes	No	Yes	Positive
65	Female	Yes	Yes	No	Yes	Yes	No	No	Yes	No	No	Yes	Yes	No	No	Positive

DATA PREPARATION :

- Data preparation stage includes data cleaning and transforming data if needed.
- Various things have to be taken into consideration for data cleaning like:
- Checking for Missing values and removing outliers.
- Handling Zero/Null Values -The zero values have been replaced by the mean of that column.

Our data set contains full of yes or no values, We planned to convert those yes / no values as Numerical values as ;

- ✓ Yes as 1 & No as 0

And for the output class label has the values as positive / negative values, we planned to convert that too as numerical values as follows:

- ✓ Positive as 1 & Negative as 0

Select appropriate attributes for analysis :

The dataset consist of 17 attributes i.e. age, gender, polyuria, Polydipsia, Sudden weight loss,Weakness, Polyphagia, Genital thrush, Visual blurring, Itching, Irritability, Delayed healing, Partial paresis, Muscle stiffness, Alopecia, Obesity.

These attributes are independent attributes and one i.e. Class is the dependent attribute. As all these attributes affect diabetes so we decided to keep all the independent variables for our process.

Data Splitting:

Data was divided into training and testing data into 75:25 ratio. 75percent was training data and 25 percent was testing data.

CHAPTER 5

MODELING :

- ✓ This phase includes application of appropriate model to the data. Data Mining Algorithms were used for modeling.
- ✓ As we have to classify the data into patients having diabetes or not, we are planning to use Decision tree classification, Logistic regression, Random Forest, Naïve Bayes, Support Vector Maching & K-Nearest Neighbour algorithms. These algorithms are good for classifying dependent variables based upon categorized independent variables.
- ✓ We compared these data mining algorithms to find the one which gives the best result based upon overall accuracy and precision.

DECISION TREE:

- ✓ Decision Tree classifier is a supervised and very powerful machine learning algorithm for classification. It involves taking decisions based on prior data. In Decision Tree classifier, we have certain attributes that form various nodes of the tree. The algorithm, in every stage, chooses a node by evaluating the highest information gain among all the attributes.

- ✓ Decision tree builds classification or regression models in the form of a tree structure. It breaks down a dataset into smaller and smaller subsets while at the same time an associated decision tree is incrementally developed. The final result is a tree with decision nodes and leaf nodes. A decision node has two or more branches. Leaf node represents a classification or decision. The topmost decision node in a tree which corresponds to the best predictor called root node. Decision trees can handle both categorical and numerical data.

KNN:

K-nearest neighbour classifier is one of the most simple and non-probabilistic machine learning algorithms. The training dataset is stored and the prediction involves looking for the closest data point from the training set.The primary use of the equation stated above is to calculate the distance between two data points xi and yi where k is the number of dimensions which is determined based on the dataset

$$\text{Euclidean} = \left(\sum_{i=1}^{k} |x_i - y_i|^p\right)^{\frac{1}{p}}$$

- ✓ Calculate the distance between test data and each row of training data. ...
- ✓ Sort the calculated distances in ascending order based on distance values.
- ✓ Get top k rows from the sorted array.
- ✓ Get the most frequent class of these rows.
- ✓ Return the predicted class.

NAIVE BAYES:

Bayesian classifiers are the statistical classifiers. Bayesian classifiers can predict class membership probabilities such as the probability that a given tuple belongs to a particular class. The naive Bayes classifier is one of the most popularly used probabilistic classifiers. It implements Bayes Theorem and discards the order and rules making the independent assumptions among the features. Hence deriving its naive nature. There are various types of Naive Bayes Algorithms, multinomial Naive Bayes, Gaussian Naive Bayes, Bernoulli Naive Bayes, and more. We have used a Gaussian Naive Bayes classifier for our model. Gaussian Naive Bayes is used for high- dimensional data. Naive Bayes algorithms are fast to train and predict data points.

LOGISTIC REGRESSION:

How it works. Logistic Regression measures the relationship between the dependent variable (our label, what we want to predict) and the one or more independent variables (our features), by estimating probabilities using it's underlying logistic function.

Logistic Regression is a binary classification algorithm that follows the equation :

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

RANDOM FOREST:

- ✓ Random forest is a supervised learning algorithm. The "forest" it builds, is an ensemble of decision trees, usually trained with the ―bagging‖ method. The general idea of the bagging method is that a combination of learning models increases the overall result.
- ✓ Put simply: random forest builds multiple decision trees and merges them together to get a more accurate and stable prediction.
- ✓ One big advantage of random forest is that it can be used for both classification and regression problems, which form the majority of current machine learning systems.

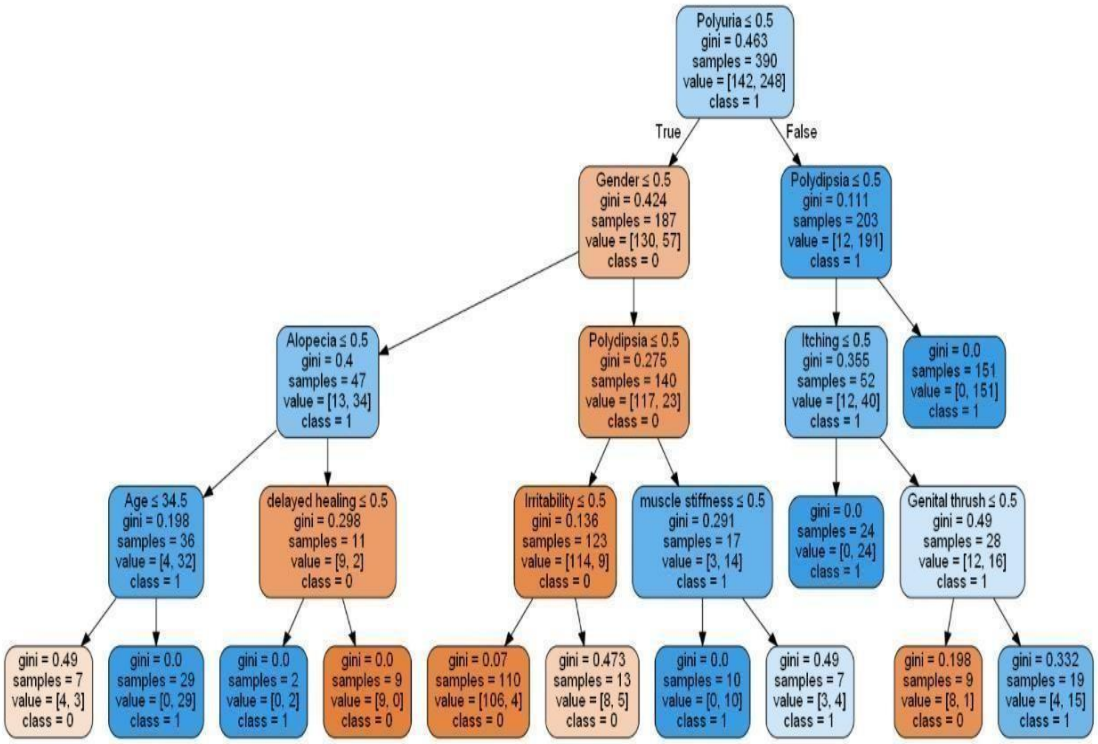
- ✓ A random forest is essentially an ensemble of a number of decision trees. The logic that sticks with random forest is to combine different sets of values from training sets to form decision trees thus reducing the chances of overfitting and misclassification by averaging the results of various decision trees. In the model described in the paper, we have used max-depth of 13 and 100 estimators to classify the data points. On increasing the max-depth, we experienced a decrease in the classification accuracy.

