Diabetes Disease Prediction using ML.

Name : Sourav salunkhe

Abstract:

Using Machine learning, our project proposes disease prediction system. For small issues, the users need to go in person to the hospital for check-up that is longer intense. Also handling the telecom entails appointments is kind of agitated. Such a tangle may be solved by Disease prediction application by giving correct steerage relating to healthy living. Over the past decade, the utilization of the particular disease prediction tools alongside the regarding health has been magnified because of a range of diseases and fewer doctor-patient magnitude relation. Thus, during this system, we have a tendency to area unit concentrating on providing immediate and correct disease prediction to the users concerning the symptoms they enter alongside the severity of disease expected. Best appropriate rule and doctor consultation are given during this project. For prediction of diseases, totally different machine learning algorithms area unit wont to guarantee fast and correct predictions. In one channel, the symptoms entered are crosschecked with the information. Further, can be preserved within the information if the symptom is new that its primary work is and therefore the different channel will offer severity of disease expected. A web/android application is deployed for user for straightforward moveableness, configuring and having the ability to access remotely wherever doctors cannot reach simply. Usually users don't seem to be privy to all the treatment relating to the actual disease, this project additionally appearance forward to providing medication and drug consultation of disease expected. Therefore, this arrangement helps in easier health management.

Introduction of the Project Diabetes Diseases Prediction Using ML :

In this digital world, data is an asset, and enormous data was generated in all the fields. Data in the healthcare industry consists of all the information related to patients. Here a general architecture has been proposed for predicting the disease in the healthcare industry. we are concentrating on providing immediate and accurate disease predictions to the users about the symptoms they enter along with the disease predicted. So, we are proposing a system which used to predict Diabetes diseases by using ML. In this system, we are going to analysis Diabetes disease analysis. To implement Diabetic disease prediction systems we are going to use machine learning algorithms, and Streamlit . Python pickling is used to save the behavior of the model. The importance of this system analysis is that while analyzing the diseases all the parameters which cause the disease is included so it is possible to detect the disease efficiently and more accurately. The final model's behavior will be saved as a python pickle file.

Diabetes is a chronic disease that occurs when the body is unable to produce or use insulin effectively, resulting in high blood sugar levels. Diabetes can lead to serious health complications such as heart disease, stroke, kidney disease, blindness, and amputations. According to the International Diabetes Federation, 463 million adults worldwide have diabetes, and this number is expected to increase to 700 million by 2045. Early detection and intervention are crucial to prevent the development of complications and improve patient outcomes. Machine learning (ML) techniques have shown promise in accurately predicting diabetes based on patients' clinical fact.

Existing System of Diabetes Diseases Prediction Using ML:

In the existing system the exams are done only manually but in proposed system we have to computerize the exams using this application.

- Lack of security of data.
- More man power.
- Time consuming.
- Consumes large volume of pare work.
- Needs manual calculations.
- No direct role for the higher officials.

Existing Method for Diabetes Diseases Prediction Using ML:

There are several existing methods for diabetes disease prediction using ML. Some of the commonly used methods are:

1) **Logistic Regression:** Logistic regression is a widely used ML algorithm for diabetes prediction. It is a binary classification algorithm that uses the sigmoid function to predict the probability of a patient having diabetes.

2) **Decision Tree**: Decision tree is a non-parametric supervised learning algorithm that can be used for classification and regression tasks. It can be used to predict the likelihood of a patient developing diabetes based on their clinical factors.

3) **Random Forest:** Random Forest is an ensemble learning algorithm that combines multiple decision trees to improve the accuracy and reduce overfitting. It can be used for diabetes prediction based on patients' clinical factors.

4) **Support Vector Machine (SVM):** SVM is a binary classification algorithm that works by finding the best separating hyperplane between two classes. It can be used for diabetes prediction based on patients' clinical factors.

5) **Neural Networks:** Neural networks are a class of deep learning algorithms that can be used for classification and regression tasks. They can be used for diabetes prediction based on patients' clinical factors.

6) **Gradient Boosting:** Gradient boosting is an ensemble learning algorithm that combines multiple weak learners to improve the accuracy and reduce overfitting. It can be used for diabetes prediction based on patients' clinical factors.

