**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. Diabetes can be controlled if it is predicted earlier. To achieve this goal this project work we will do early prediction of Diabetes in a human body or a patient for a higher accuracy through applying, Various Machine Learning Techniques. Machine learning techniques Provide better result for prediction by con- structing models from datasets collected from patients. In this work we will use Machine Learning Classification and ensemble techniques on a dataset to predict diabetes. Which is Logistic Regression, Decision Tree Classifier, Random Forest Classifier and Gradient Boosting Classifier. 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 is capable of predicting diabetes effectively. Our Result shows that all the algorithms have performed very well, but us of 4 the accuracy of gradient boost is consistent in both training and testing. The Dataset is collected form Kaggle Repository which contains 768 Instances with 9 features. This study's goal is to predict whether the patient is affected with diabetes or not, where this presence is valued from no presence to likely presence. The researchers accelerating their research works to develop software with the help of machine learning algorithms which can help doctors to decide both prediction and diagnosing the Diabetes more effectively. The main objective of this research project was to predict whether a patient is having diabetes or not using machine learning algorithms.

**Existing System**

In this system, the input details are obtained from the patient. Then from the user inputs, using ML techniques diabetes disease is analysed. Now, the obtained results are compared with the results of existing models within the same domain and found to be improved. The data of heart disease patients collected from the Kaggle Data Repository is used to discover patterns with Logistic Regression, Decision tree, Random Forest and Gradient Boosting. The results are compared for performance and accuracy with these algorithms. The proposed hybrid method returns results of more than 80% competing with the other existing methods.

**Proposed System**

Classification is one of the most important decision-making techniques in many real-world problems. In this work, the main objective is to classify the data as diabetic or non-diabetic and improve the classification accuracy. For many classifications problem, the higher number of samples chosen but it doesn’t lead to higher classification accuracy. In many cases, the performance of algorithm is high in the context of speed but the accuracy of data classification is low. The main objective of our model is to achieve high accuracy. Classification accuracy can be increase if we use much of the data set for training and few data sets for testing. This survey has analysed various classification techniques for classification of diabetic and non-diabetic data. Thus, it is observed that techniques like Logistic Regression, Decision Tree, Random Forest and Gradient Boosting are most suitable for implementing the Diabetes prediction system.

**Problem Objective**

The Main Objective of Developing this project is:

Predict whether a patient should be diagnosed with Diabetes Disease. This is a binary outcome.

* Positive (+) = 1, patient diagnosed with Diabetes Disease.
* Negative (-) = 0, patient not diagnosed with Diabetes Disease
* Experiment with various Classification Models & see which yields greatest accuracy.
* Examine trends & correlations within our data
* Determine which features are most important to Positive/Negative Diabetes Disease diagnosis.

1. To Implement machine learning model to predict future possibility of diabetes disease by various machine learning algorithms.
2. To determine significant risk factors based on dataset which may lead to diabetes diseases.
3. To analyse feature selection methods and understand their working principle.

**Machine Learning Workflow Undertaken:**

We can define the machine learning workflow in 5 stages.

1. Gathering data
2. Data pre-processing
3. Researching the model that will be best for the type of data
4. Training and testing the model
5. Evaluation

**What is the machine learning model?**

The machine learning model is nothing but a piece of code; which an engineer or data scientist models by training it with the data according to the need of the project and making the model learn through the data and allowing it to predict or give the solution that we want whenever we ask it to give. So, whenever we give our model the new data which we want it to predict, we will get the predicted value according to the model training, the trained model might or might not perform well on the test data that we want it to predict, due to various reasons, so before trying to train any model we need to make sure that the algorithm that is going to use is appropriate for the desired class that we want to predict and based on the data that we are using.

**1. Gathering Data**

The process of gathering data depends on the type of project we desire to make, if we want to make an ML project that uses real-time data, then we can build an IoT system that uses different sensors data. Then the sensor data can be connected to the database where we want to store it. But the collected data cannot be used directly for performing the analysis, Since the collected data might be very irrelevant, extremely large values, unorganized text data, or noisy data to the project that we are working on. Therefore, to solve this problem Data Preparation is done meaning data cleaning.

# 2. Data pre-processing

Data pre-processing is one of the most important steps in machine learning. It is the most important step that helps us to provide accurate and cleaned data for training so that we can get accurate results. In machine learning, there is an 80/20 rule, where, every data scientist should spend his/her 80% time for data pre-processing and 20% time to perform the analysis and build the actual machine learning model.

**What is data pre-processing?**

Data pre-processing is a process of cleaning the raw data and making it a meaningful and understandable format. i.e., the data which is collected in the real world is not clean and consists of a lot of irrelevant data and inconsistent data, so we first convert our raw into a meaningful way to generate an efficient machine learning model pipeline. In other words, whenever the data is gathered from different sources it is collected in a raw format and this data isn’t feasible for the analysis.  
Therefore, certain steps are executed to convert the data into a small clean, and meaningful way, where our model can understand. This part of the process is called data pre-processing.

## **Why do we need it?**

As we know that data pre-processing is a process of cleaning the raw data into clean data so that it can be used to train the model. So, data pre-processing is an essential step to generate a machine learning model to help us predict, classify, forecast the data when we pass unknown data.

Below mentioned are some of the examples which mostly occur in raw data, when the data is collected.

1. **Missing data:** Missing data/ missing values are the empty spaces in our data that might have occurred due to various reasons, such as Structured missing values, missing completely at random, Missing at Random, missing not at random, each has its method of treating missing values and in the case of Sensor data, missing values might be high due to technical issues, electricity or other environmental factors.

2. **Noisy data:** This type of data is also called outliers; this can occur due to human errors (humans manually gathering the data) or some technical problem of the device at the time of collection of data. Outliers can be easily termed as extremely low values or extremely high values which are not related to the particular data variable.

3. **Inconsistent data:** This type of data error might mostly happen due to human errors (mistakes with the name or values) or duplication of data. Inconsistent in numbering formats or if the data variable is not consistent through all the rows, then we can term it as an inconsistent data variable, making a data variable consistent is very important since it might result in a certain type of bias when the model is trained and it might also result in inaccurate analysis results when we visualize them onto plots.

## **Three Types of Data**

1. Numeric e.g., Income, Age

2. Categorical e.g., Gender, Nationality

3. Ordinal e.g., Low/Medium/High, Education, Ranking

## How can data pre-processing be performed?

These are a lot of techniques that one can incur to pre-process the data to convert it from raw to a meaningful and understandable format.

1. **Conversion of data:** As we know Machine Learning models can only handle numeric features, hence categorical and ordinal data must be somehow converted into numeric features, we have a lot of transformation functions that we can use on our categorical/ string data type to convert it to numerical formats, such as 1. One Hot Encoding, 2. get dummies (), map ().

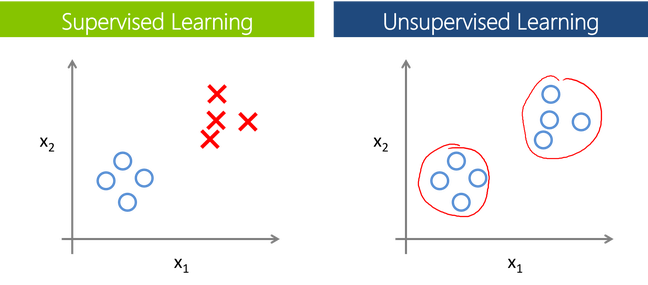
2. **Ignoring the missing values:** Whenever we encounter any missing values in our dataset it depends on the developer working on the project whether to delete the missing values or to impute some other values such as, mean, median, Predicting the missing value, or many other techniques.