SVM:

A support vector machine is a very important and versatile machine learning algorithm, it is capable of doing linear and nonlinear classification, regression and outlier detection. Support vector machines also known as SVM is another algorithm widely used by machine learning people for both classification as well as regression problems but is widely used for classification tasks. It is preferred over other classification algorithms because it uses less computation and gives notable accuracy. It is good because it gives reliable results even if there is less data.

The distance between data points is measured by the Gaussian kernel:
Here, xi and xj are data points, ‖ xi – xj ‖ denotes Euclidean distance. The choice of kernel functions is dependent on the respective data and specific domain problem. The various values of C and gamma is checked over the validation set. The appropriate value of C for best accuracy on the validation set is selected based on its accuracy.

THE DECISION TREE MODEL :



The Decision Tree obtained gives the result. The depth taken for this tree is 4 and total number of nodes are 23.

Node Description:

Here 0 or <=0.5 represents "No" & 1 or >=0.5 represents "Yes"

- ✓ The root node split in this tree started with Polyuria attribute.
- ✓ No of samples – It is the count of those samples whose Polyuria value is ―No‖
- ✓ Value – It gives the total no of samples for outcome „0‟ and „1‟.
- ✓ Class – Diabetic – ‗1‘ or Non Diabetic – ‗0‘
- ✓ Gini Impurity – This function measures the quality of a split. This factor measures how a randomly chosen element from the set would be incorrectly classified i.e. It is probability of misclassification of a record. I t is used to minimize misclassifications.

Decision Rules :

1. IF Polyuria=No AND Gender=Female AND Alopecia=No AND Age<=34.5 THEN class=negative
2. IF Polyuria=No AND Gender=Female AND Alopecia=No AND Age>=34.5 THEN class=positive
3. IF Polyuria=No AND Gender=Female AND Alopecia=Yes AND Delayed healing=No THEN class=positive
4. IF Polyuria=No AND Gender=Female AND Alopecia=Yes AND Delayed healing=Yes THEN class=negative
5. IF Polyuria=No AND Gender=Male AND Polydipsia=No AND Irritability=No THEN class=negative
6. IF Polyuria=No AND Gender=Male AND Polydipsia=No AND Irritability=Yes THEN class=negative
7. IF Polyuria=No AND Gender=Male AND Polydipsia=yes AND muscle stiffness=No THEN class=positive
8. IF Polyuria=No AND Gender=Male AND Polydipsia=yes AND muscle stiffness=Yes THEN class=positive
9. IF Polyuria=Yes AND Polydipsia=No AND Itching=No THEN class=positive
10. IF Polyuria=Yes AND Polydipsia=No AND Itching=Yes AND Genital thrush=No THEN class=negative
11. IF Polyuria=Yes AND Polydipsia=No AND Itching=Yes AND Genital thrush=Yes THEN class=positive
12. IF Polyuria=Yes AND Polydipsia=Yes THEN class=positive

Feature Importance:

- With the help of decision tree, we were able to figure out which features played animportant role.
- In the following, the highest importance features.

```
feature_importances
```

	importance
Polydipsia	0.226520
Polyuria	0.186996
Gender	0.105599
Age	0.090247
partial paresis	0.052769
sudden weight loss	0.049667
Alopecia	0.044813
Irritability	0.040498
Polyphagia	0.037707
Itching	0.030581
delayed healing	0.028225
weakness	0.023994
visual blurring	0.022873
muscle stiffness	0.022644
Genital thrush	0.020334
Obesity	0.016532

