***Helping patients know whether they are diabetic or not using Machine Learning***

Diabetes is an illness which is basically caused by high glucose level in the human body. If left untreated it can cause a lot of complications which might also affect to the most of the organs present in the body. In simple, terms diabetes is caused when the blood glucose level is higher than normal, which is basically cause when there is high or low amount of insulin secretion in the body. The process involved in getting diabetes is basically the carbohydrates in the food which we eat is digested in the intestines and convert it into Sugar called glucose, this glucose which is flowing in our blood should be passed to the body cells, so in order to do this the pancreas create a hormone which is insulin, this helps the glucose present in the blood to be flown to body cells. There are different types of Diabetes i.e.

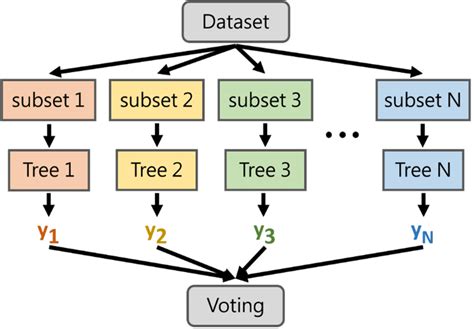
1. Tyep-1: No insulin is generated by the pancreas.
2. Type-2: Insulin is generated but the body cells are not able to sue them properly.
3. Type-3 / gestational diabetes: This is basically caused during the pregnancy which changes the metabolism of a human and these changes are the results of hormones being produced that keep insulin from doing its job.

***Base Paper***

1. <https://www.researchgate.net/publication/351607168_Diabetes_Prediction_System_using_Machine_Learning>
2. <https://www.sciencedirect.com/science/article/pii/S1877050920300557>

***Algorithm Description***

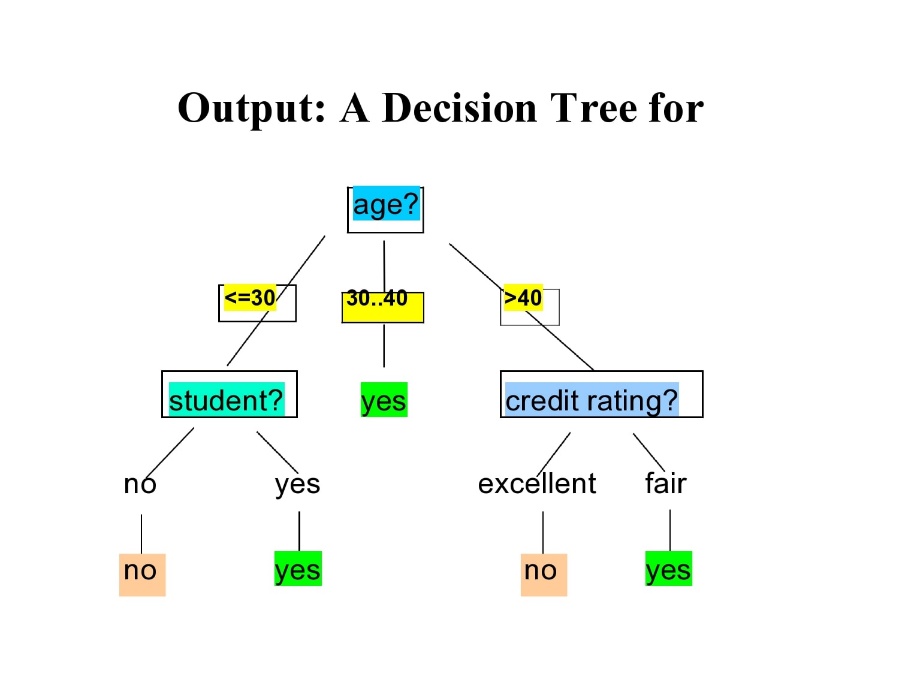
***Random Forest Classifier:*** Random Forest Classifier is an ensemble algorithm which works with multiple algorithms parallelly. This is a supervised algorithm and it can be used with both classification and regression problems. The output of the new data is estimated either by using majority voting or average voting technique. Since the algorithm works with bagging technique, multiple decision trees are used to provide the output for the specific input. 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. Random forest works best with large datasets and high dimensional.