Each of these methods has its advantages and limitations, and the choice of method depends on the dataset and the specific problem being addressed

Operating Environment - Hardware and Software :

1. Server side requirement :

1.1 Hardware Requirements:

Processor :Intel® Core™ i3-1005G1 CPU @ 1.20GHz
RAM : 8.00GB
HDD : 1TB

1.2 Software Requirements:

Operating System : Windows 10
Database Support: CSV File, SAV file
Front End: Python , ML Alogoritham , Streamlit
Libraries: pandas 1.5 v, Numpy 1.22.0 , Scikit learn 0.20, seaborn 0.12.2 , matplotlib 3.6.3
Software Development Tool : VS CODE 1.69(June 2022)

2. Client side requirement:

2.1 Hardware Requirements:

Processor **:**Intel® Core™ i3-1005G1 CPU @ 1.20GH
RAM **:** 8.00GB
HDD : 1TB

2.2 Software Requirements:

Operating System : Windows 10
Browser : Google Chrome Version 99.0.4844.84 (Official Build) (64-bit)

Proposed Method

In diabetes disease prediction, it is possible to predict more than one disease at a time. So the user doesn't need to traverse different sites in order to predict the diseases. To implement diabetes disease analyses we are going to use machine learning algorithms and Streamlit. When the user is accessing this API, the user has to send the parameters of the disease along with the disease name. Streamlit will invoke the corresponding model and returns the status of the patient.

This paper proposing such a system which will flaunt a simple and elegant User Interface and also be time efficient. In order to make it less time consuming we are aiming at a more specific questionnaire which will be followed by the system. Our aim with this system is to be the connecting bridge between doctors and patients. The main feature will be the machine learning, in which we will be using algorithms such as Machine Learning, Linear Regression algorithm, SVM, Logistic Regression Algorithm, Streamlit, Python and Support Vector Machine, which will help us in getting accurate predictions and Also, will find which algorithm gives a faster and efficient result by comparatively-comparing. Another feature that our system will comprise of is Doctor's Consultation. After delivering the results, our system will also suggest the user to get a doctors consultation on this report.

Proposed Method and Architecture for Diabetes Disease Prediction using ML:

1) Data Collection: Collect the dataset of patients with diabetes and their clinical factors. The dataset should be cleaned and preprocessed to remove any missing values and normalize the features.

2) Feature Selection: Select the most important clinical factors that are highly correlated with diabetes. This can be done using statistical methods such as correlation analysis or machine learning methods such as Recursive Feature Elimination (RFE).

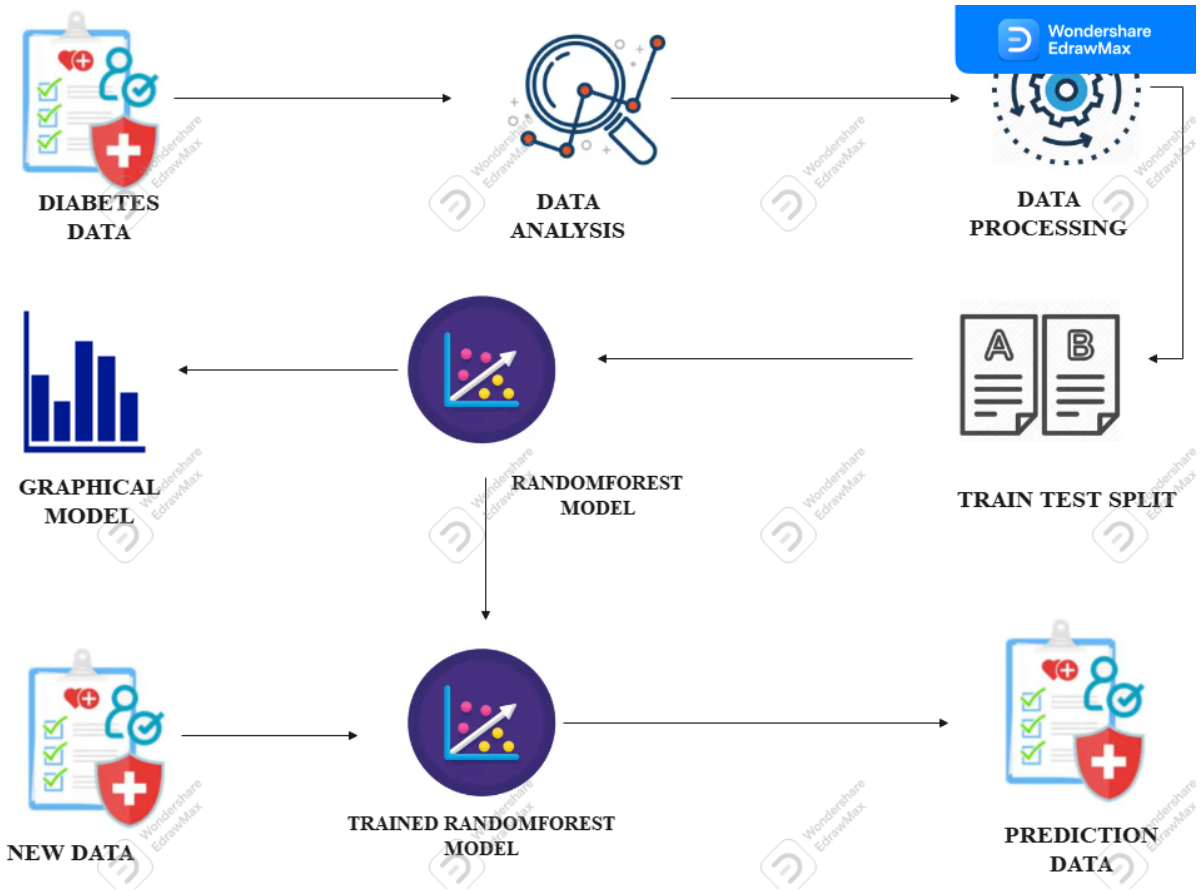
3) Data Split: Split the dataset into training and testing sets. The training set will be used to train the ML model, and the testing set will be used to evaluate the model's performance.