CHAPTER 6

IMPLEMENTATION :

```
import numpy as np
import pandas as pd

import matplotlib.pyplot as plt
import seaborn as sns

%matplotlib inline
import warnings
warnings.filterwarnings("ignore")

df = pd.read_csv('diabetes_dataset.csv')
df.head()
df.tail()
df.info()
df['Gender'] = df['Gender'].apply(str)
df['class'].value_counts(), df['Gender'].value_counts()
 df.columns
df.isna().sum()

df['Gender'] = df['Gender'].map({'Male': 1, 'Female': 0})
df['Polyuria'] = df['Polyuria'].map({'Yes': 1, 'No': 0})
df['Polydipsia'] = df['Polydipsia'].map({'Yes': 1, 'No': 0})
df['sudden weight loss'] = df['sudden weight loss'].map({'Yes': 1, 'No': 0})
df['weakness'] = df['weakness'].map({'Yes': 1, 'No': 0})
df['Polyphagia'] = df['Polyphagia'].map({'Yes': 1, 'No': 0})
df['Genital thrush'] = df['Genital thrush'].map({'Yes': 1, 'No': 0})
df['visual blurring'] = df['visual blurring'].map({'Yes': 1, 'No': 0})
df['Itching'] = df['Itching'].map({'Yes': 1, 'No': 0})
df['Irritability'] = df['Irritability'].map({'Yes': 1, 'No': 0})
```

```
df['delayed healing'] = df['delayed healing'].map({'Yes': 1, 'No': 0})
df['partial paresis'] = df['partial paresis'].map({'Yes': 1, 'No': 0})
df['muscle stiffness'] = df['muscle stiffness'].map({'Yes': 1, 'No': 0})
df['Alopecia'] = df['Alopecia'].map({'Yes': 1, 'No': 0})
df['Obesity'] = df['Obesity'].map({'Yes': 1, 'No': 0})
df['class'] = df['class'].map({'Positive': 1, 'Negative': 0})
df.head()

from sklearn.model_selection import train_test_split

X = df.drop(['class'], axis='columns')
y = df['class']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.25,
random_state=50)

# Creating Random Forest Model
from sklearn.ensemble import RandomForestClassifier
rf = RandomForestClassifier(n_estimators=100, random_state=69)
rf.fit(X_train, y_train)
from sklearn.metrics import confusion_matrix, classification_report, accuracy_score
y_pred = rf.predict(X_test)
rfcm = confusion_matrix(y_test, y_pred)
print(rfcm)
print(classification_report(y_test, y_pred))
print("Random forest accuracy: ",rf.score(X_test,y_test)*100,"%")

feature_importances = pd.DataFrame(rf.feature_importances_,
                                   index = X_train.columns,

columns=['importance']).sort_values('importance',ascending=False)
feature_importances
```

```
#Decision tree model :
import seaborn as sns
from sklearn.tree import DecisionTreeClassifier
from sklearn import metrics
clf = DecisionTreeClassifier(max_depth=4, random_state=0)
tree_= clf.fit(X_train, y_train)
y_pred = clf.predict(X_test)
print("Accuracy:",metrics.accuracy_score(y_test, y_pred))
dtcm = confusion_matrix(y_test, y_pred)
print(dtcm)
print(classification_report(y_test, y_pred))
print("Decision tree accuracy: ",clf.score(X_test,y_test)*100,"%")

#k  nearest neighbour
from sklearn.neighbors import KNeighborsClassifier
knn= KNeighborsClassifier(n_neighbors=10)
knn.fit(X_train, y_train)

y_pred = knn.predict(X_test)
print("Accuracy:",metrics.accuracy_score(y_test, y_pred))
knncm = confusion_matrix(y_test, y_pred)
print(knncm)
print(classification_report(y_test, y_pred))
print("KNN accuracy: ",knn.score(X_test,y_test)*100,"%")

#logistic regression :
from sklearn.linear_model import LogisticRegression
logreg = LogisticRegression(C=10, max_iter=500)
logreg.fit(X_train, y_train)
#logreg.predict([[30,1,1,1,1,0,1,0,1,0,0,0,0,0,0,0]])
y_pred = logreg.predict(X_test)
print("Accuracy:",metrics.accuracy_score(y_test, y_pred))
lrcm= confusion_matrix(y_test, y_pred)
print(lrcm)
```

```
print(classification_report(y_test, y_pred))
print("Logistic Regression accuracy: ",logreg.score(X_test,y_test)*100,"%")

#support vector machine :
from sklearn.svm import SVC
svm = SVC(C=1000)
svm.fit(X_train, y_train)
#svm.predict([[30,1,1,1,1,0,1,0,1,0,0,0,0,0,0,0]])
y_pred = svm.predict(X_test)
print("Accuracy:",metrics.accuracy_score(y_test, y_pred))
svmcm = confusion_matrix(y_test, y_pred)
print(svmcm)
print(classification_report(y_test, y_pred))
print("SVM accuracy: ",svm.score(X_test,y_test)*100,"%")

#naive bayes
from sklearn.naive_bayes import GaussianNB
NBmodel = GaussianNB()
NBmodel.fit(X_train, y_train);
y_pred = NBmodel.predict(X_test)
print("Accuracy:",metrics.accuracy_score(y_test, y_pred))
nbcm = confusion_matrix(y_test, y_pred)
print(nbcm)
print(classification_report(y_test, y_pred))
print("Naive Bayes accuracy: ",NBmodel.score(X_test,y_test)*100,"%")

#confussion matrix :
plt.figure(figsize=(12,12))
cmtitl = ['Logistic Regression','K Nearest Neighbour','Decision Tree','Naïve
Bayes','Random forest','SVM']
cms = [lrcm,knncm,dtcm,nbcm,rfcm,svmcm]
num=0
for x in cms:
    plt.subplot(3,2,num+1,)
```

```
    #plt.xticks(range(2),["No Diabetes","Diabetes"],fontsize=16)
    #plt.yticks(range(2),["No Diabetes","Diabetes"],fontsize=16)
    group_names = ['True Neg','False Pos','False Neg','True Pos']
    group_counts = ['{0:0.0f}'.format(value) for value in x.flatten()]
    group_percentages = ['{0:.2%}'.format(value) for value in
x.flatten()/np.sum(svmcm)]
    labels = [f'{v1}\n{v2}\n{v3}' for v1, v2, v3 in
zip(group_names,group_counts,group_percentages)]
    labels = np.asarray(labels).reshape(2,2)
    sns.heatmap(x, annot=labels, annot_kws={"fontsize":13}, fmt='', cmap='binary')
    plt.title(f'{cmtitl[num]}\n',fontsize=14)
    num += 1
    plt.tight_layout()
plt.show()

#graph :
models = ["K-Nearest Neighbors", "Logistic Regression", "Decision Tree", "Random
Forest", "Support Vector Machine","Naive Bayes"]
best_test_accuracy =
[knn.score(X_test,y_test)*100,logreg.score(X_test,y_test)*100,clf.score(X_test,y_test)
*100,rf.score(X_test,y_test)*100,svm.score(X_test,y_test)*100,NBmodel.score(X_tes
t,y_test)*100]
plt.figure(figsize=(8,6))
plt.bar(models, best_test_accuracy, align='center')
plt.xlabel("Models")
plt.xticks(rotation=30)
plt.ylabel("Best Accuracy")

print(f'Best Accuracy Achieved : {str(max(best_test_accuracy))[:5]}% ')
```

SCREENSHOTS :

Importing essential libraries

In [1]:
```
import numpy as np
import pandas as pd

import matplotlib.pyplot as plt
import seaborn as sns

%matplotlib inline
```

In [2]:
```
import warnings
warnings.filterwarnings("ignore")
```

Reading the dataset

In [3]:
```
df = pd.read_csv('diabetes_dataset.csv')
```

Data Exploration

In [4]:
```
df.head()
```

Out[4]:

	Age	Gender	Polyuria	Polydipsia	sudden weight loss	weakness	Polyphagia	Genital thrush	visual blurring	Itching	Irritability	delayed healing	partial paresis	muscle stiffness	Alopecia	Obesity	cla
0	40	Male	No	Yes	No	Yes	No	No	No	Yes	No	Yes	No	Yes	Yes	Yes	Positi
1	58	Male	No	No	No	Yes	No	No	Yes	No	No	No	Yes	No	Yes	No	Positi
2	41	Male	Yes	No	No	Yes	Yes	No	No	Yes	No	Yes	No	Yes	Yes	No	Positi
3	45	Male	No	No	Yes	Yes	Yes	Yes	No	Yes	No	Yes	No	No	No	No	Positi
4	60	Male	Yes	Yes	Yes	Yes	Yes	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Positi

In [5]:
```
df.tail()
```

Out[5]:

	Age	Gender	Polyuria	Polydipsia	sudden weight loss	weakness	Polyphagia	Genital thrush	visual blurring	Itching	Irritability	delayed healing	partial paresis	muscle stiffness	Alopecia	Obesity	
2194	72	Male	Yes	No	Yes	Yes	Yes	No	No	No	Yes	Yes	Yes	Yes	Yes	No	P
2195	70	Male	Yes	No	Yes	Yes	Yes	Yes	No	No	Yes	Yes	Yes	Yes	Yes	No	P
2196	69	Female	Yes	No	Yes	Yes	Yes	Yes	Yes	Yes	No	Yes	No	No	Yes	No	P
2197	58	Male	No	Yes	Yes	Yes	Yes	No	Yes	Yes	No	No	Yes	No	Yes	Yes	P
2198	47	Male	Yes	No	Yes	Yes	Yes	No	No	No	No	Yes	Yes	Yes	No	No	P