3. **Filling the missing values:** The process of deleting the missing values can sometimes be efficient only if the missing values are a handful if the missing values are very large then in this case, we might also check the importance of the variables to our model and we can then delete the entire column with the most number of missing values, or we can impute some other value in place of missing value, by further analysis the dataset or by using statistical measurements, or by using some predicting algorithms to predict the missing values.

4. **Outlier’s detection:** As we have already seen, Outliers can be easily termed as extremely low values or extremely high values which are not related to the particular data variable. These are the error values are present in our data set that deviates drastically from other observations in a data set. [Example: human weight = 800 Kg; due to mistyping of extra 0, the entire value is changed, so this can be treated as an outlier, but in reality, outliers can be identified either by using plots or by statistical analysis.]

So, our main goal should be to train the best performing machine learning model nearly accurately using the cleaned/pre-processed data, so that it can help us to give the solutions whenever something new data is passed onto it related to the data that we have trained.

Note: The better your data is, the better the results will be.



## **Supervised Learning:**

Supervised learning is a branch of machine learning where for each row in the dataset, each row is tagged with a particular label known as the target class.

Supervised Learning is categorized into 2 other categories which are “**Classification**” and “**Regression**”.

## **Classification:**

**The classification**problem is when the target variable is **categorical**(i.e., the output variable consists of classes such as —Class A or B or something else, there might be 2 classes or more than 2 classes.).

A classification problem can be described as a problem where the target class in a dataset consists of categories, such as “Yes” or “No”, or “spam” or “not spam”.

Widely used Classification algorithms:

* **K-Nearest Neighbour**
* **Naive Bayes**
* **Decision Trees/Random Forest**
* **Support Vector Machine**
* **Logistic Regression**

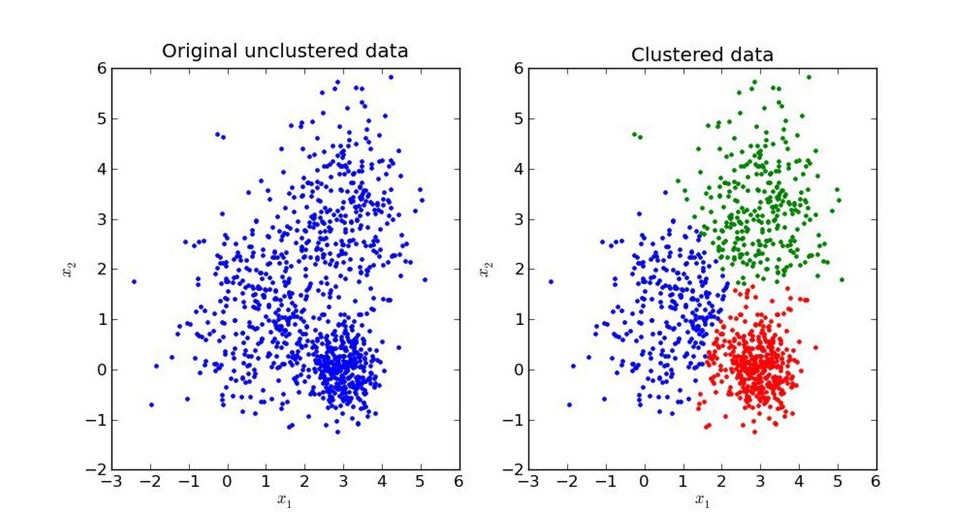
## **Regression:**

While a **Regression**problem is when the target variable is **continuous**(i.e., the output is numeric), Regression problem can be easily termed as the problem where we have to forecast about the future or what we do not know right now, it can be anything (Example: House Price Prediction, Stock market trends)

Widely used Regression Algorithms:

* **Linear Regression**
* **Support Vector Regression**
* **Decision Trees/Random Forest**
* **Gaussian Progresses Regression**
* **Ensemble Methods**

## **Unsupervised Learning:**



Unsupervised Learning is another branch of Machine Learning where we won’t be having any labels for each row of our data unlike supervised learning, so in this case, the model will try to segregate things based on the features and the data available. In simple terms it segregates the data in terms of clusters, the most important thing in unsupervised learning is the curse of finding the optimal k value (the number of clusters we would like to make).

Similar to Supervised, Unsupervised learning can also be categorized into 2 other categories as “**Clustering**” and “**Association**”.

## **Clustering**:

Clustering is a process of learning to assign labels to examples by leveraging an unlabelled dataset, Because the dataset is completely unlabelled, deciding on whether the learned model is optimal is much more complicated than in supervised learning.

Clustering Algorithms available:

* **DBSCAN**
* **HDBSCAN**
* **K-Means Clustering**
* **Hierarchical Clustering**
* **Gaussian Mixtures**
* **Spectral Clustering**

## **Overview of models under categories:**

**Machine Learning**

**Unsupervised**

**Supervised**

**Clustering**

**Regression**

**Classification**

**Decision Tree, Random Forest**

**Decision Tree**

**HDBSCAN**

**Hierarchical**

**Gaussian Mixture**

**K-Means**

**DBSCAN**

**Neural Networks**

**Ensemble Methods**

**SVR, GPR**

**Linear Regression**

**Neural Networks**

**K-Nearest Neighbors**

**SVM**

**Naïve Bayes**

# 4. Training and Testing the model.

Before building any machine learning Project, training is the most important part, where we train our model using the data available and make the machine learn and understand the data, after which when the model has learned from the data, we provide the model with another dataset to evaluate how good our model is performing, if it is performing well, we then test the model using test data, where we get to know the final performance of our model, which can be measure using various metrics, such as Accuracy, recall, precision, and through classification report.

This whole process of building and deploying a model is done using 3 different datasets which are split using train\_test\_split (), which are ‘**Training data**’, ‘**Validation data**’, and ‘**Testing data**’.

First, we train our classifier/regressor model using ‘**training data set**’, we then tune the parameters, to make the model more efficient ad accurate using ‘**validation set**’, and then we test the performance of our classifier on unseen data/ ‘**test data set**’ which should not be or by any means should be used for training or validating, if we use the same data for testing purpose, our model might perform well, but it will lead to overfitting. The test set will only be available during testing the classifier.

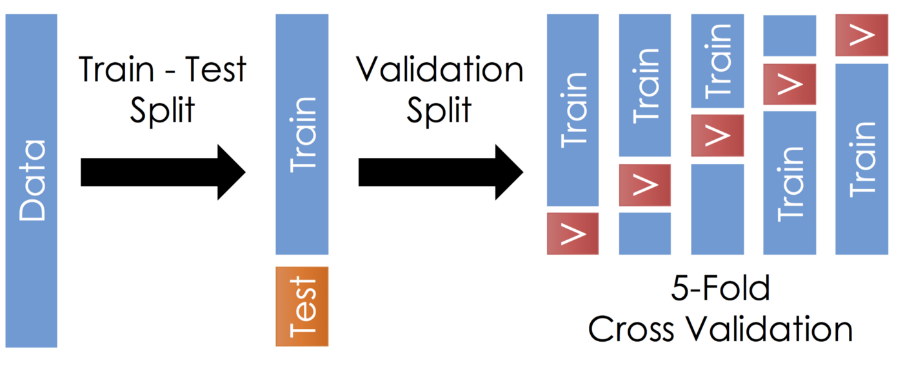


**Training set:** The training set is like a learning material which we give our model to learn so that it understands the data and applies what is understood to the data which we then use that trained model to predict the values with the new data. Training a machine learning model can be done using various algorithms, where the algorithms need to be selected carefully according to the problem that we are trying to solve.

**Validation set:** Cross-validation/ Validation Set is primarily used for estimating how good our model is performing on unseen. Based on the validation data, we can then tune our parameters to make the model more efficient and reliable for deployment.

**Test set:** Test Dataset is simply an unknown dataset, related to the original data which we split at the start so that the model is not aware of the values present in the dataset. The Test dataset is used for assessing the final performance of the model and how well the model is performing.