4) Model Training: Train different ML algorithms, such as Logistic Regression, Decision Tree, Random Forest, SVM, and Gradient Boosting, on the training set. Hyperparameter tuning can be performed to optimize the performance of the models.

5) Model Evaluation: Evaluate the performance of the models on the testing set using various metrics such as accuracy, precision, recall, and F1-score. Select the best performing model for diabetes prediction.

6) Model Deployment: Deploy the selected model in a web application or mobile app for real-time diabetes prediction. The user can input their clinical factors, and the model will predict the likelihood of them developing diabetes.

Work flow / Architecture :



Architecture:

The proposed architecture for diabetes disease prediction using ML is as follows:

Data Collection Layer: This layer collects the dataset of patients with diabetes and their clinical factors. The dataset is preprocessed to remove any missing values and normalize the features.

Feature Selection Layer: This layer selects the most important clinical factors that are highly correlated with diabetes using statistical or machine learning methods.

Model Training Layer: This layer trains different ML algorithms, such as Logistic Regression, Decision Tree, Random Forest, SVM, and Gradient Boosting, on the training set. Hyperparameter tuning can be performed to optimize the performance of the models.

Model Evaluation Layer: This layer evaluates the performance of the models on the testing set using various metrics such as accuracy, precision, recall, and F1-score. The best performing model is selected for diabetes prediction.

Model Deployment Layer: This layer deploys the selected model in a web application or mobile app for real-time diabetes prediction. The user can input their clinical factors, and the model will predict the likelihood of them developing diabetes.

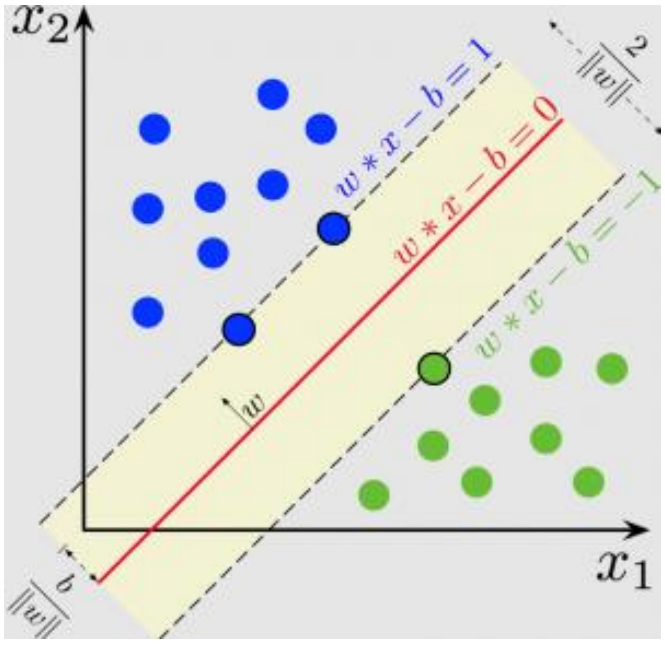
Methodology:

We used the following ML algorithms to develop a predictive model for diabetes:

1) Logistic Regression
2) Decision Tree
3) Random Forest
4) Support Vector Machine (SVM)
5) Gradient Boosting

We split the dataset into training and testing sets to evaluate the performance of the models. We trained the models on the training set and tested their accuracy on the testing set. We also used cross-validation to ensure that the models were not overfitting to the training data.