***References:***

1. <https://www.geeksforgeeks.org/random-forest-regression-in-python/>
2. <https://www.geeksforgeeks.org/random-forest-regression-in-python/>

***Decision Tree Classifier:*** A decision tree is a tool for making decisions and the process for making decisions is in a tree like structure, decision tree is a supervised machine learning algorithm mainly used for predicting the outcome after computing all the attributes. The process flow of Decision tree goes from Root node to leave node i.e., the decision node.

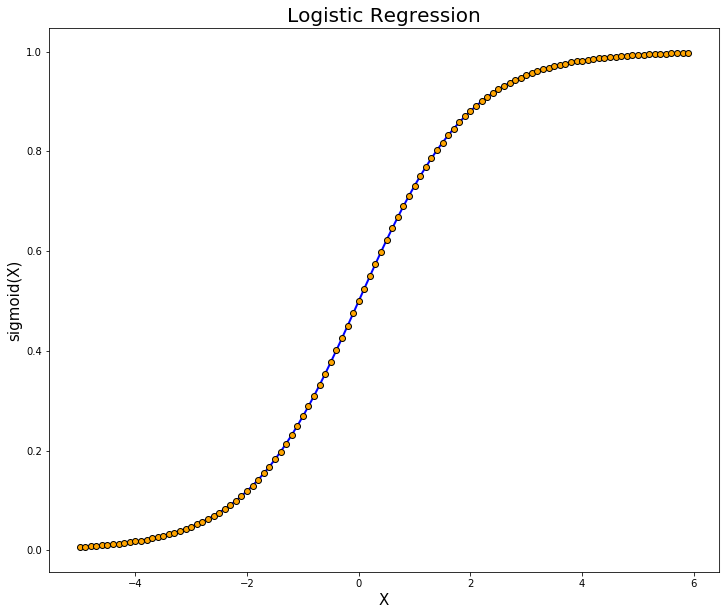


***References:***

1. <https://www.ibm.com/topics/decision-trees>
2. <https://www.mastersindatascience.org/learning/introduction-to-machine-learning-algorithms/decision-tree/>

***Logistic Regression:*** Logistic Regression is a Supervised algorithm which mostly works in the case of binary classification problems. Logistic regression is a sophisticated algorithm where the data to be trained using this algorithm should be properly presented i.e., Normalized/Scaled, Columns should be Converted to numerical and data should be neat and clean. The output is presented in the form of logit score, where this helps us to predict the likelihood of an event occurring of a given problem. The main reason of getting a **S** curve in the below chart is that the sigmoid function does the trick of converting the given number in the range between 0 and 1.

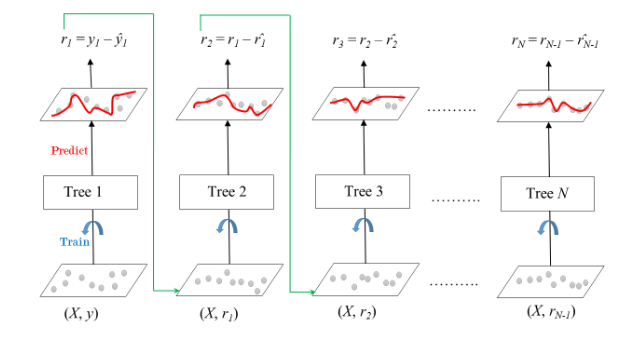
Sigmoid(x) = ***Y = 1 / 1+e -z***



***References:***

1. <https://www.geeksforgeeks.org/understanding-logistic-regression/>
2. <https://www.ibm.com/topics/logistic-regression>

***Gradient Boosting Classifier:*** Gradient Boosting is a boosting algorithm which consists of ensemble of classification and regression trees. In GB each predictor tries to correct the error made by its predecessor. Unlike the Adaboost algorithm here we don’t change the training instances, instead each predictor/tree is trained by the residual errors caused by the previous trees. All the leaners/trees in the gradient boosting have the same equal weights, since the weights are referred as learning rate which is small in magnitude.