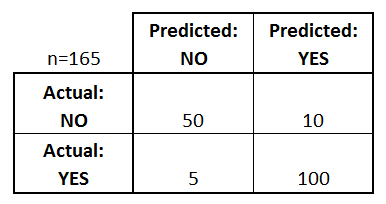
**Simple Example of Train, Test, Validation Splits.**



Once the model is trained by the required parameters and achieved a good accuracy score, we can then use the trained model for predicting our test data/unseen data. Once this is done, we can evaluate our model or plot the evaluation metrics of our model performance using **Classification report, Confusion Matrix**, **AUC\_ROC** Curve, etc.

**Confusion Matrix**

A confusion matrix is simply an evaluation metric matrix especially used for analysing the behaviour of the model and estimating how good the model is performing. It has 4 parameters, which are ‘**True positives’**,**‘True Negatives’**,**‘False Positives,** and ‘**False Negative’**. Which again derived into various formulas such as TPR, FPR, which helps us to get further performance analysis. Below mentioned is an ideal Confusion Matrix, In Confusion matrix the more the TP and Tn the better the model is, although depending on the project we are working on, we might care about FN and FP, when reducing the number of FP and FN, might become an important step during model evaluation.



* **True positives:** Both Predicted and Actual value is True.
* **True negatives:** Both Predicted and Actual Value is False.
* **False positives:** In this case, the actual value is False, but the model has predicted True.
* **False negatives: In this case, the actual value is True, but the model has predicted False.**

Using The TP, TN, FP, FN we can derive some formulas such as:

*Accuracy = (True Positives +True Negatives) / (Total number of classes)*

i.e., for the above example:

Accuracy = (100 + 50) / 165 = 0.9090 (90.9% accuracy)

Similarly, we can do it for Recall, Precision etc.

## 5. Evaluation

Model Evaluation is an integral part of the model development process. It helps us to find the best model that represents our data and it will help us to determine the best parameters which are used to fine-tune the model. This Evaluation step will help us determine the best model out of many models that we create.

**METHODOLOGIES**

**Dataset’s descriptions**:

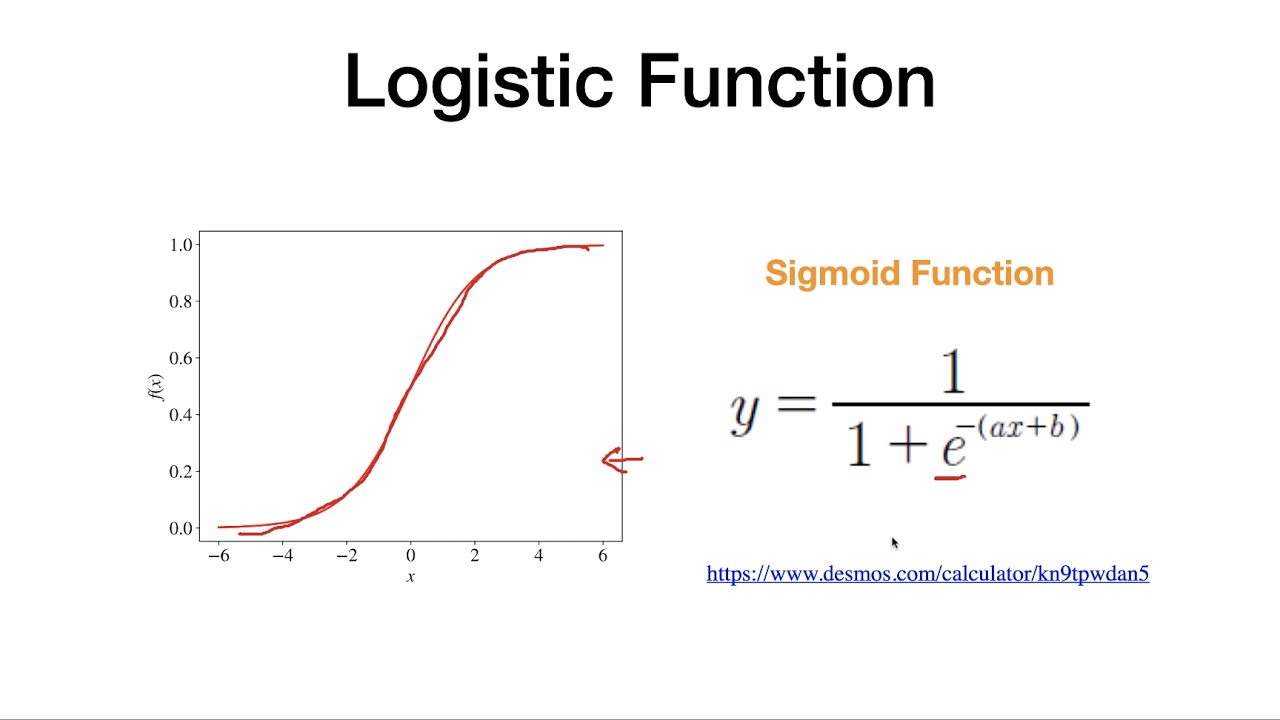
* train.csv - the training set for the fault severity
* test.csv - the test set for fault severity
* sample\_submission.csv – a sample of the correct format for the input
* event\_type.csv: type of event related to the main dataset
* log\_feature.csv - features extracted from log files
* resource\_type.csv: resource type related to the main dataset
* severity\_type.csv: severity type of a warning message coming from the log

All the above CSV's except train.csv, test.csv, and sample\_submission.csv, have been merged to make it has a single CSV file based on a specific primary key.

**Project Outcomes**

The main purpose of designing this system is to predict the risk of future Diabetes disease. we have used Various Machine Learning algorithms to evaluate and train the model, such as Logistic Regression, Decision Tree Classifier, Random Forest Classifier and Gradient Boosting Classifier. These algorithms are discussed below in detail.

1. **Logistic Regression**: Logistic Regression is a supervised classification algorithm. It is a predictive analysis algorithm based on the concept of probability. Logistic Regression relies highly on the proper presentation of data. So, to make the model more powerful, important features from the available data set are selected using Backward elimination and recursive elimination techniques. This type of statistical analysis (also known as logit model) is often used for predictive analytics and modelling, and extends to applications in machine learning. In this analytics approach, the dependent variable is finite or categorical: either A or B (binary regression) or a range of finite options A, B, C or D (multinomial regression). It is used in statistical software to understand the relationship between the dependent variable and one or more independent variables by estimating probabilities using a logistic regression equation. This type of analysis can help you predict the likelihood of an event happening or a choice being made. For example, you may want to know the likelihood of a visitor choosing an offer made on your website — or not (dependent variable). Your analysis can look at known characteristics of visitors, such as sites they came from, repeat visits to your site, behaviour on your site (independent variables). Logistic regression models help you determine a probability of what type of visitors are likely to accept the offer — or not. As a result, you can make better decisions about promoting your offer or make decisions about the offer itself.