Model Building

In [12]:
```
from sklearn.model_selection import train_test_split

X = df.drop(['class'], axis='columns')
y = df['class']
```

In [13]:
```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.25, random_state=50)
```

Random Forest

In [14]:
```
# Creating Random Forest Model
from sklearn.ensemble import RandomForestClassifier
rf = RandomForestClassifier(n_estimators=100, random_state=69)
rf.fit(X_train, y_train)
from sklearn.metrics import confusion_matrix, classification_report, accuracy_score
y_pred = rf.predict(X_test)
rfcm = confusion_matrix(y_test, y_pred)
print(rfcm)
print(classification_report(y_test, y_pred))
print("Random forest accuracy: ",rf.score(X_test,y_test)*100,"%")
```

```
[[206   0]
 [  0 344]]
              precision    recall  f1-score   support

           0       1.00      1.00      1.00       206
           1       1.00      1.00      1.00       344

    accuracy                           1.00       550
   macro avg       1.00      1.00      1.00       550
weighted avg       1.00      1.00      1.00       550

Random forest accuracy:  100.0 %
```

DATA PREPROCESSING PART

Checking Missing Values

In [10]:
```
df.isna().sum()
```

Out[10]:
```
Age                 0
Gender              0
Polyuria            0
Polydipsia          0
sudden weight loss  0
weakness            0
Polyphagia          0
Genital thrush      0
visual blurring     0
Itching             0
Irritability        0
delayed healing     0
partial paresis     0
muscle stiffness    0
Alopecia            0
Obesity             0
class               0
dtype: int64
```

In [11]:
```
df['Gender'] = df['Gender'].map({'Male': 1, 'Female': 0})
df['Polyuria'] = df['Polyuria'].map({'Yes': 1, 'No': 0})
df['Polydipsia'] = df['Polydipsia'].map({'Yes': 1, 'No': 0})
df['sudden weight loss'] = df['sudden weight loss'].map({'Yes': 1, 'No': 0})
df['weakness'] = df['weakness'].map({'Yes': 1, 'No': 0})
df['Polyphagia'] = df['Polyphagia'].map({'Yes': 1, 'No': 0})
df['Genital thrush'] = df['Genital thrush'].map({'Yes': 1, 'No': 0})
df['visual blurring'] = df['visual blurring'].map({'Yes': 1, 'No': 0})
df['Itching'] = df['Itching'].map({'Yes': 1, 'No': 0})
df['Irritability'] = df['Irritability'].map({'Yes': 1, 'No': 0})
df['delayed healing'] = df['delayed healing'].map({'Yes': 1, 'No': 0})
df['partial paresis'] = df['partial paresis'].map({'Yes': 1, 'No': 0})
df['muscle stiffness'] = df['muscle stiffness'].map({'Yes': 1, 'No': 0})
df['Alopecia'] = df['Alopecia'].map({'Yes': 1, 'No': 0})
df['Obesity'] = df['Obesity'].map({'Yes': 1, 'No': 0})
df['class'] = df['class'].map({'Positive': 1, 'Negative': 0})
df.head()
```

Out[11]:

	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

Knowing the features Importance in the Dataset

In [15]:
```
feature_importances = pd.DataFrame(rf.feature_importances_,
                                   index = X_train.columns,
                                    columns=['importance']).sort_values('importance',ascending=False)
```

In [16]:
```
feature_importances
```

Out[16]:

	importance
Polydipsia	0.226796
Polyuria	0.193949
Gender	0.102319
Age	0.095986
sudden weight loss	0.068168
partial paresis	0.046621
Alopecia	0.039582
Irritability	0.035154
delayed healing	0.030704
Polyphagia	0.029453
Itching	0.028135
visual blurring	0.023759
muscle stiffness	0.022779
Genital thrush	0.021299
weakness	0.019866
Obesity	0.015430

Decision tree

In [17]:
```
import seaborn as sns
from sklearn.tree import DecisionTreeClassifier
from sklearn import metrics
clf = DecisionTreeClassifier(max_depth=4, random_state=0)
tree_= clf.fit(X_train, y_train)
y_pred = clf.predict(X_test)
print("Accuracy:",metrics.accuracy_score(y_test, y_pred))
dtcm = confusion_matrix(y_test, y_pred)
print(dtcm)
print(classification_report(y_test, y_pred))
print("Decision tree accuracy: ",clf.score(X_test,y_test)*100,"%")
```

```
Accuracy: 0.9145454545454546
[[176  30]
 [ 17 327]]
              precision    recall  f1-score   support

           0       0.91      0.85      0.88       206
           1       0.92      0.95      0.93       344

    accuracy                           0.91       550
   macro avg       0.91      0.90      0.91       550
weighted avg       0.91      0.91      0.91       550

Decision tree accuracy:  91.45454545454545 %
```

K Nearest Neighbours

In [18]:
```
from sklearn.neighbors import KNeighborsClassifier
knn= KNeighborsClassifier(n_neighbors=10)
knn.fit(X_train, y_train)

y_pred = knn.predict(X_test)
print("Accuracy:",metrics.accuracy_score(y_test, y_pred))
knncm = confusion_matrix(y_test, y_pred)
print(knncm)
print(classification_report(y_test, y_pred))
print("KNN accuracy: ",knn.score(X_test,y_test)*100,"%")
```

```
Accuracy: 0.9363636363636364
[[200   6]
 [ 29 315]]
              precision    recall  f1-score   support

           0       0.87      0.97      0.92       206
           1       0.98      0.92      0.95       344

    accuracy                           0.94       550
   macro avg       0.93      0.94      0.93       550
weighted avg       0.94      0.94      0.94       550

KNN accuracy:  93.63636363636364 %
```

LogisticRegression

In [19]:
```
from sklearn.linear_model import LogisticRegression
logreg = LogisticRegression(C=10, max_iter=500)
logreg.fit(X_train, y_train)
#Logreg.predict([[30,1,1,1,1,0,1,0,1,0,0,0,0,0,0,0]])
y_pred = logreg.predict(X_test)
print("Accuracy:",metrics.accuracy_score(y_test, y_pred))
lrcm= confusion_matrix(y_test, y_pred)
print(lrcm)
print(classification_report(y_test, y_pred))
print("Logistic Regression accuracy: ",logreg.score(X_test,y_test)*100,"%")
```

```
Accuracy: 0.9218181818181819
[[184  22]
 [ 21 323]]
              precision    recall  f1-score   support

           0       0.90      0.89      0.90       206
           1       0.94      0.94      0.94       344

    accuracy                           0.92       550
   macro avg       0.92      0.92      0.92       550
weighted avg       0.92      0.92      0.92       550

Logistic Regression accuracy:  92.18181818181819 %
```