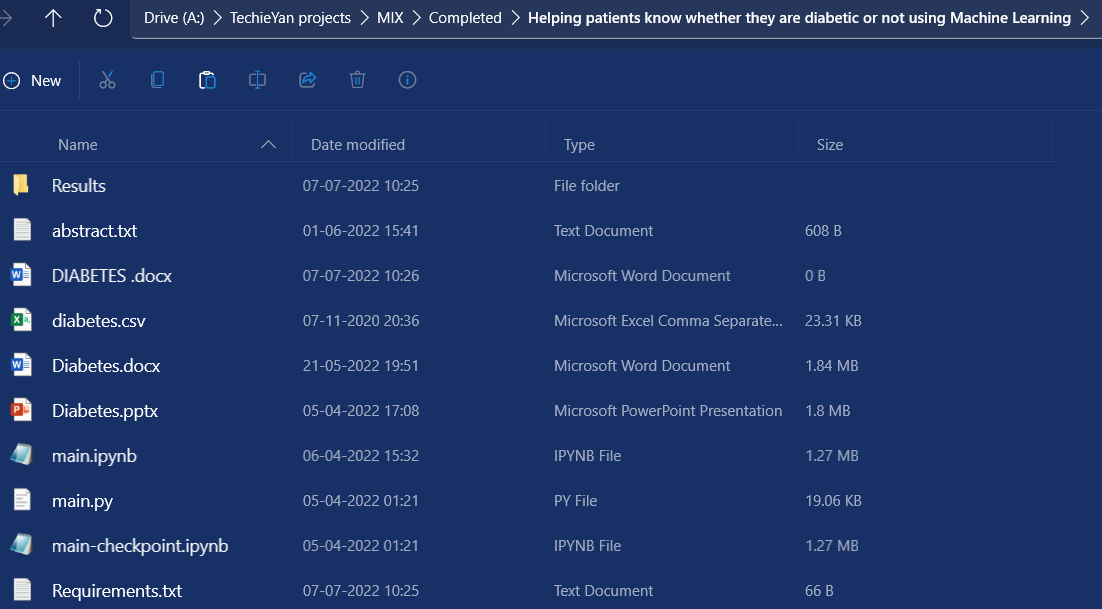
***Steps to Execute the Code!***

**Note:** Make sure you have added path while installing the software’s.

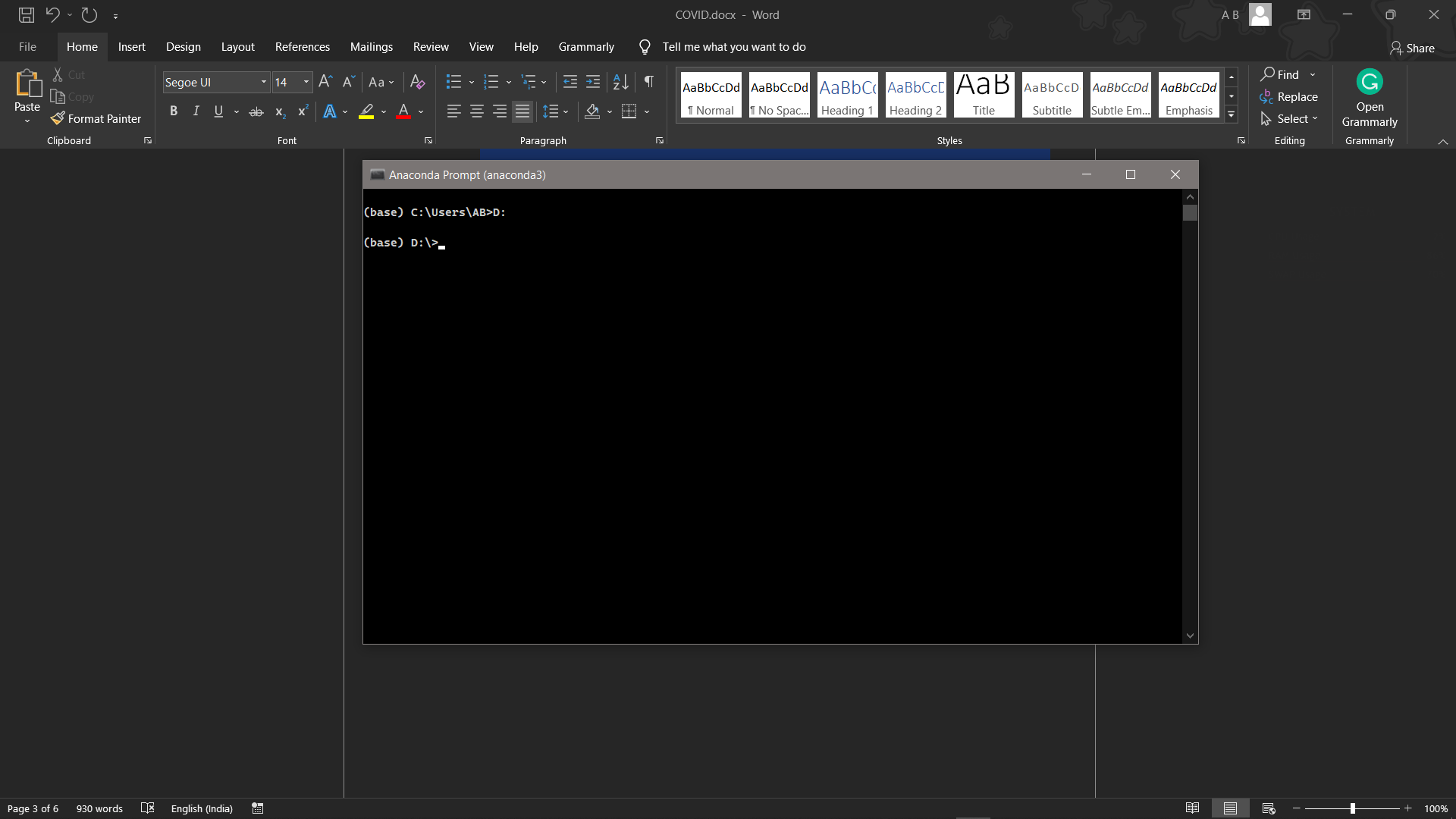
<https://techieyantechnologies.com/2022/07/how-to-install-anaconda/>

<https://techieyantechnologies.com/2022/06/get-started-with-creating-new-environment-in-anaconda-configuring-jupyter-notebook-and-installing-libraries-using-requirements-txt-2/>

