***Identifying weight category of person using Random Forest***

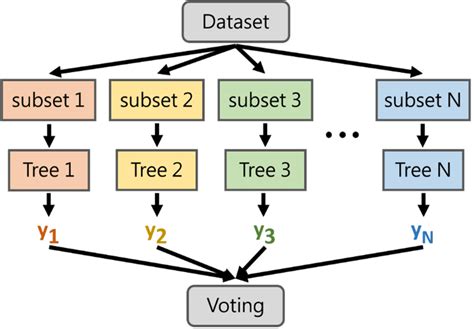
Identifying the weight of a person involves in classifying multiple target classes and since categorising each variable of a person with a tree like structure makes it easy to understand for a model, we use Random Forest algorithm which makes a tree structure which can make the prediction of target class easy and efficient.

***Base Paper***

1. <https://www.sciencedirect.com/science/article/pii/S2666518221000401>

***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/>

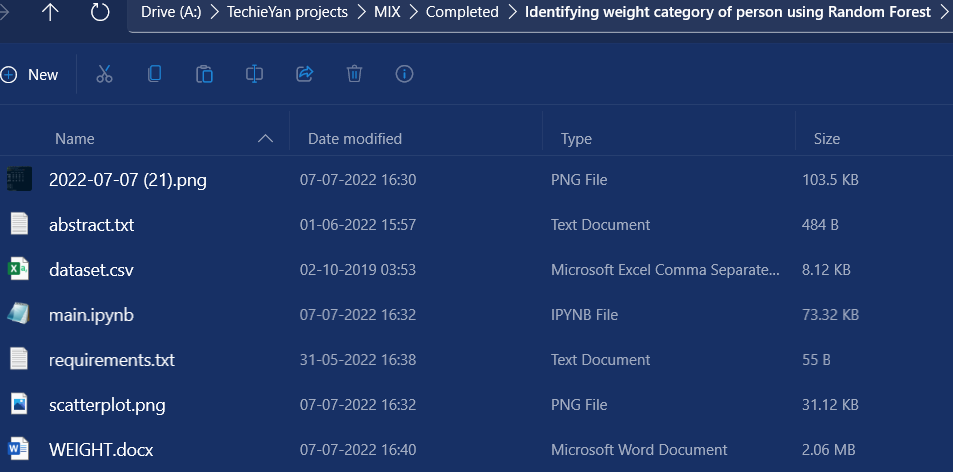
***Steps to Execute the Code!***

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

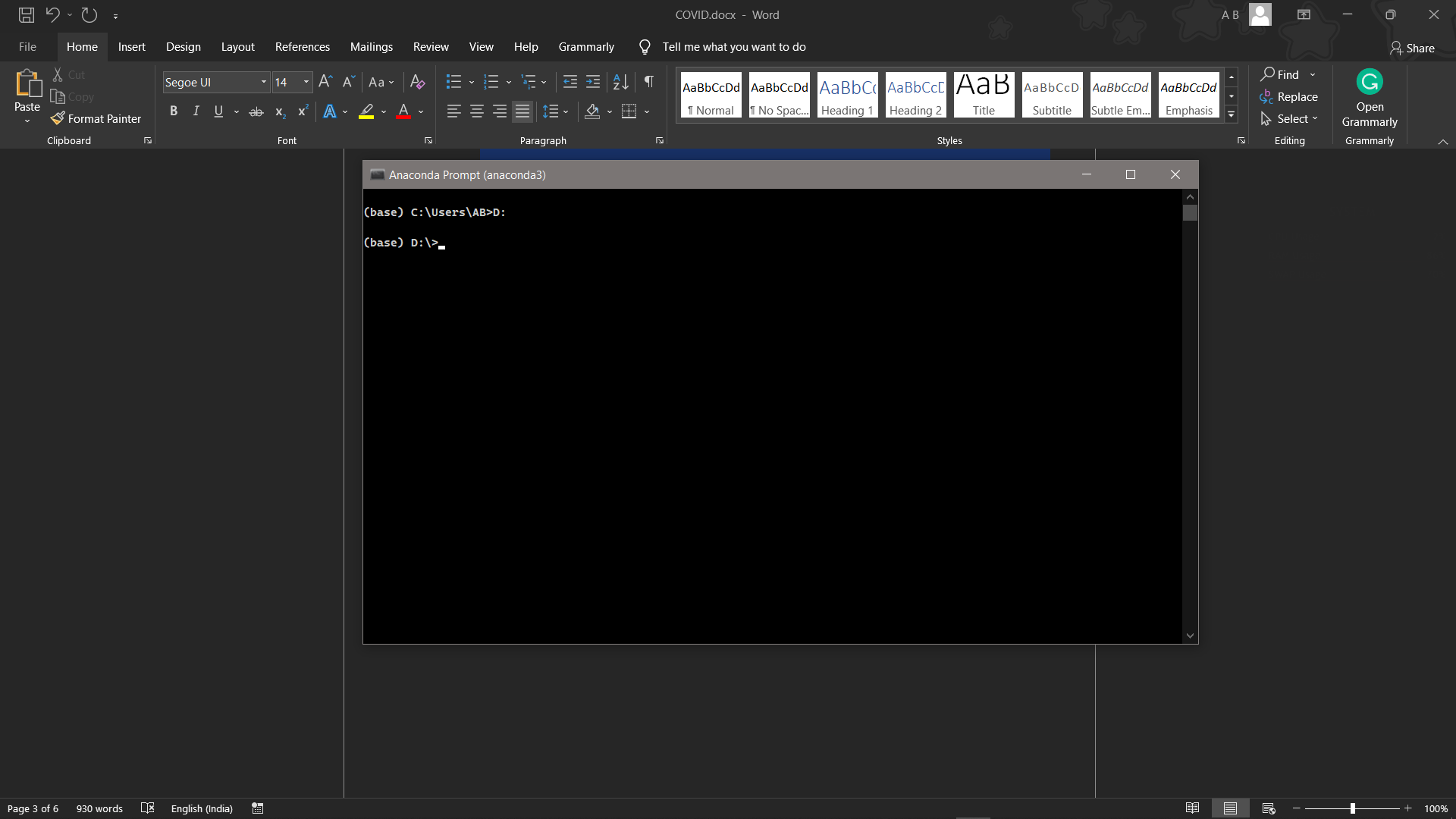
<https://techieyantechnologies.com/2022/07/how-to-install-anaconda/>