Support Vector Machine

In [20]:
```
from sklearn.svm import SVC
svm = SVC(C=1000)
svm.fit(X_train, y_train)
#svm.predict([[30,1,1,1,1,0,1,0,1,0,0,0,0,0,0,0]])
y_pred = svm.predict(X_test)
print("Accuracy:",metrics.accuracy_score(y_test, y_pred))
svmcm = confusion_matrix(y_test, y_pred)
print(svmcm)
print(classification_report(y_test, y_pred))
print("SVM accuracy: ",svm.score(X_test,y_test)*100,"%")
```

```
Accuracy: 0.9454545454545454
[[192  14]
 [ 16 328]]
              precision    recall  f1-score   support

           0       0.92      0.93      0.93       206
           1       0.96      0.95      0.96       344

    accuracy                           0.95       550
   macro avg       0.94      0.94      0.94       550
weighted avg       0.95      0.95      0.95       550

SVM accuracy:  94.54545454545455 %
```

Naive Bayes

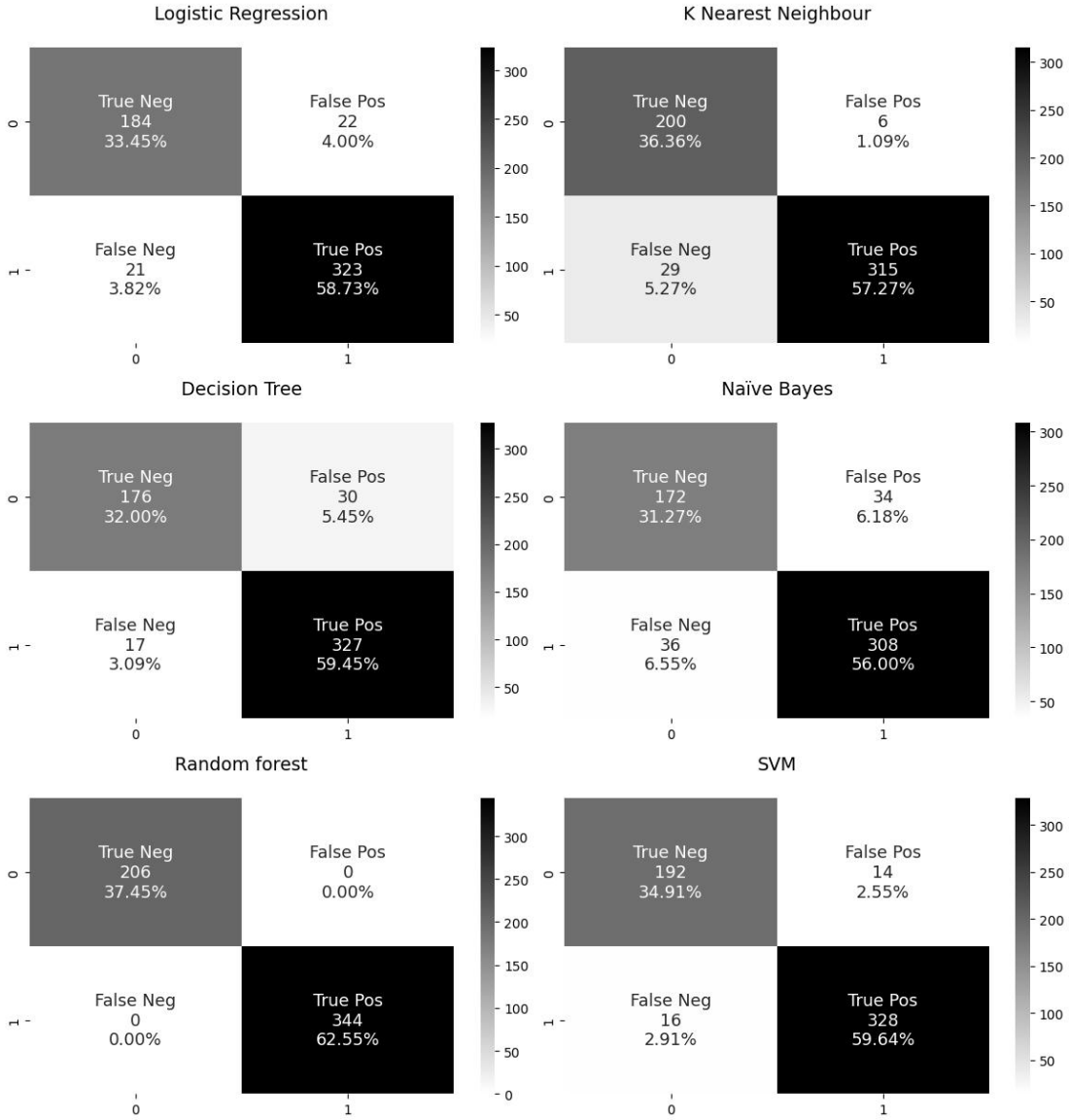
```
21]: from sklearn.naive_bayes import GaussianNB
NBmodel = GaussianNB()
NBmodel.fit(X_train, y_train);
y_pred = NBmodel.predict(X_test)
print("Accuracy:",metrics.accuracy_score(y_test, y_pred))
nbcm = confusion_matrix(y_test, y_pred)
print(nbcm)
print(classification_report(y_test, y_pred))
print("Naive Bayes accuracy: ",NBmodel.score(X_test,y_test)*100,"%")
```

```
Accuracy: 0.8727272727272727
[[172  34]
 [ 36 308]]
              precision    recall  f1-score   support

           0       0.83      0.83      0.83       206
           1       0.90      0.90      0.90       344

    accuracy                           0.87       550
   macro avg       0.86      0.87      0.86       550
weighted avg       0.87      0.87      0.87       550

Naive Bayes accuracy:  87.27272727272727 %
```

CONFUSION MATRIX CODE :