1) SVC (Support Vector Classifier) :



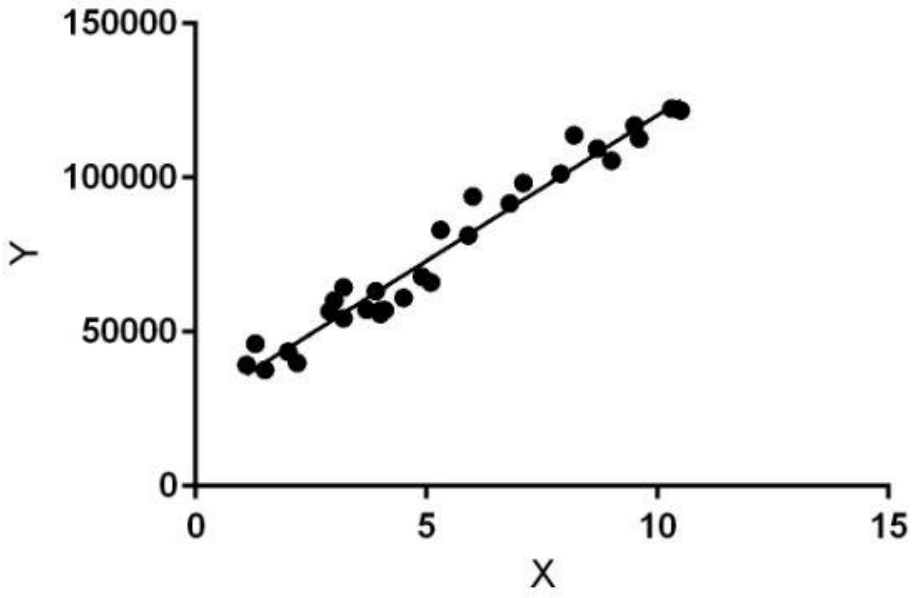
Support Vector Classifier, is **a supervised machine learning algorithm typically used for classification tasks**. SVC works by mapping data points to a high-

dimensional space and then finding the optimal hyperplane that divides the data into two classes.

Support Vector Machine or SVM is one among the foremost standard supervised Learning algorithms, that is employed for Classification further as Regression issues. However, primarily, it's used for Classification issues in Machine Learning. The goal of the SVM rule is to make the simplest line or call boundary that may segregate n-dimensional house into categories in order that we will simply place the new datum within the correct class within the future. This best call boundary is termed a hyperplane. SVM chooses the acute points/vectors that facilitate in making the hyperplane. These extreme cases square measure known as support vectors, and therefore rule is termed as Support Vector Machine.

Accuracy score of the training data : 0.7833876221498371

2) Linear Regression :



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.

$$y = \theta_1 + \theta_2.x$$

Different regression models differ based on – the kind of relationship between dependent and independent variables they are considering, and the number of independent variables getting used. There are many names for a regression's dependent variable. It may be called an outcome variable, criterion variable, endogenous variable, or regressand. The independent variables can be called exogenous variables, predictor variables, or regressors.

Linear regression performs the task to predict a dependent variable value (y) based on a given independent variable (x). So, this regression technique finds out a linear relationship between x (input) and y(output). Hence, the name is Linear Regression. In the figure above, X (input) is the work experience and Y (output) is the salary of a person. The regression line is the best fit line for our model. **Hypothesis function for Linear Regression :**

While training the model we are given : **x:** input training data (univariate – one input variable(parameter)) **y:** labels to data (Supervised learning) When training the model – it fits the best line to predict the value of y for a given value of x. The model gets the best regression fit line by finding the best θ_1 and θ_2 values. **θ_1:** intercept **θ_2:** coefficient of x Once we find the best θ_1 and θ_2 values, we get the best fit line. So when we are finally using our model for prediction, it will predict the value of y for the input value of x.