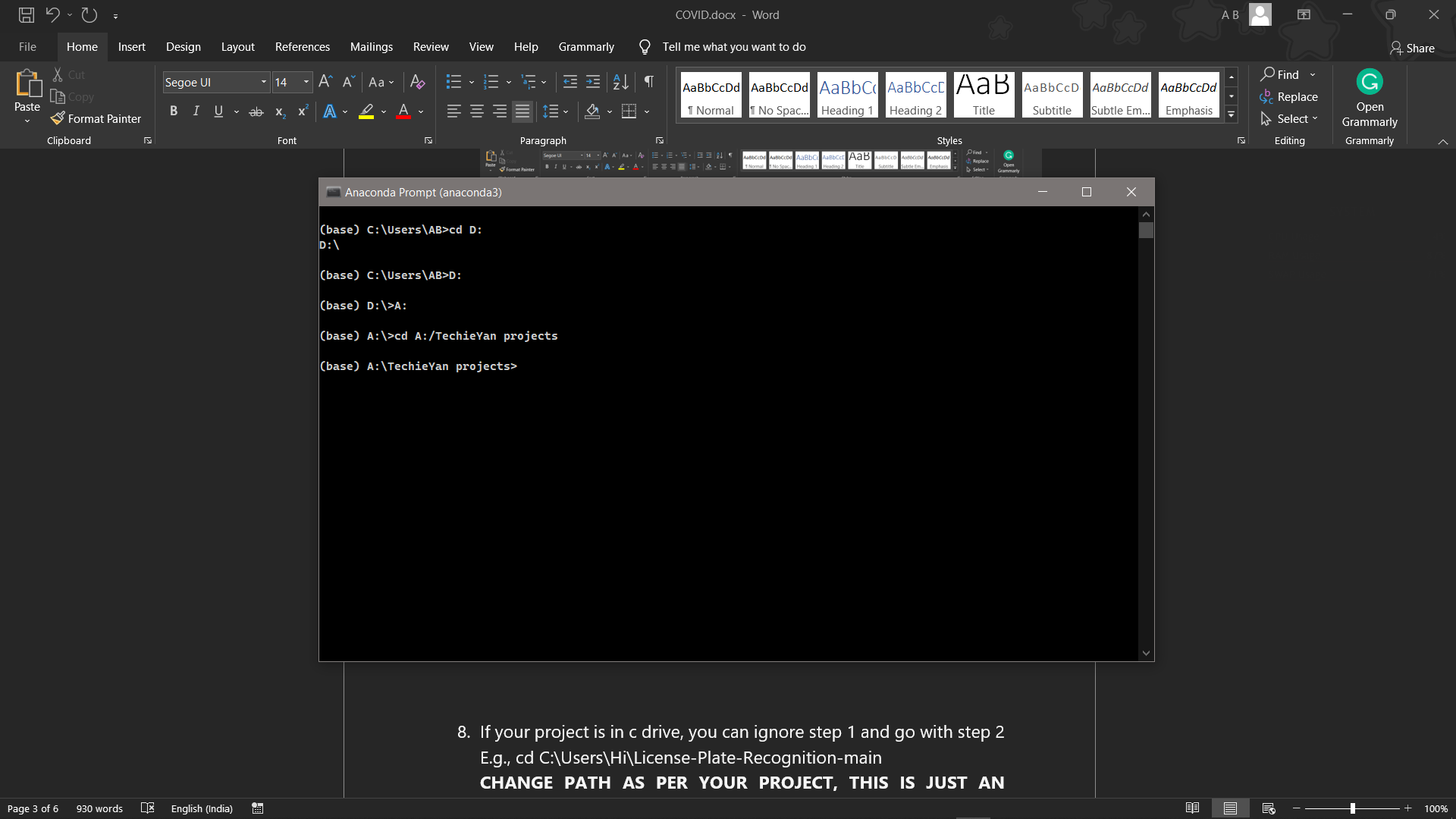
1. Install the prerequisites/software’s required to execute the code from reading the above blog which is provided in the link above.
2. Press windows key and type in anaconda prompt a terminal opens up.
3. Go to the directory where your requirement.txt file is present, not just requirement.txt, if you want to execute any .py or .ipynb files, you need to go to that specific folder or path, where they are saved.



1. <<directory of your file:>>. E.g., If my file is in d drive, then
2. Type d:



1. cd d:\License-Plate-Recognition-main #CHANGE PATH AS PER YOUR PROJECT, THIS IS JUST AN EXAMPLE

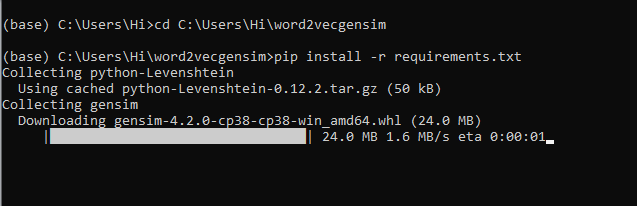


1. If your project is in c drive, you can ignore step 4 and go with step 5.

E.g., cd C:\Users\Hi\License-Plate-Recognition-main

**CHANGE PATH AS PER YOUR PROJECT, THIS IS JUST AN EXAMPLE**

1. Run pip install -r requirements.txt or conda install requirements.txt (Requirements.txt is a text file consisting of all the necessary libraries required for executing this python file. If it gives any error while installing libraries, you might need to install them individually.), example: pip install “module\_name” i.e., pip install pandas



1. To run .py file make sure you are in the anaconda terminal with the anaconda path being set as your executable file/folder is being saved. Then type **python main.py** in the terminal, before running open the main.py and make sure to change the path of the dataset.
2. If you would like to run .ipynb file, Please [follow the link to setup and open jupyter notebook](https://techieyantechnologies.com/2022/06/get-started-with-creating-new-environment-in-anaconda-configuring-jupyter-notebook-and-installing-libraries-using-requirements-txt-2/), You will be redirected to the local server there you can select which ever .ipynb file you’d like to run and click on it and execute each cell one by one by pressing shift+enter.