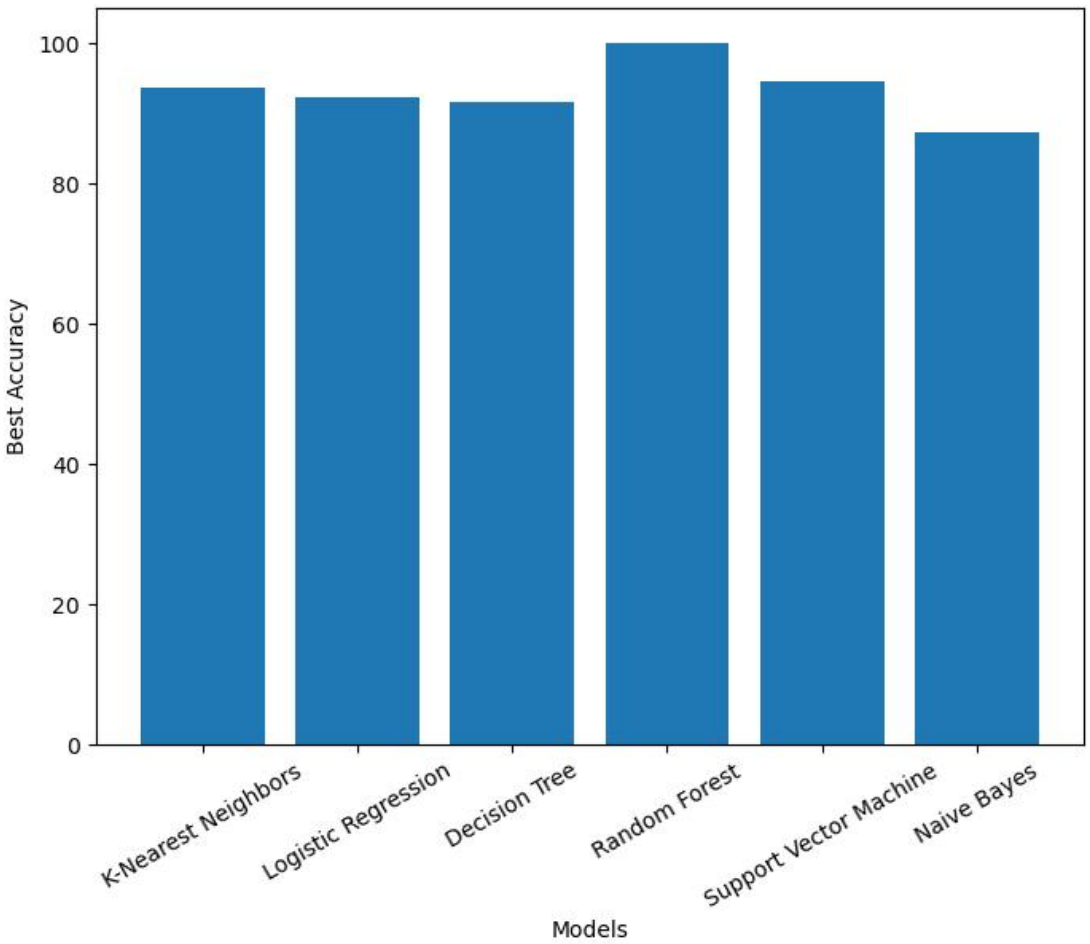
```
In [22]: plt.figure(figsize=(12,12))
cmtitl = ['Logistic Regression','K Nearest Neighbour','Decision Tree','Naïve Bayes','Random forest','SVM']
cms = [lrcm,knncm,dtcm,nbcm,rfcm,svmcm]
num=0
for x in cms:
    plt.subplot(3,2,num+1,)
    #plt.xticks(range(2),["No Diabetes","Diabetes"],fontsize=16)
    #plt.yticks(range(2),["No Diabetes","Diabetes"],fontsize=16)
    group_names = ['True Neg','False Pos','False Neg','True Pos']
    group_counts = ['{0:0.0f}'.format(value) for value in x.flatten()]
    group_percentages = ['{0:.2%}'.format(value) for value in x.flatten()/np.sum(svmcm)]
    labels = [f'{v1}\n{v2}\n{v3}' for v1, v2, v3 in  zip(group_names,group_counts,group_percentages)]
    labels = np.asarray(labels).reshape(2,2)
    sns.heatmap(x, annot=labels, annot_kws={"fontsize":13}, fmt='', cmap='binary')
    plt.title(f'{cmtitl[num]}\n',fontsize=14)
    num += 1
    plt.tight_layout()
plt.show()
```

CONFUSSION MATRIX :



ACCURACY GRAPH :

```
In [23]: models = ["K-Nearest Neighbors", "Logistic Regression", "Decision Tree", "Random Forest", "Support Vector Machine","Naive Bayes"]
         best_test_accuracy = [knn.score(X_test,y_test)*100,logreg.score(X_test,y_test)*100,clf.score(X_test,y_test)*100,rf.score(X_test,y
         plt.figure(figsize=(8,6))
         plt.bar(models, best_test_accuracy, align='center')
         plt.xlabel("Models")
         plt.xticks(rotation=30)
         plt.ylabel("Best Accuracy")

         print(f'Best Accuracy Achieved : {str(max(best_test_accuracy))[:5]}% ')

         Best Accuracy Achieved : 100.0%
```



Give inputs in this order for prediction

Age, Gender, Polyuria, Polydipsia, sudden weight loss, weakness, Polyphagia, Genital thrush, visual blurring, Itching, Irritability, delayed healing, partial paresis, muscle stiffness, Alopecia, Obesity

```
In [24]: rf.predict([[40,1,0,1,1,1,1,0,1,0,0,0,0,0,0,0]])
Out[24]: array([1], dtype=int64)

In [25]: clf.predict([[40,1,0,0,1,0,1,1,1,0,0,1,1,0,1,0]])
Out[25]: array([0], dtype=int64)

In [26]: knn.predict([[40,1,0,0,1,0,1,0,1,0,0,0,0,0,0,0]])
Out[26]: array([0], dtype=int64)
```

CHAPTER 7

CODE FOR USER INTERFACE :

```
import streamlit as st
import pandas as pd
from PIL import Image
from sklearn.metrics import accuracy_score
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import train_test_split

df = pd.read_csv(r"C:\Users\jeeva\Desktop\CAPSTONE\PROJECT
2\IMPLEMENT\diabetes_dataset.csv")

df['Gender'] = df['Gender'].map({'Male': 1, 'Female': 0})
df['Polyuria'] = df['Polyuria'].map({'Yes': 1, 'No': 0})
df['Polydipsia'] = df['Polydipsia'].map({'Yes': 1, 'No': 0})
df['sudden weight loss'] = df['sudden weight loss'].map({'Yes': 1, 'No': 0})
df['weakness'] = df['weakness'].map({'Yes': 1, 'No': 0})
df['Polyphagia'] = df['Polyphagia'].map({'Yes': 1, 'No': 0})
df['Genital thrush'] = df['Genital thrush'].map({'Yes': 1, 'No': 0})
df['visual blurring'] = df['visual blurring'].map({'Yes': 1, 'No': 0})
df['Itching'] = df['Itching'].map({'Yes': 1, 'No': 0})
df['Irritability'] = df['Irritability'].map({'Yes': 1, 'No': 0})
df['delayed healing'] = df['delayed healing'].map({'Yes': 1, 'No': 0})
df['partial paresis'] = df['partial paresis'].map({'Yes': 1, 'No': 0})
df['muscle stiffness'] = df['muscle stiffness'].map({'Yes': 1, 'No': 0})
df['Alopecia'] = df['Alopecia'].map({'Yes': 1, 'No': 0})
df['Obesity'] = df['Obesity'].map({'Yes': 1, 'No': 0})
df['class'] = df['class'].map({'Positive': 1, 'Negative': 0})
```

```
st.title('Diabetes Checkup')
st.subheader('Training Data')
st.write(df.describe())

st.subheader('Visualization')
st.line_chart(df)

x = df.drop(['class'], axis = 'columns')
y = df['class']

x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.25, random_state=50)
st.subheader('Symptoms')
st.subheader('Give 1 for Male and Yes')
st.subheader('Give 0 for Female and No')
def user_report():
    Age = st.slider('Age', 18,100,18)
    Gender = st.selectbox('Gender',["1","0"])
    Polyuria = st.selectbox('Polyuria',["1","0"])
    Polydipsia = st.selectbox('Polydipsia',["1","0"])
    suddenweightloss = st.selectbox('sudden weightloss',["1","0"])
    weakness = st.selectbox('weakness',["1","0"])
    Polyphagia = st.selectbox('Polyphagia',["1","0"])
    Genitalthrush = st.selectbox('Genital thrush',["1","0"])
    visualblurring = st.selectbox('visual blurring',["1","0"])
    Itching = st.selectbox('Itching',["1","0"])
    Irritability = st.selectbox('Irritability',["1","0"])
    delayedhealing = st.selectbox('delayed healing',["1","0"])
    partialparesis = st.selectbox('partial paresis',["1","0"])
    musclestiffness = st.selectbox('muscle stiffness',["1","0"])
    Alopecia = st.selectbox('Alopecia',["1","0"])
    Obesity = st.selectbox('Obesity',["1","0"])
```

```
    user_report ={
        'Age':Age,
        'Gender':Gender,
        'Polyuria':Polyuria,
        'Polydipsia':Polydipsia,
        'sudden weight loss':suddenweightloss,
        'weakness':weakness,
        'Polyphagia':Polyphagia,
        'Genital thrush':Genitalthrush,
        'visual blurring':visualblurring,
        'Itching':Itching,
        'Irritability':Irritability,
        'delayed healing':delayedhealing,
        'partial paresis':partialparesis,
        'muscle stiffness':musclestiffness,
        'Alopecia':Alopecia,
        'Obesity':Obesity
    }
    report_data =pd.DataFrame(user_report, index=[0])
    return report_data
user_data = user_report()
rf = RandomForestClassifier()
rf.fit(x_train, y_train)

st.subheader('Accuracy: ')
st.write(str(accuracy_score(y_test, rf.predict(x_test))*100)+'%')
user_result = rf.predict(user_data)
st.subheader('Your Report: ')
output = ''
if user_result[0]==1:
    output = 'Positive'
else:
    output = 'Negative'
st.write(output)
```