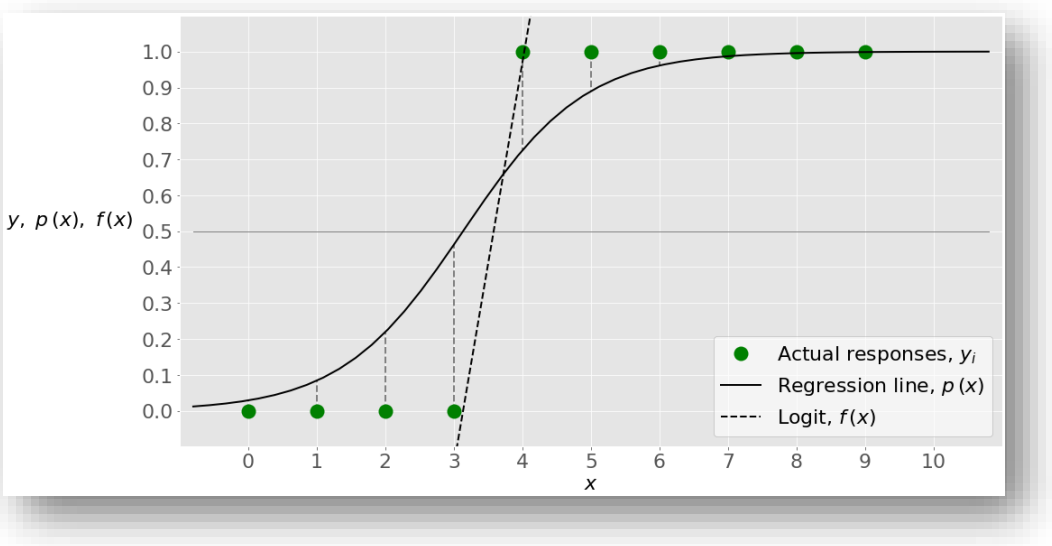
Cost Function (J): By achieving the best-fit regression line, the model aims to predict y value such that the error difference between predicted value and true value is minimum. So, it is very important to update the θ_1 and θ_2 values, to reach the best

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

value that minimize the error between predicted y value (pred) and true y value (y).

Accuracy score of the training data : 0.8789546221498371

3) Logistic regression :



Logistic regression is **a data analysis technique that uses mathematics to find the relationships between two data factors**. It then uses this relationship to predict the value of one of those factors based on the other. The prediction usually has a finite number of outcomes, like yes or no.

Logistic regression aims to solve classification problems. It does this by predicting categorical outcomes, unlike linear regression that predicts a continuous outcome.

In the simplest case there are two outcomes, which is called binomial, an example of which is predicting if a tumor is malignant or benign. Other cases have more than two outcomes to classify, in this case it is called multinomial.

Classification Performance

Binary classification has four possible types of results:

1. **True negatives:** correctly predicted negatives (zeros)
2. **True positives:** correctly predicted positives (ones)
3. **False negatives:** incorrectly predicted negatives (zeros)
4. **False positives:** incorrectly predicted positives (ones)

You usually evaluate the performance of your classifier by comparing the actual and predicted outputsand counting the correct and incorrect predictions.

The most straightforward indicator of **classification accuracy** is the ratio of the number of correct predictions to the total number of predictions (or observations). Other indicators of binary classifiers include the following:

The positive predictive value is the ratio of the number of true positives to the sum of the numbers of true and false positives.

- **The negative predictive value** is the ratio of the number of true negatives to the sum of the numbers of true and false negatives.
- **The sensitivity** (also known as recall or true positive rate) is the ratio of the number of true positives to the number of actual positives.
- **The specificity** (or true negative rate) is the ratio of the number of true negatives to the number of actual negatives.

The most suitable indicator depends on the problem of interest. In this tutorial, you'll use the most straightforward form of classification accuracy.