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

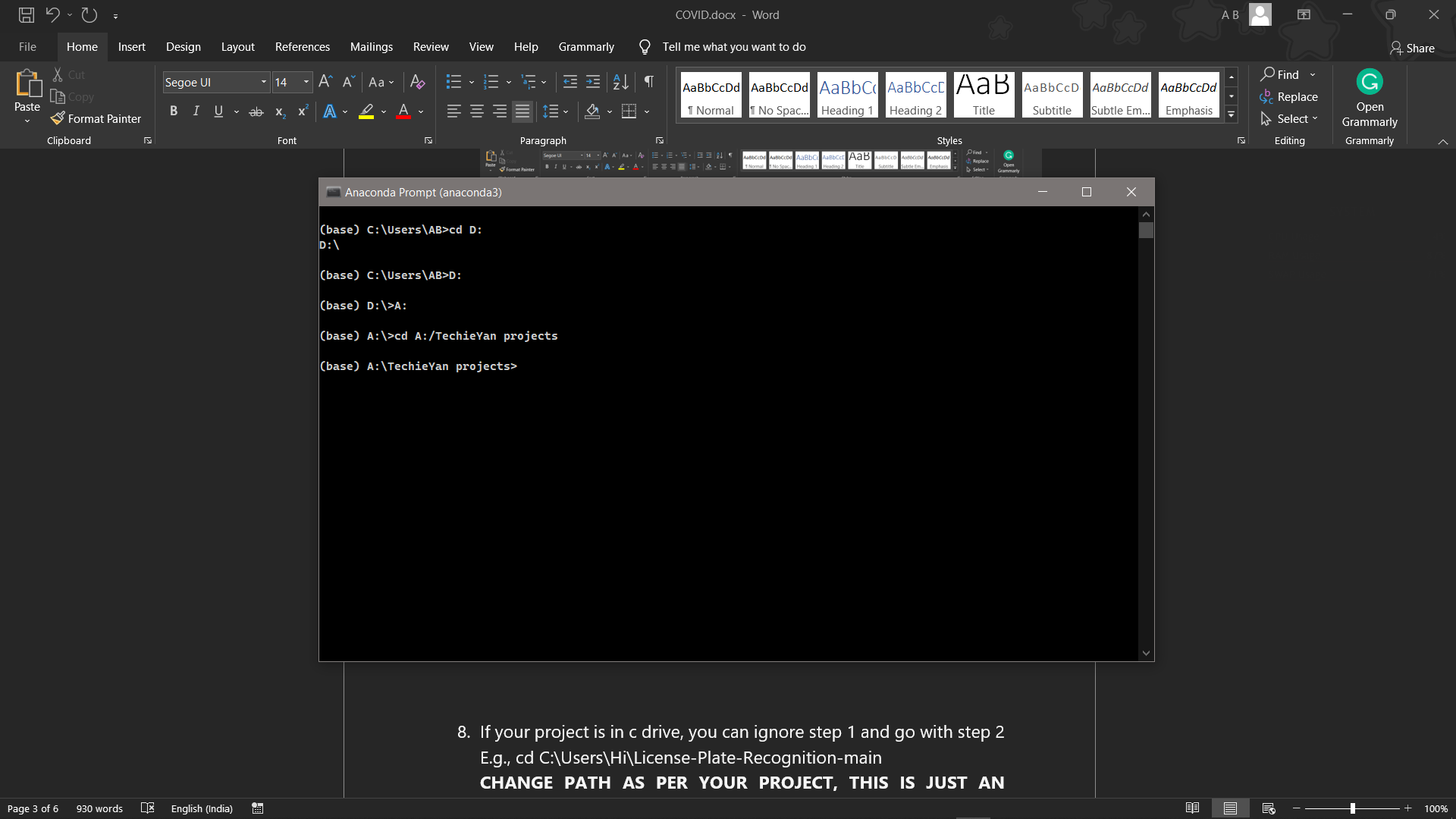
1. Install the prerequisites/software’s required to execute the code from reading the above blog which is provided in the link above.
2. Press windows key and type in anaconda prompt a terminal opens up.
3. Go to the directory where your requirement.txt file is present, not just requirement.txt, if you 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