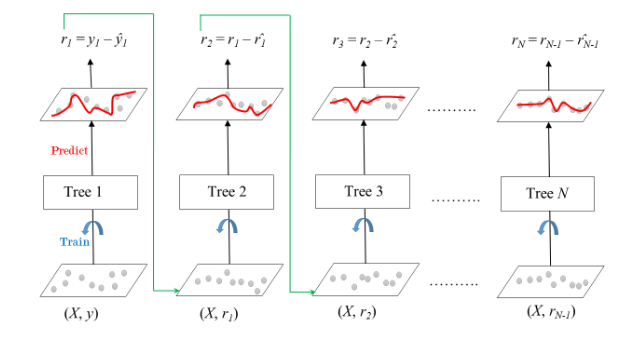


1. **Random Forest Classifier**: Random Forest is a supervised machine learning algorithm. This Technique can be used for both regression and classification tasks but generally performs better in classification tasks. As the name suggests, Random Forest technique considers multiple decision trees before giving an output. So, it is basically an ensemble of decision trees. This technique is based on the belief that a greater number of trees would converge to the right decision. For classification, it uses a voting system and then decides the class whereas in regression it takes the mean of all the outputs of each of the decision trees. It works well with large datasets with high dimensionality. The random forest algorithm is an extension of the bagging method as it utilizes both bagging and feature randomness to create an uncorrelated forest of decision trees. Feature randomness, also known as feature bagging or “[the random subspace method](https://www.stat.berkeley.edu/~breiman/randomforest2001.pdf)”(link resides outside IBM) (PDF, 121 KB), generates a random subset of features, which ensures low correlation among decision trees. This is a key difference between decision trees and random forests. While decision trees consider all the possible feature splits, random forests only select a subset of those features. If we go back to the “should I surf?” example, the questions that I may ask to determine the prediction may not be as comprehensive as someone else’s set of questions. By accounting for all the potential variability in the data, we can reduce the risk of overfitting, bias, and overall variance, resulting in more precise predictions.



1. **Gradient Boosting Classifier:** Gradient boosting is a method standing out for its prediction speed and accuracy, particularly with large and complex datasets. From Kaggle competitions to machine learning solutions for business, this algorithm has produced the best results. We already know that errors play a major role in any machine learning algorithm. There are mainly two types of error, bias error and variance error. Gradient boost algorithm helps us minimize bias error of the model.

Before getting into the details of this algorithm we must have some knowledge about AdaBoost Algorithm which is again a boosting method. This algorithm starts by building a decision stump and then assigning equal weights to all the data points. Then it increases the weights for all the points which are misclassified and lowers the weight for those that are easy to classify or are correctly classified. A new decision stump is made for these weighted data points. The idea behind this is to improve the predictions made by the first stump. I have talked more about this algorithm here. The main difference between these two algorithms is that Gradient boosting has a fixed base estimator i.e., Decision Trees whereas in AdaBoost we can change the base estimator according to our needs.



1. **Decision Tree:** A decision tree is a [decision support](https://en.wikipedia.org/wiki/Decision_support_system) tool that uses a [tree-like](https://en.wikipedia.org/wiki/Tree_(graph_theory)) [model](https://en.wikipedia.org/wiki/Causal_model) of decisions and their possible consequences, including [chance](https://en.wikipedia.org/wiki/Probability) event outcomes, resource costs, and [utility](https://en.wikipedia.org/wiki/Utility). It is one way to display an [algorithm](https://en.wikipedia.org/wiki/Algorithm) that only contains conditional control statements. Decision trees are commonly used in [operations research](https://en.wikipedia.org/wiki/Operations_research), specifically in [decision analysis](https://en.wikipedia.org/wiki/Decision_analysis), to help identify a strategy most likely to reach a [goal](https://en.wikipedia.org/wiki/Goal), but are also a popular tool in [machine learning](https://en.wikipedia.org/wiki/Decision_tree_learning). A decision tree is a [flowchart](https://en.wikipedia.org/wiki/Flowchart)-like structure in which each internal node represents a "test" on an attribute (e.g. whether a coin flip comes up heads or tails), each branch represents the outcome of the test, and each leaf node represents a class label (decision taken after computing all attributes). The paths from root to leaf represent classification rules. In [decision analysis](https://en.wikipedia.org/wiki/Decision_analysis), a decision tree and the closely related [influence diagram](https://en.wikipedia.org/wiki/Influence_diagram) are used as a visual and analytical decision support tool, where the [expected values](https://en.wikipedia.org/wiki/Expected_value) (or [expected utility](https://en.wikipedia.org/wiki/Expected_utility)) of competing alternatives are calculated.

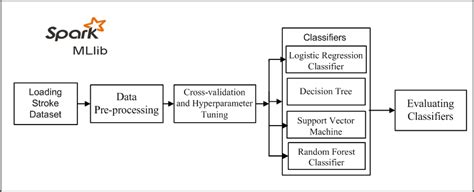


A decision tree consists of three types of nodes

1. Decision nodes – typically represented by squares
2. Chance nodes – typically represented by circles
3. End nodes – typically represented by triangles

Decision trees are commonly used in [operations research](https://en.wikipedia.org/wiki/Operations_research) and [operations management](https://en.wikipedia.org/wiki/Operations_management). If, in practice, decisions have to be taken online with no recall under incomplete knowledge, a decision tree should be paralleled by a [probability](https://en.wikipedia.org/wiki/Probability) model as a best choice model or online selection model [algorithm](https://en.wikipedia.org/wiki/Algorithm). Another use of decision trees is as a descriptive means for calculating [conditional probabilities](https://en.wikipedia.org/wiki/Conditional_probability). Decision trees, [influence diagrams](https://en.wikipedia.org/wiki/Influence_diagrams), [utility functions](https://en.wikipedia.org/wiki/Utility_function), and other [decision analysis](https://en.wikipedia.org/wiki/Decision_analysis) tools and methods are taught to undergraduate students in schools of business, health economics, and public health, and are examples of operations research or [management science](https://en.wikipedia.org/wiki/Management_science) methods.

**Architectural Design**



**Flow Chart**

**Testing Dataset**

**Training Dataset**

**Algorithm**

**Evaluation**

**Model**

**Production data**

**Data**

**Prediction**

**Advantages**

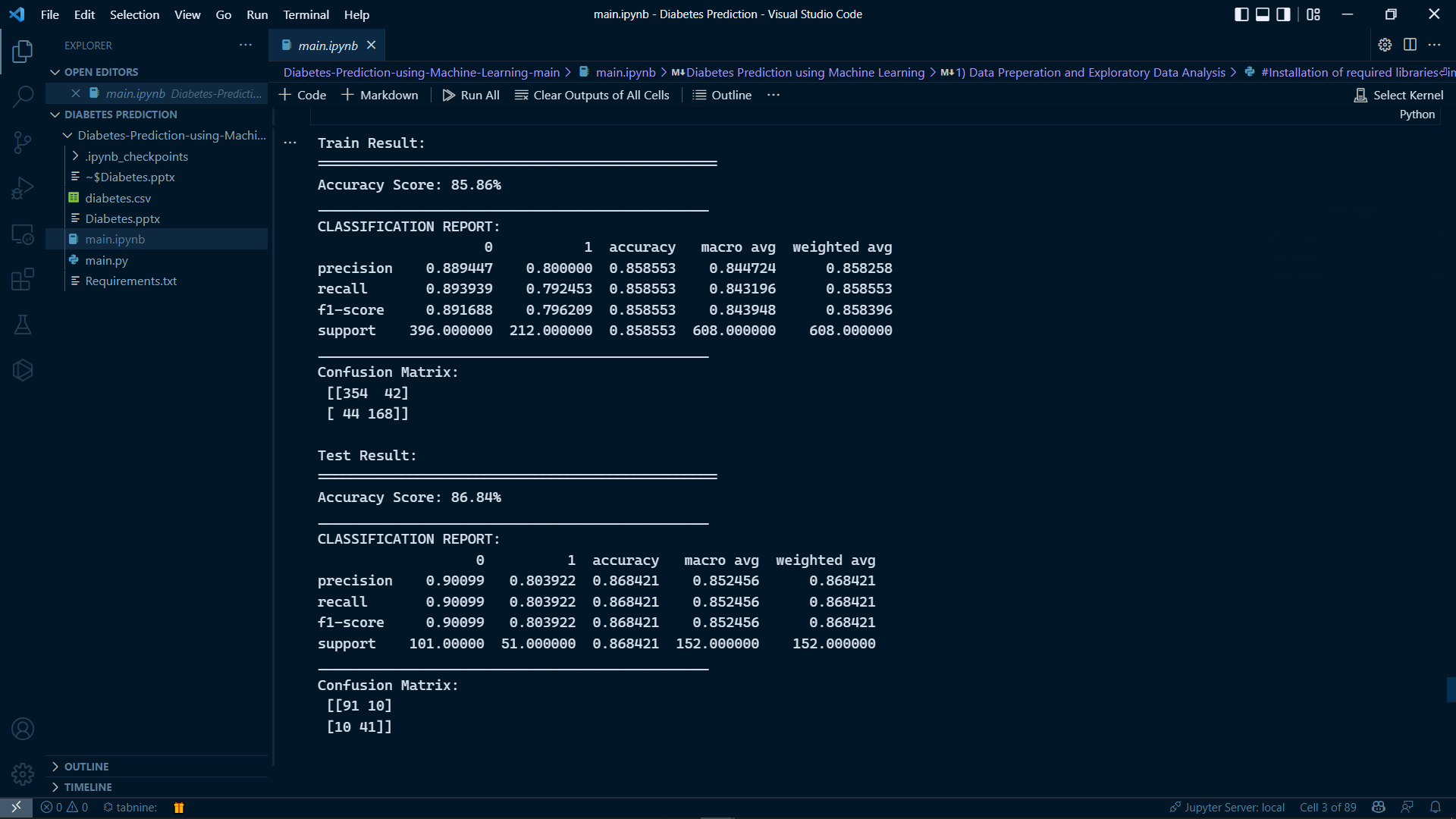
1. Increased accuracy for effective Diabetes disease diagnosis.
2. Handles roughest(enormous) amount of data using random forest algorithm and feature selection.
3. Reduce the time complexity of doctors.
4. Cost effective for patients.