**Make sure to change the path of the dataset in the code**

***Dataset Description***

The Dataset is collected form Kaggle Repository which contains 769 Instances with 9 features. Some of the features which correspond and highly correlated with our target class are, pregnancies, glucose level, blood pressure, Insulin, Age. 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.



***Issues Faced***

1. We might face an issue while installing specific libraries, in this case, you might need to install the libraires manually. Example: pip install “module\_name/library” i.e., pip install **pandas**

2. Make sure you have the latest or specific version of python, since sometimes it might cause version mismatch.

3. Adding path to environment variables in order to run python files and anaconda environment in code editor, specifically in any code editor.

4. Make sure to change the **paths in the code** accordingly where your dataset/model is saved.

**Refer to the Below links to get more details on installing python and anaconda and how to configure it.**

<https://techieyantechnologies.com/2022/07/how-to-install-anaconda/>

<https://techieyantechnologies.com/2022/06/get-started-with-creating-new-environment-in-anaconda-configuring-jupyter-notebook-and-installing-libraries-using-requirements-txt-2/>

***Note:***

**All the required data has been provided over here. Please feel free to contact me for model weights and if you face any issues.**

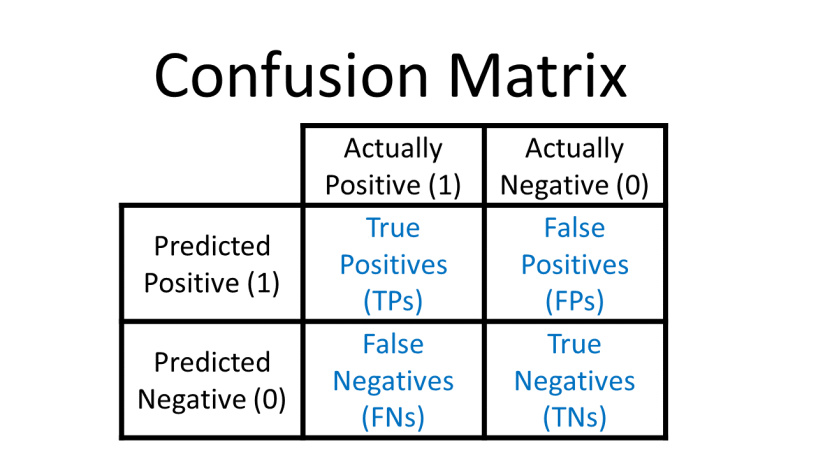
<https://www.linkedin.com/in/abhinay-lingala-5a3ab7205/>