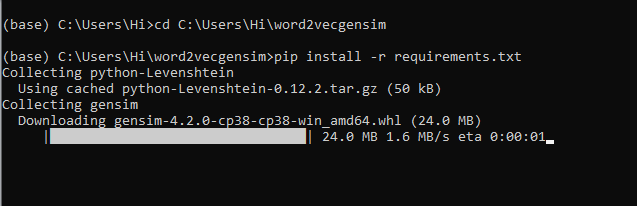


1. If your project is in c drive, you can ignore step 4 and 5.

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. If you would like to run .ipynb file, After the above steps type jupyter notebook in your terminal, 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, 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/), If you face any trouble opening jupyter notebook.

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

***Dataset Description***

The Dataset is collected form Kaggle Repository which contains 501 Instances with 4 features. The 4 features namely which contribute more to the model are gender, weight, height and index(target\_class). The target\_class(index) in categorised into 6 classes such as

1. Extremely Weak -> 0
2. Weak –> 1
3. Normal -> 2
4. Overweight -> 3
5. Obesity -> 4
6. Extremely Obese –> 5

This study's goal is to predict which category of weight does a person belong to:



***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.**

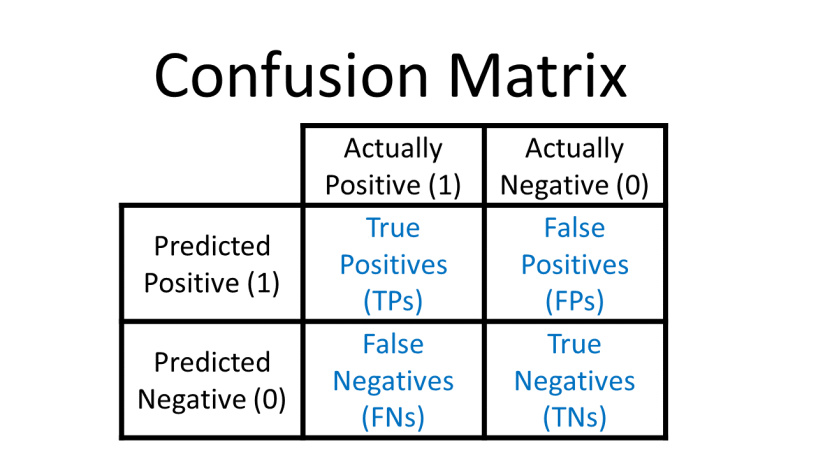
<https://www.linkedin.com/in/abhinay-lingala-5a3ab7205/>

***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:***



***Classification Report:*** Classification report helps us to understand and evaluate how good the model is performing which consists of different evaluation metrics, such as

**Precision:** This will count number of correct predictions of a single class divided by the total number of observations of the same class.