**Disadvantages**

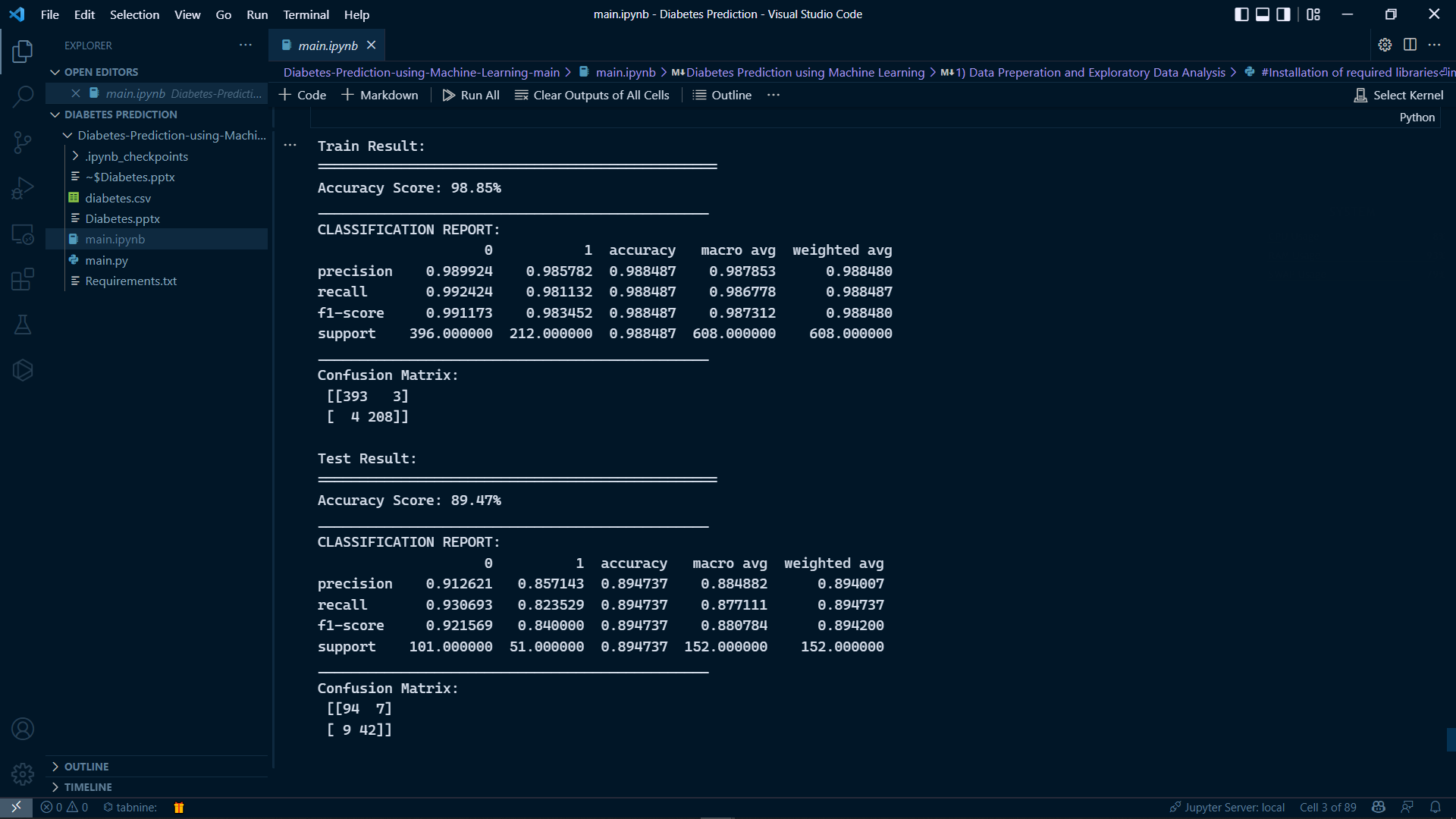
1. Prediction of Type of Diabetes disease results might not be accurate.
2. Data mining techniques does not help to provide effective decision making.