***Yes, you now have more knowledge than yesterday, Keep Going.***

***Evaluation Metrics***

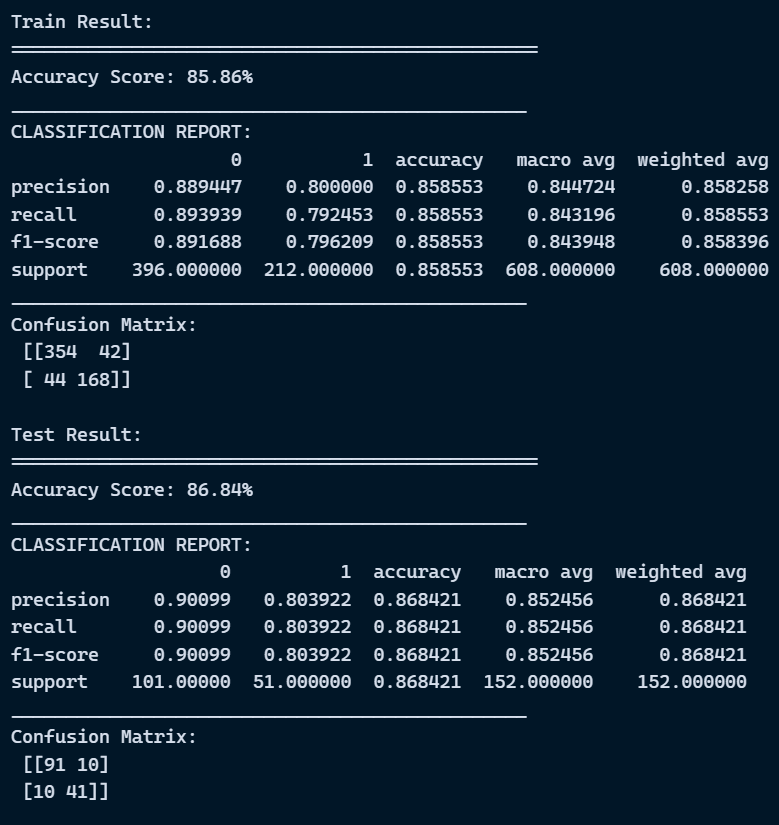
Evaluation metrics are considered as one of the most important steps in any machine learning and deep learning projects, where it will allow us to evaluate how good our model is performing on the new data or on unseen data. There are a lot of evaluation metrics which can be used in order to assess how good our model is performing such as roc\_auc\_curve, f1\_score, recall, precision and each of which work for specific problem we deal. So, for our project we have gone with confusion matrix and classification report which helps us to evaluate not just the accuracy of the model but also the other metrics such as precision, recall and f1\_score.