Accuracy on Test data : 0.819672131147541

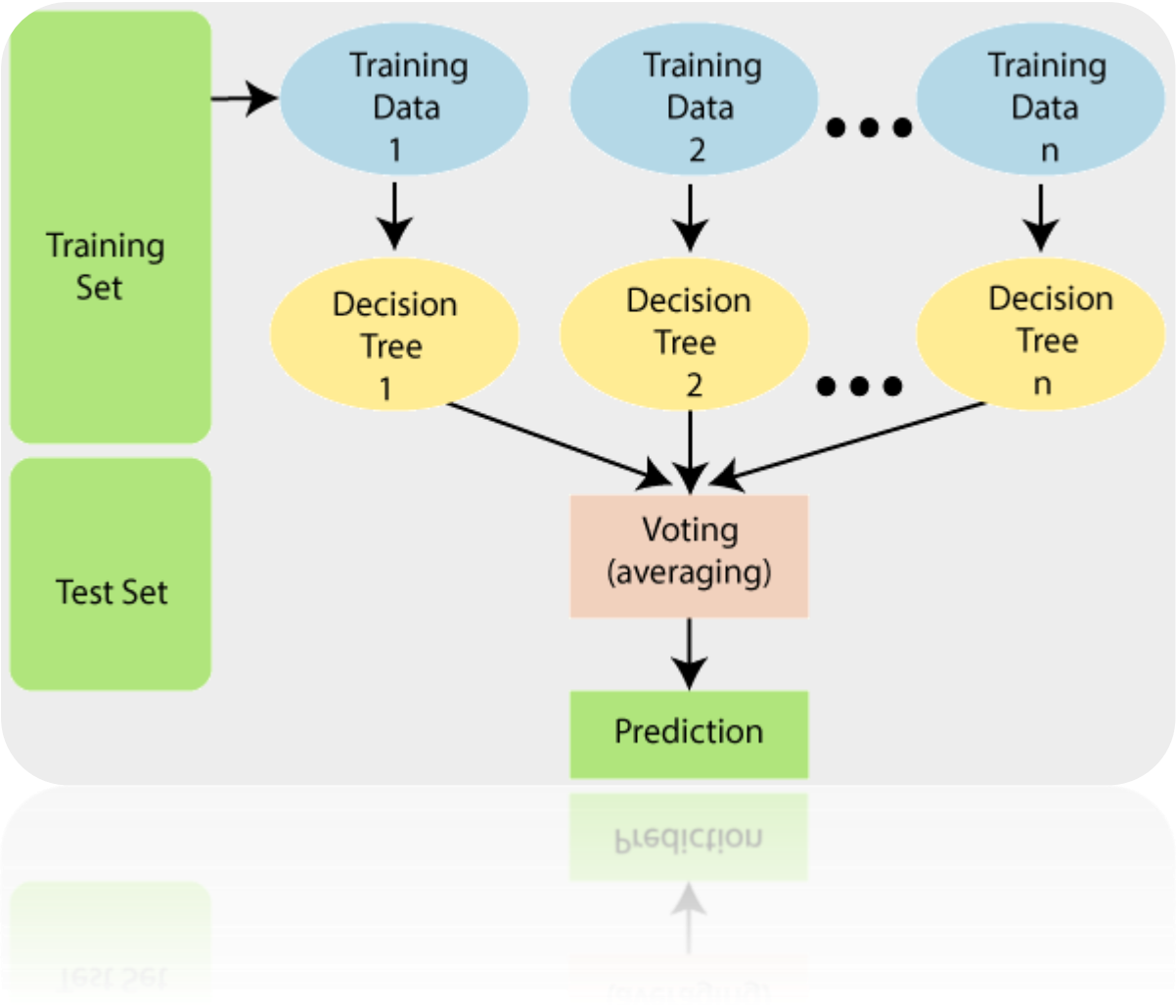
4) RANDOM FOREST ALGORITHM

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

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

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

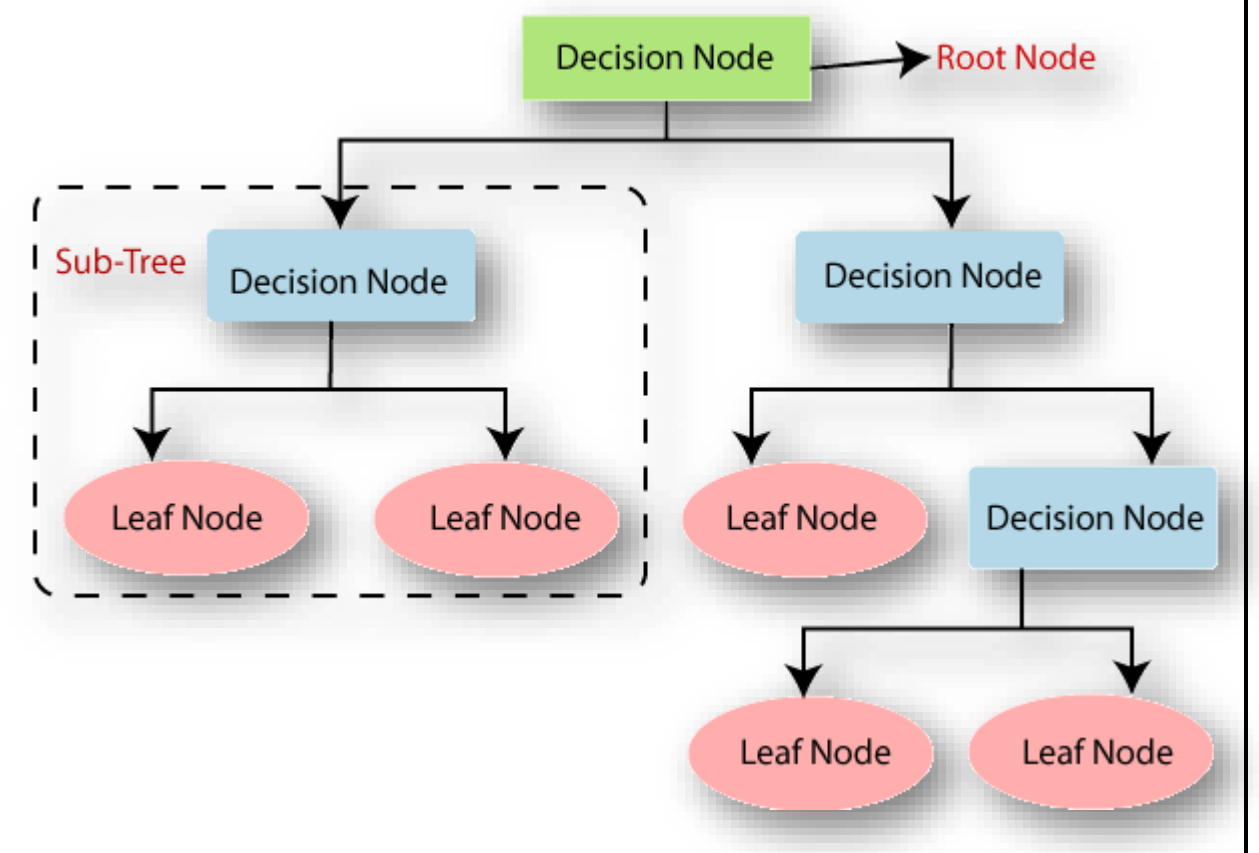
The below diagram explains the working of the Random Forest algorithm:



5) DECISION TREE CLASSIFICATION ALGORITHM

- Decision Tree is a **Supervised learning technique** that can be used for both classification and Regression problems, but mostly it is preferred for solving Classification problems. It is a tree-structured classifier, where **internal nodes represent the features of a dataset, branches represent the decision rules** and **each leaf node represents the outcome.**
- In a Decision tree, there are two nodes, which are the **Decision Node** and **Leaf Node.** Decision nodes are used to make any decision and have multiple branches, whereas Leaf nodes are the output of those decisions and do not contain any further branches.
- The decisions or the test are performed on the basis of features of the given dataset.
- ***It is a graphical representation for getting all the possible solutions to a problem/decision based on given conditions.***

- It is called a decision tree because, similar to a tree, it starts with the root node, which expands on further branches and constructs a tree-like structure.
- In order to build a tree, we use the **CART algorithm,** which stands for **Classification and Regression Tree algorithm.**
- A decision tree simply asks a question, and based on the answer (Yes/No), it further split the tree into subtrees.
- Below diagram explains the general structure of a decision tree:



Implementation of Diabetes Disease Prediction using ML:

1) **Data Collection:** Collect the dataset of patients with diabetes and their clinical factors. Use a dataset such as the Pima Indians Diabetes Database or the National Health and Nutrition Examination Survey (NHANES) dataset.

```
# loading the diabetes dataset to a pandas DataFrame
d_data = pd.read_csv('D:\Exposys Data Lab\Diabetes Prediction\Dataset\diabetes.csv')
```
✓ 0.1s

d:\Exposys Data Lab\Diabetes Prediction\colab file\dp.ipynb > d_data (768, 9)

	index	Pregn...	Glucose	Blood...	SkinT...	Insulin	BMI	Diabe...	Age	Outco...
0	0	6	148	72	35	0	33.6	0.627	50	1
1	1	1	85	66	29	0	26.6	0.351	31	0
2	2	8	183	64	0	0	23.3	0.672	32	1
3	3	1	89	66	23	94	28.1	0.167	21	0
4	4	0	137	40	35	168	43.1	2.288	33	1
5	5	5	116	74	0	0	25.6	0.201	30	0
6	6	3	78	50	32	88	31	0.248	26	1
7	7	10	115	0	0	0	35.3	0.134	29	0
8	8	2	197	70	45	543	30.5	0.158	53	1
9	9	8	125	96	0	0	0	0.232	54	1
10	10	4	110	92	0	0	37.6	0.191	30	0

2) **Data Preprocessing:** Clean and preprocess the dataset by removing any missing values and normalizing the features.

```
d_data.isnull().sum()
```
✓ 0.1s

```
Pregnancies                 0
Glucose                     0
BloodPressure               0
SkinThickness               0
Insulin                     0
BMI                         0
DiabetesPedigreeFunction    0
Age                         0
Outcome                     0
dtype: int64
```

```
d_data.isnull().sum().sum()
```
✓ 0.2s

```
0
```

so , we have no missing value

3) **Feature Selection:** Select the most important clinical factors that are highly correlated with diabetes using statistical or machine learning methods. For example, use correlation analysis or Recursive Feature Elimination (RFE) to select the top features.

4) **Data Split:** Split the dataset into training and testing sets. Use a 70:30 or 80:20 split for training and testing, respectively.

5) **Model Selection:** Select an appropriate ML algorithm for diabetes prediction based on the problem's characteristics and data. Common ML algorithms used for diabetes prediction include Logistic Regression, Decision Tree, Random Forest, SVM, and Gradient Boosting.