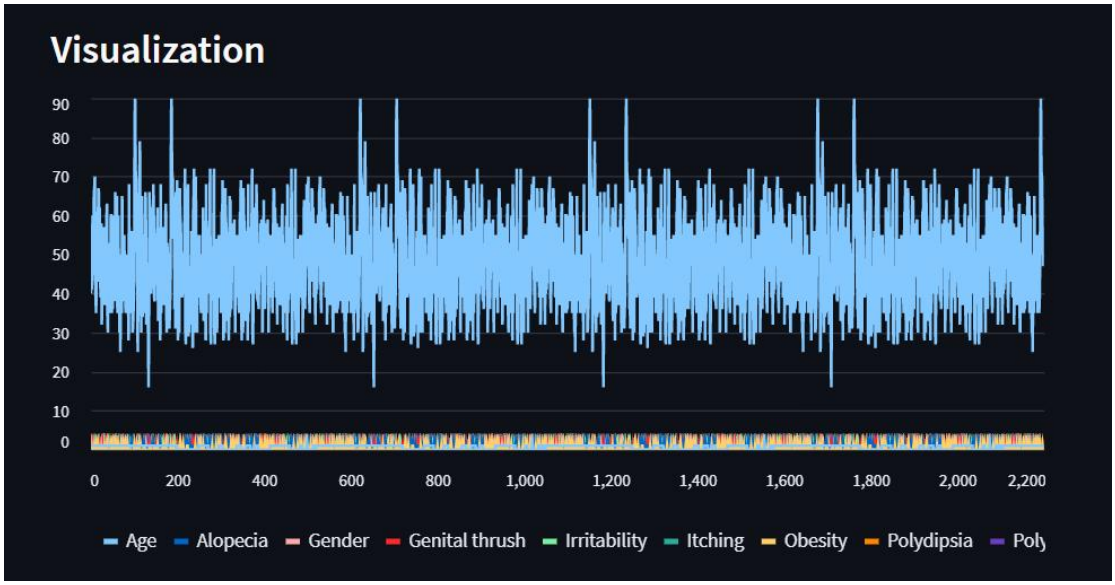
USER INTERFACE :

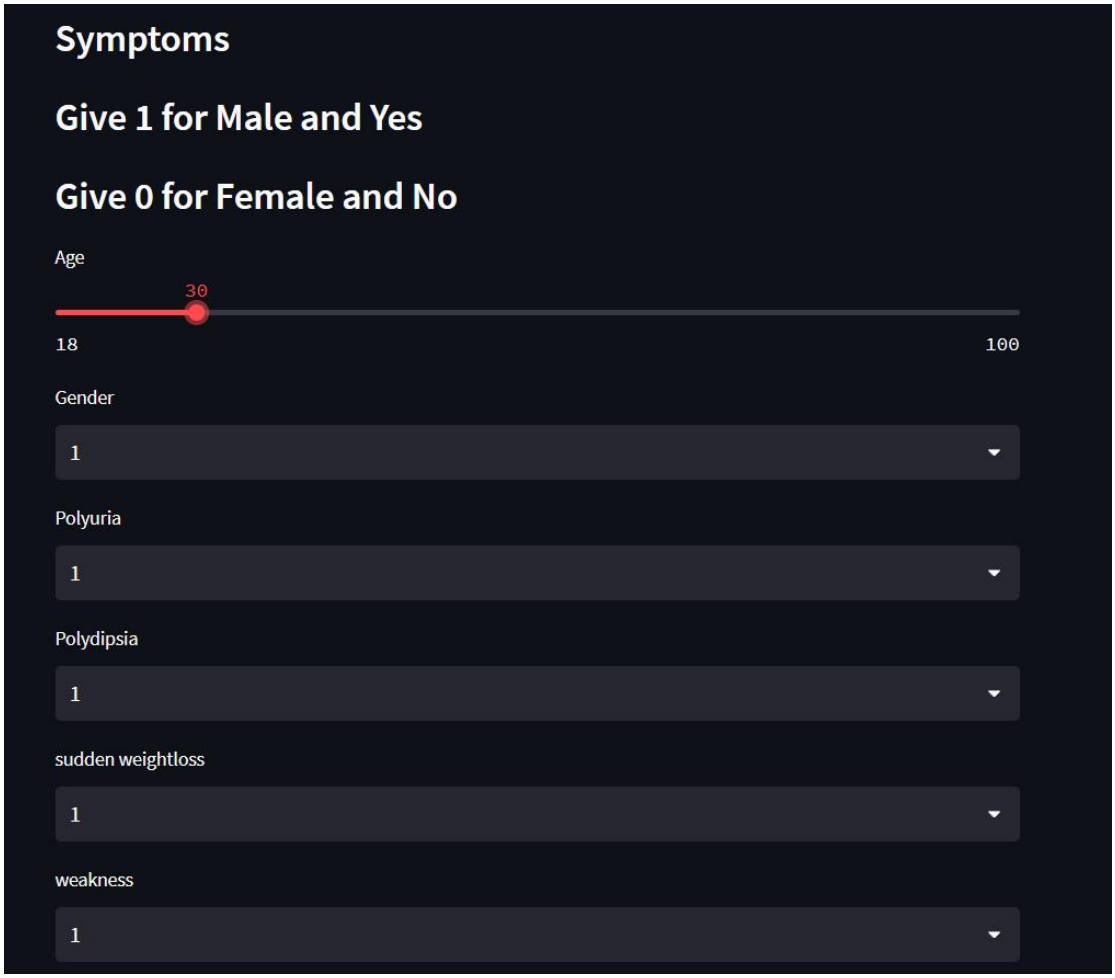
```python
import streamlit as st
import pandas as pd
from PIL import Image
from sklearn.metrics import accuracy_score
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import train_test_split

df = pd.read_csv(r"C:\Users\jeeva\Desktop\CAPSTONE\PROJECT 2\IMPLEMENT\diabetes_dataset.csv")


df['Gender'] = df['Gender'].map({'Male': 1, 'Female': 0})
df['Polyuria'] = df['Polyuria'].map({'Yes': 1, 'No': 0})
df['Polydipsia'] = df['Polydipsia'].map({'Yes': 1, 'No': 0})
df['sudden weight loss'] = df['sudden weight loss'].map({'Yes': 1, 'No': 0})
df['weakness'] = df['weakness'].map({'Yes': 1, 'No': 0})
df['Polyphagia'] = df['Polyphagia'].map({'Yes': 1, 'No': 0})
df['Genital thrush'] = df['Genital thrush'].map({'Yes': 1, 'No': 0})
df['visual blurring'] = df['visual blurring'].map({'Yes': 1, 'No': 0})
df['Itching'] = df['Itching'].map({'Yes': 1, 'No': 0})
df['Irritability'] = df['Irritability'].map({'Yes': 1, 'No': 0})
df['delayed healing'] = df['delayed healing'].map({'Yes': 1, 'No': 0})
df['partial paresis'] = df['partial paresis'].map({'Yes': 1, 'No': 0})
df['muscle stiffness'] = df['muscle stiffness'].map({'Yes': 1, 'No': 0})
df['Alopecia'] = df['Alopecia'].map({'Yes': 1, 'No': 0})
df['Obesity'] = df['Obesity'].map({'Yes': 1, 'No': 0})
df['class'] = df['class'].map({'Positive': 1, 'Negative': 0})



st.title('Diabetes Checkup')
st.subheader('Training Data')
st.write(df.describe())
```