***Confusion matrix:***

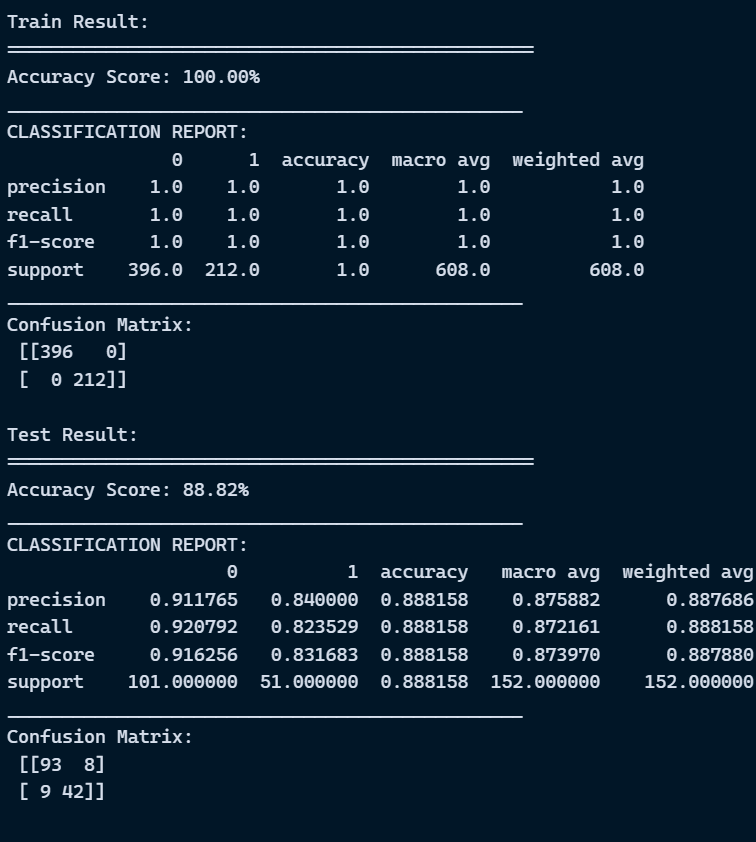


***Results***

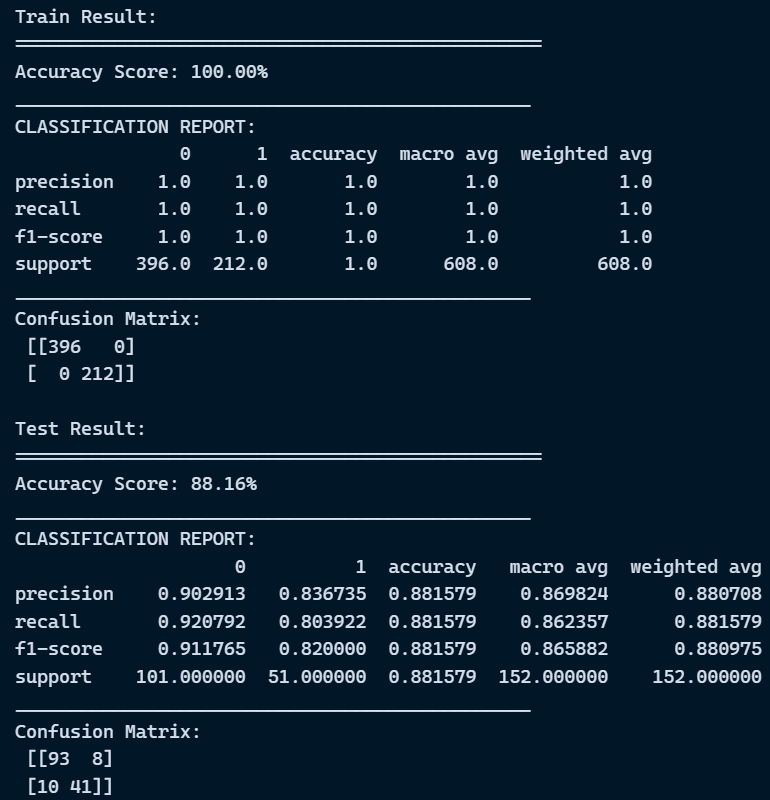
1. ***Logistic Regression***

******

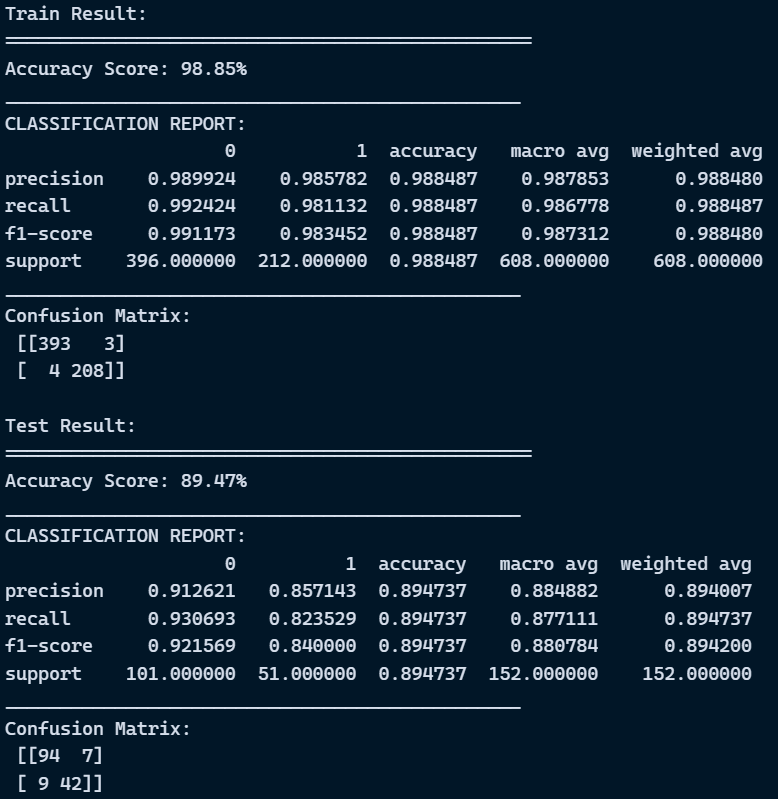
1. ***Decision Tree***

******

1. ***Random Forest Classifier***



1. ***Support Vector Machine***



***Exploratory Data Analysis***