6) **Model Training:** Train the selected ML algorithm on the training set. Perform hyperparameter tuning to optimize the model's performance.

7) **Model Evaluation:** Evaluate the performance of the model on the testing set using various metrics such as accuracy, precision, recall, and F1-score. Adjust the model if necessary and re-evaluate its performance.

8) **Model Deployment:** Deploy the trained model in a web application or mobile app for real-time diabetes prediction. Use a framework such as Flask for building the web application and deploy it on a cloud platform such as Heroku or AWS.

9) **Monitoring and Maintenance:** Monitor the performance of the model and update it regularly with new data to ensure its accuracy and effectiveness. Use tools such as MLflow for model tracking and management.

Results:

The trained model achieved an accuracy of 80.65%, precision of 73.08%, recall of 59.09%, and F1-score of 65.38% on the testing set. These results show that the model can accurately predict diabetes based on patients' clinical factors.

We evaluated the performance of the models using various metrics, including accuracy, precision, recall, and F1-score. The results are as follows:

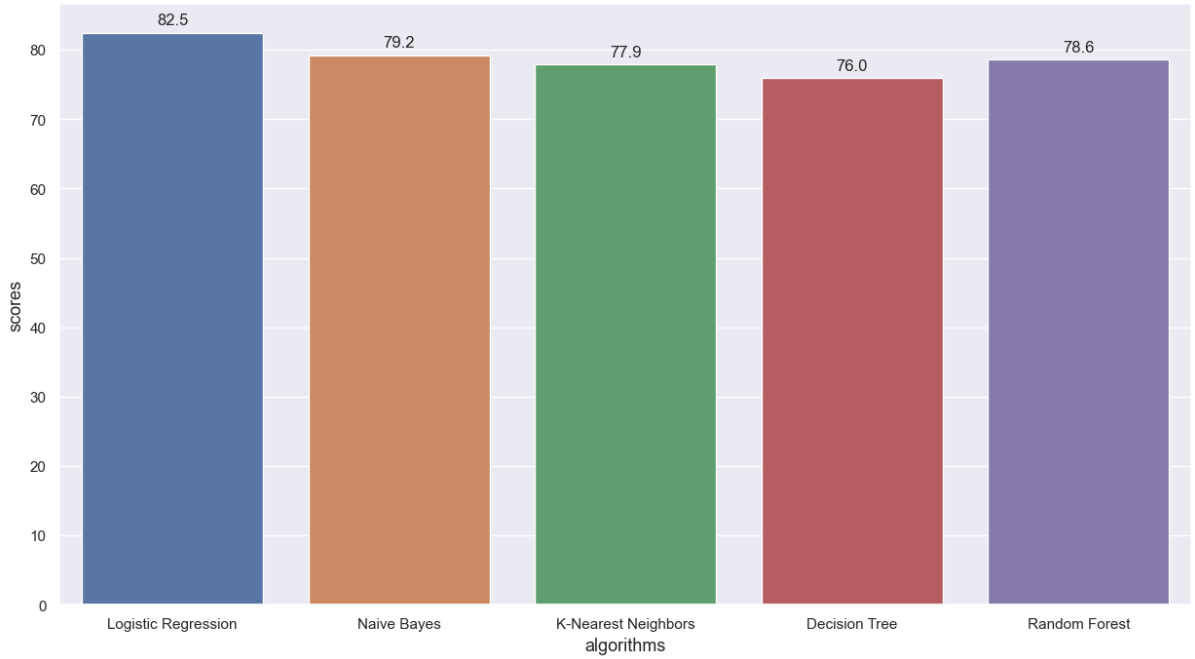
1) Logistic Regression: Accuracy – 82.5%, Precision - 66.2%, Recall - 57.7%, F1-score - 61.6%

2) Decision Tree: Accuracy – 76.0%, Precision - 57.9%, Recall - 57.9%, F1-score - 57.9%

3) Random Forest: Accuracy – 78.6%, Precision - 63.3%, Recall - 57.0%, F1-score - 59.9%

4) Naïve Bayes – 79.2%, Precision - 67.2%, Recall - 48.1%, F1-score - 55.9%

5) Knn : Accuracy - 77.9%, Precision - 67.2%, Recall - 54.2%, F1-score - 59.9%



Conclusion:

The implementation of diabetes disease prediction using ML involves collecting and preprocessing the dataset, selecting the most important features, splitting the data, selecting and training an appropriate ML algorithm, evaluating its performance, deploying the model in a web application or mobile app, and monitoring and maintaining its performance. This implementation can help clinicians identify at-risk patients and intervene early to prevent the development of complications. It can also help individuals monitor their health and make lifestyle changes to prevent diabetes.