Diabetes Checkup

Training Data

	Age	Gender	Polyuria	Polydipsia	sudden weight loss	weakness	Polyphagia	Genital thrush
count	2,199	2,199	2,199	2,199	2,199	2,199	2,199	2,199
mean	48.1478	0.623	0.5039	0.4584	0.4256	0.5944	0.4625	0.2237
std	12.145	0.4847	0.5001	0.4984	0.4946	0.4911	0.4987	0.4168
min	16	0	0	0	0	0	0	0
25%	39	0	0	0	0	0	0	0
50%	48	1	1	0	0	1	0	0
75%	57	1	1	1	1	1	1	0
max	90	1	1	1	1	1	1	1


Visualization
90
80
70
60
50
40
30
20
10
0
0
200
400
600
800
1,000
1,200
1,400
1,600
1,800
2,000
2,200
Age
Alopecia
Gender
Genital thrush
Irritability
Itching
Obesity
Polydipsia
Poly


Symptoms
Give 1 for Male and Yes
Give 0 for Female and No
Age
30
18
100
Gender
1
Polyuria
1
Polydipsia
1
sudden weightloss
1
weakness
1

Polyphagia

1

Genital thrush

1

visual blurring

1

Itching

1

Irritability

1

delayed healing

1

partial paresis

1

muscle stiffness

1

Alopecia

1

Obesity

1

Accuracy:

100.0%

Your Report:

Positive

CHAPTER 8

8.1 CONCLUSION

- ✓ The Random forest achieved the highest accuracy. Different options were taken into consideration to improve the accuracy. So finally by removing outliers, categorizing data, we were able to achieve desired accuracy.

- ✓ During this process we figured out few attributes that played an important role . Out of 16 attributes Polyuria, Polydipsia, Gender and age were the important ones. As per our results and data the other factors like Obesity, Genital thrush, muscle stiffness and etc., had negligible effect in determining diabetes.

- ✓ We even compared our model with Other Models and inferred that Random Forest is the best among the others.

8.2 REFERENCES :

1. Cut Fiarni, Evasaria M. Sipayung, Siti Maemunah,Analysis and Prediction of Diabetes Complication Disease using Data Mining Algorithm,Procedia Computer Science,Volume 161,2019,

2. Bala Manoj Kumar P, Srinivasa Perumal R, Nadesh R K, Arivuselvan K,Type 2: Diabetes mellitus prediction using Deep Neural Networks classifier,International Journal of Cognitive Computing in Engineering,
Volume 1,2020,

3. Saloni Kumari, Deepika Kumar, Mamta Mittal,An ensemble approach for classification and prediction of diabetes mellitus using soft voting classifier,International Journal of Cognitive Computing in Engineering,
Volume 2,2021,

4. Chollette C. Olisah, Lyndon Smith, Melvyn Smith,Diabetes mellitus prediction and diagnosis from a data preprocessing and machine learning perspective,Computer Methods and Programs in Biomedicine,Volume 220,
2022,

5. Mary Foong Fong Chong, John E Connolly, Yap Seng Chong, Johan G Eriksson, Mengling Feng, Neerja Karnani,Population-centric risk prediction modeling for gestational diabetes mellitus: A machine learning approach,Diabetes Research and Clinical Practice,Volume 185,2022,

6. Shahid Mohammad Ganie, Majid Bashir Malik,An ensemble Machine Learning approach for predicting Type-II diabetes mellitus based on lifestyle indicators,Healthcare Analytics,Volume 2,2022,

7. Victor Chang, Meghana Ashok Ganatra, Karl Hall, Lewis Golightly, Qianwen Ariel Xu,An assessment of machine learning models and algorithms for early prediction and diagnosis of diabetes using health indicators,
Healthcare Analytics,Volume 2,2022,

8. Vandana Rawat, Shivangi Joshi, Shikhar Gupta, Devesh Pratap Singh, Neelam Singh,Machine learning algorithms for early diagnosis of diabetes mellitus: A comparative study,Materials Today: Proceedings,
Volume 56, Part 1,2022,

9. Jyotismita Chaki, S. Thillai Ganesh, S.K Cidham, S. Ananda Theertan,Machine learning and artificial intelligence based Diabetes Mellitus detection and self-management: A systematic review,Journal of King Saud University - Computer and Information Sciences,Volume 34, Issue 6, Part B,2022,

10. Rashi Rastogi, Mamta Bansal,Diabetes prediction model using data mining techniques,Measurement: Sensors,
2022,