**Experimental Results of Algorithms**

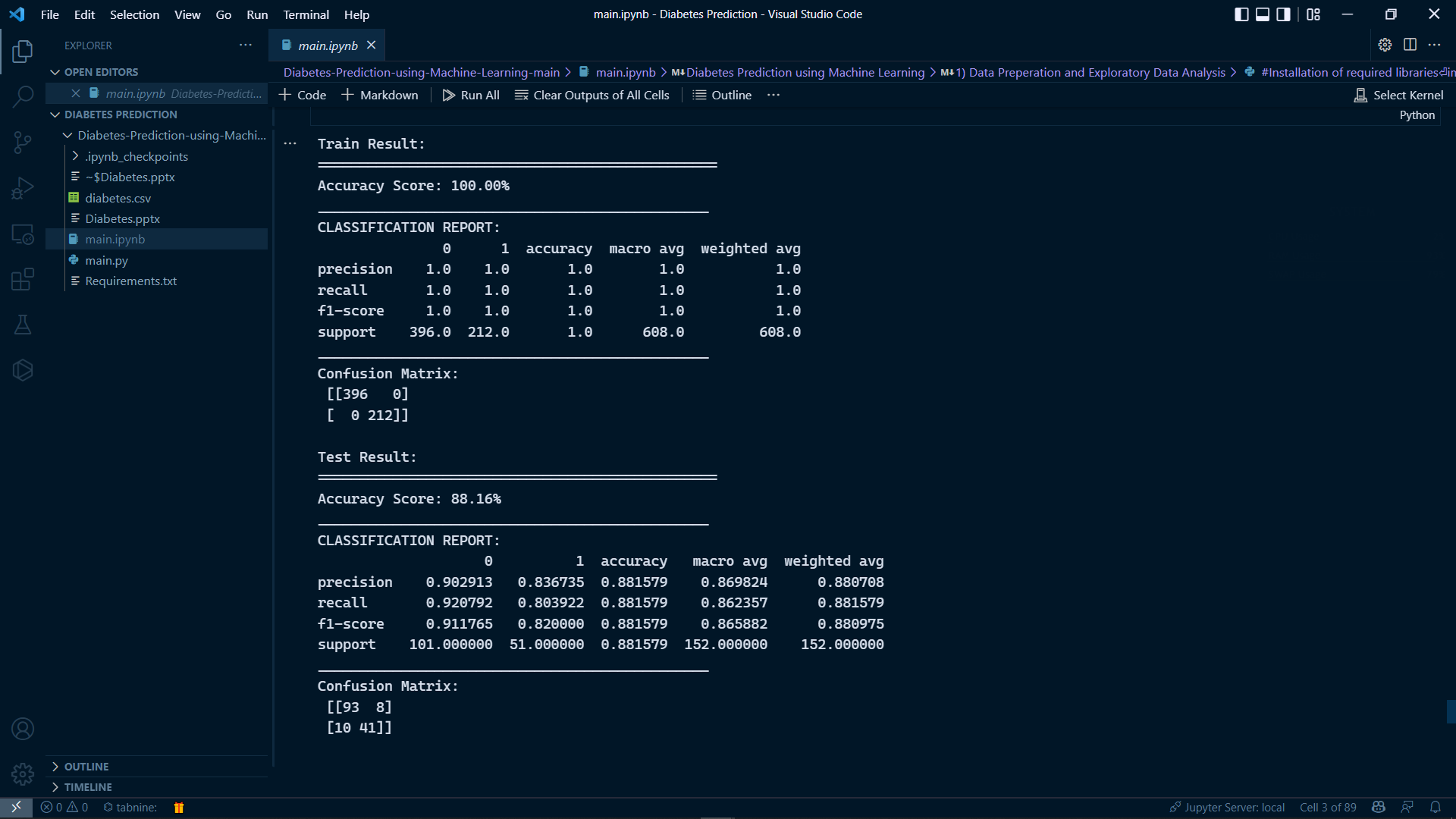
1. **Logistic Regression**



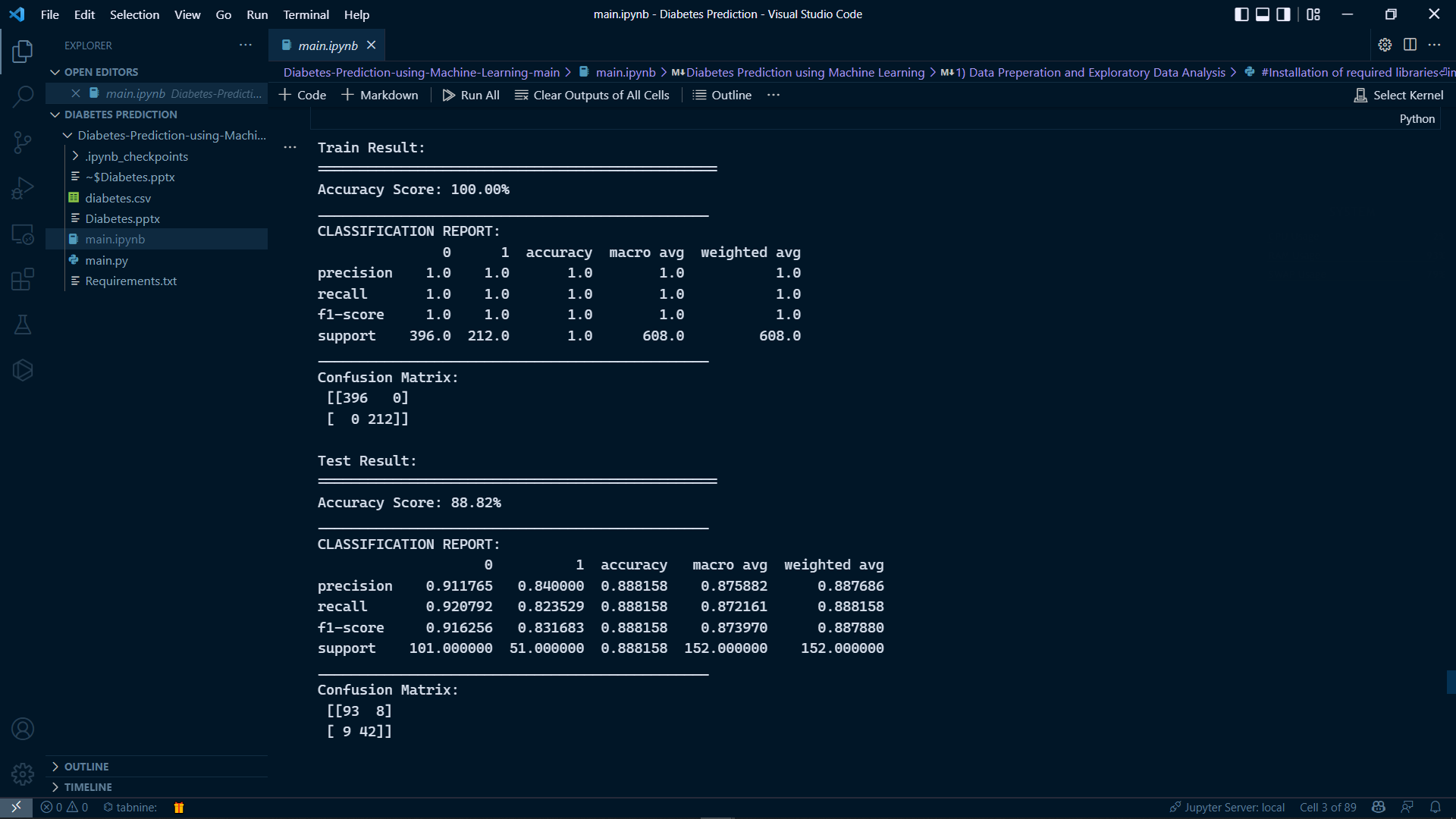
1. **Decision Tree**

****

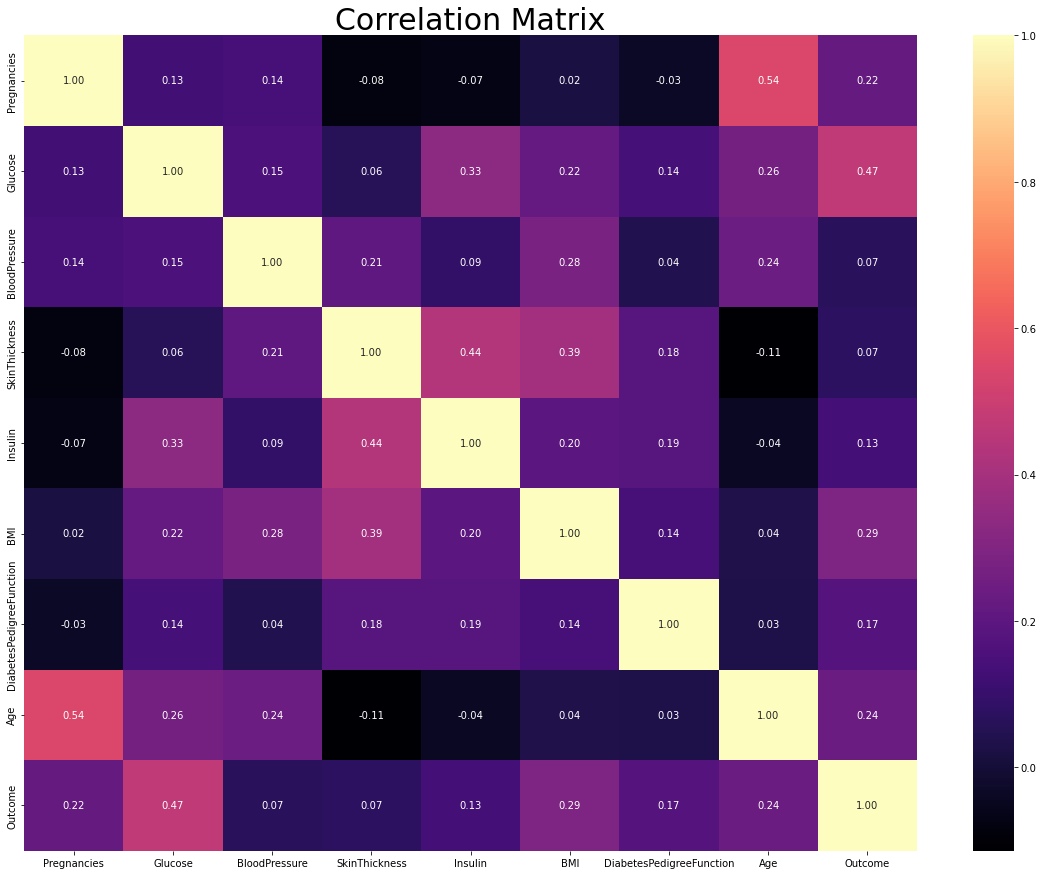
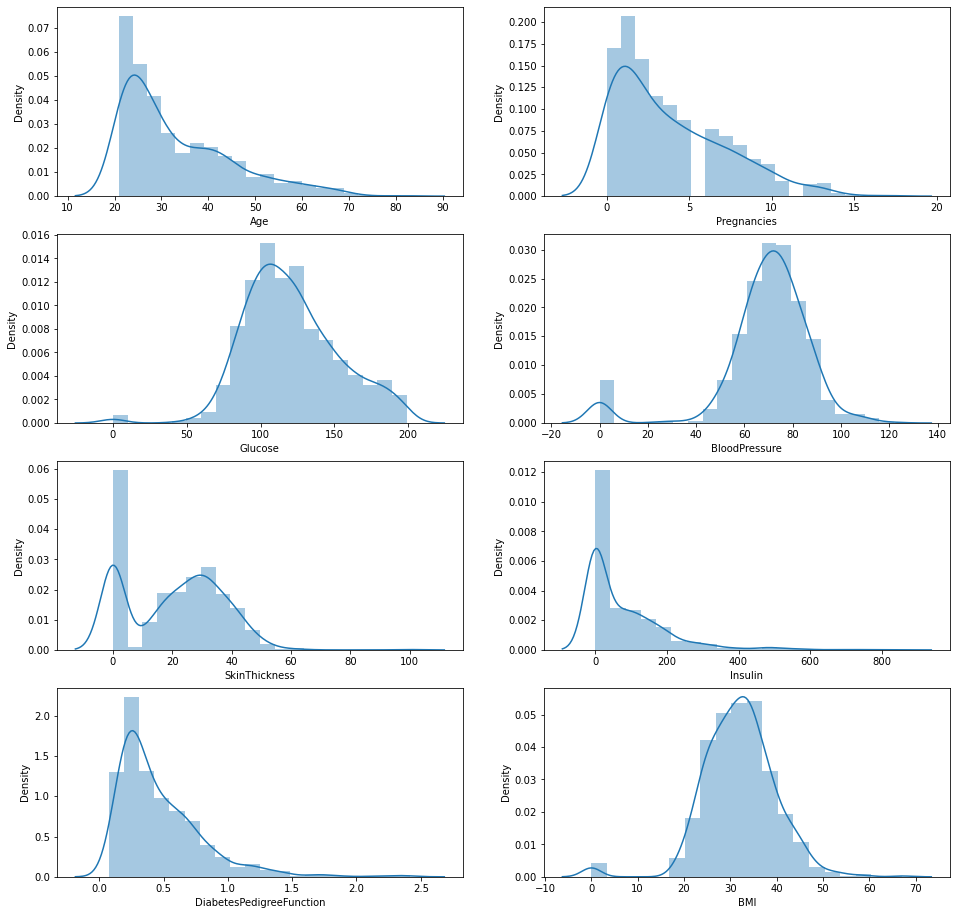
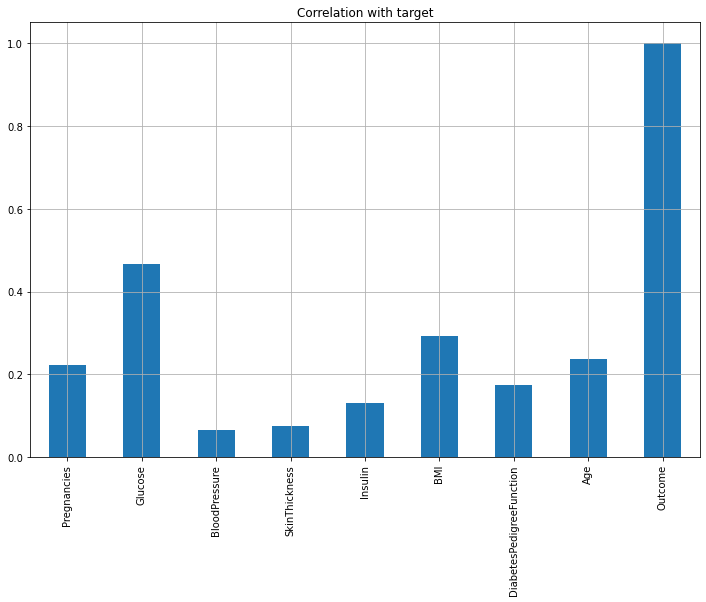
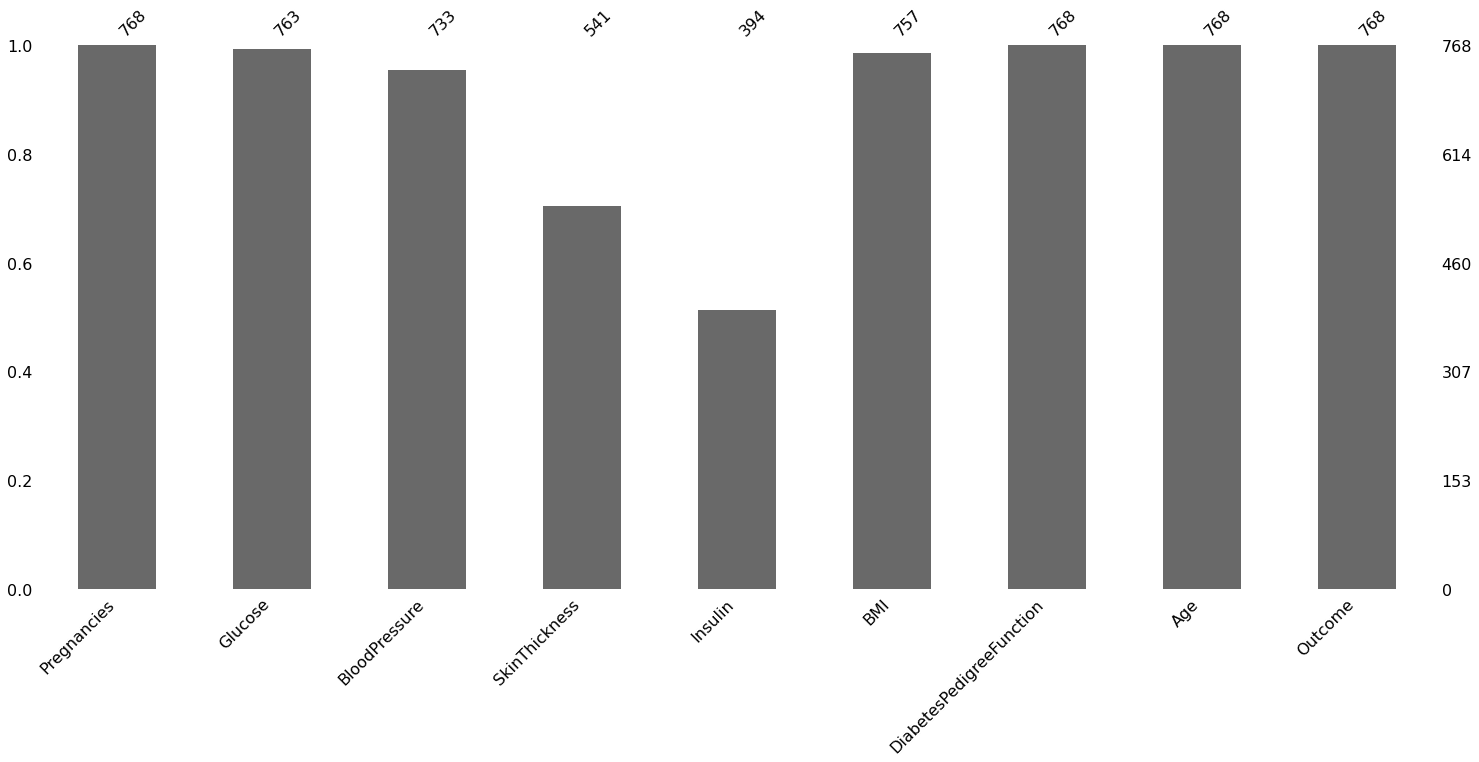
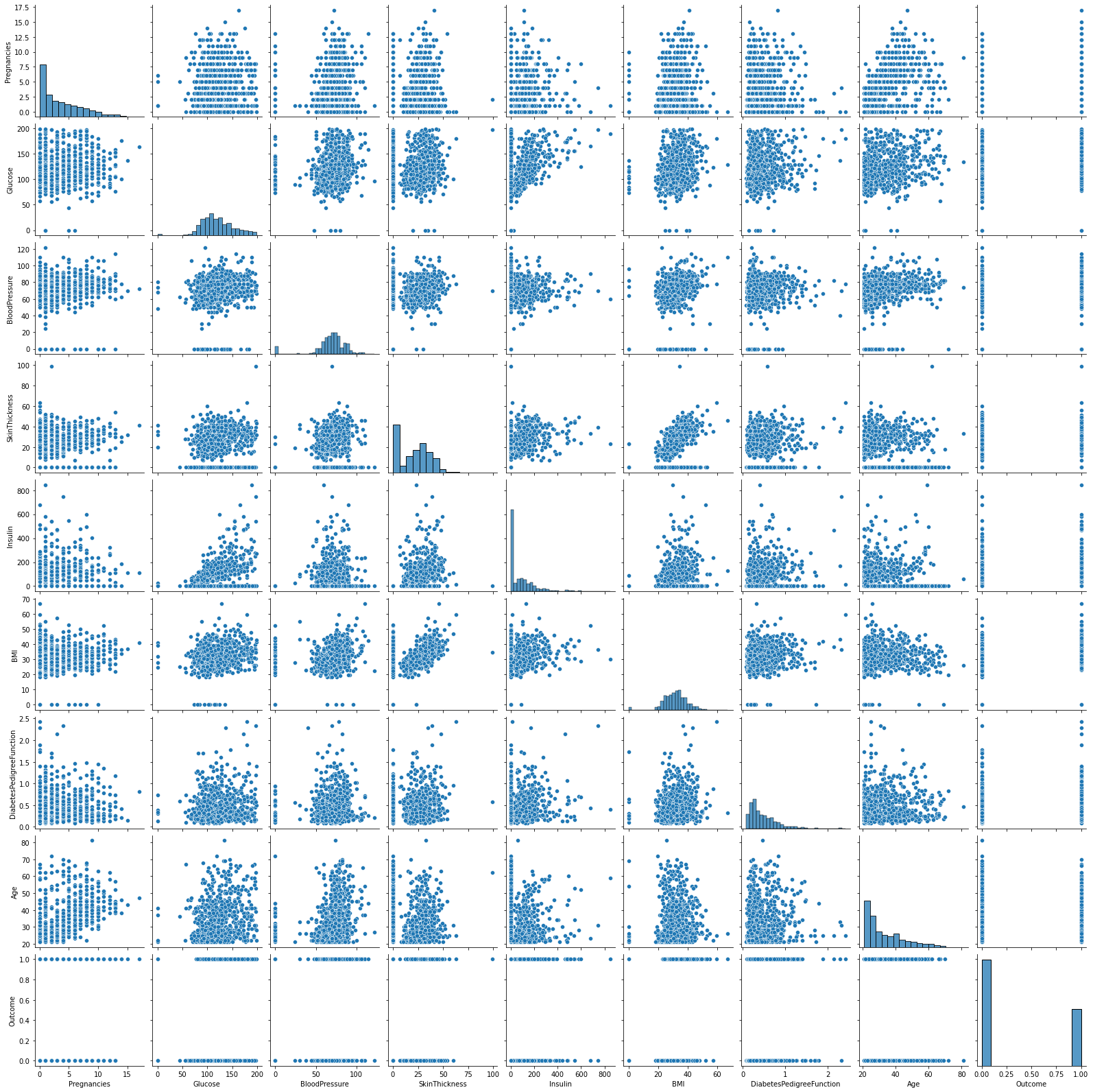
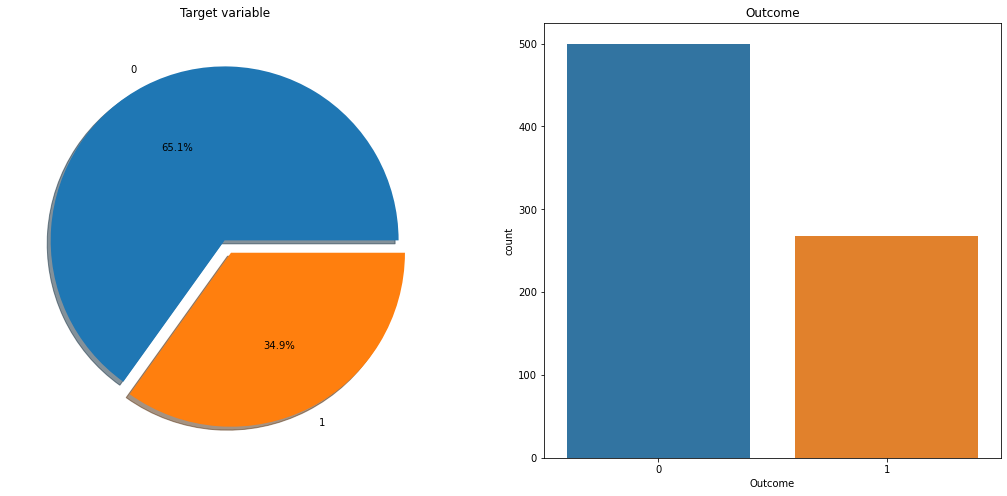
1. **Random Forest**

****

1. **Gradient Boosting Classifier**

****

**Results**

1. **Correlation Matrix**
2. **Density Plot**
3. **Feature Importance** 
4. **Missing Values**
5. **Pair Plot**
6. **Outcome Variable**

**Conclusion**

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 Random Forest, Decision Tree, Logistic Regression and Gradient Boosting classifiers are used. And more than 80% classification accuracy has been achieved. The Experimental results can be assisted in health care to take early prediction and make early decision to cure diabetes and save humans life. Machine learning has the great ability to revolutionize the diabetes risk prediction with the help of advanced computational methods and availability of large amount of epidemiological and genetic diabetes risk dataset. Detection of diabetes in its early stages is the key for treatment. This work has described a machine learning approach to predicting diabetes levels. The technique may also help researchers to develop an accurate and effective tool that will reach at the table of clinicians to help them make better decision about the disease status.

**References**

1. <https://www.ijert.org/diabetes-prediction-using-machine-learning-techniques>
2. <https://www.cdc.gov/diabetes/basics/diabetes.html>
3. <https://www.kaggle.com/datasets/mathchi/diabetes-data-set>