Precision = TP/(TP + FP)

**Recall:** Recall is calculated by counting the actual number of classes in a single class divided by total number of observations of that particular class.

Recall = TP/(TP+FN)

F1\_score: This is a Harmonic average of Precision and recall.

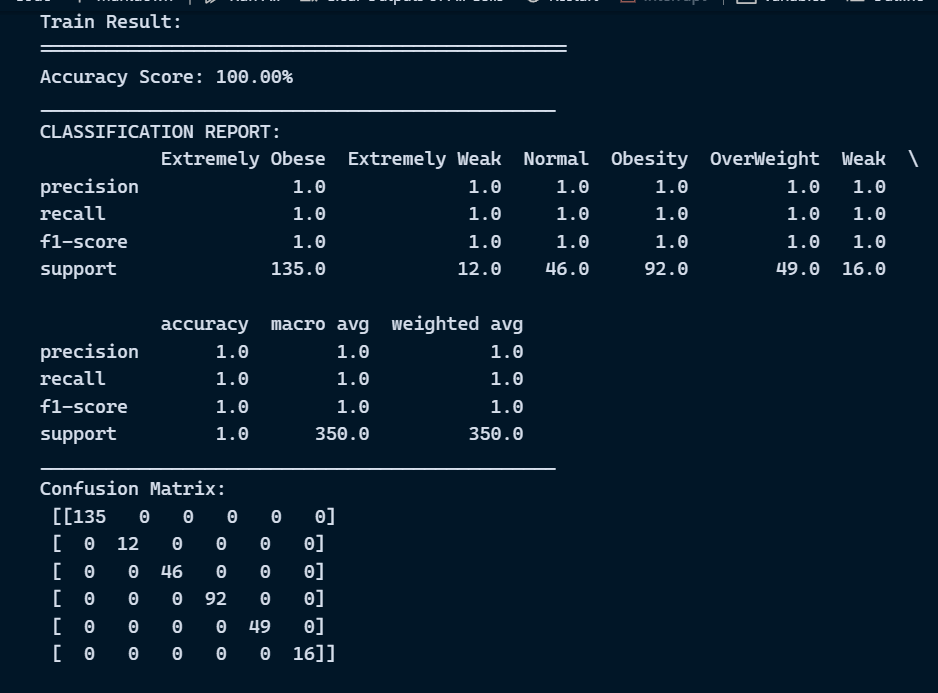
F1 Score = 2\*(Recall \* Precision) / (Recall + Precision)

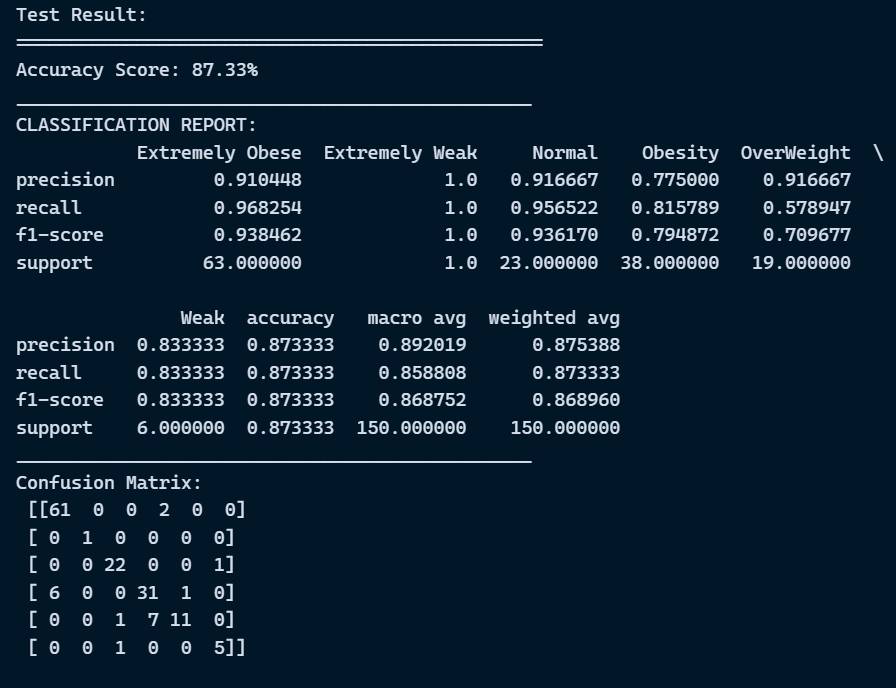
***Reference:***

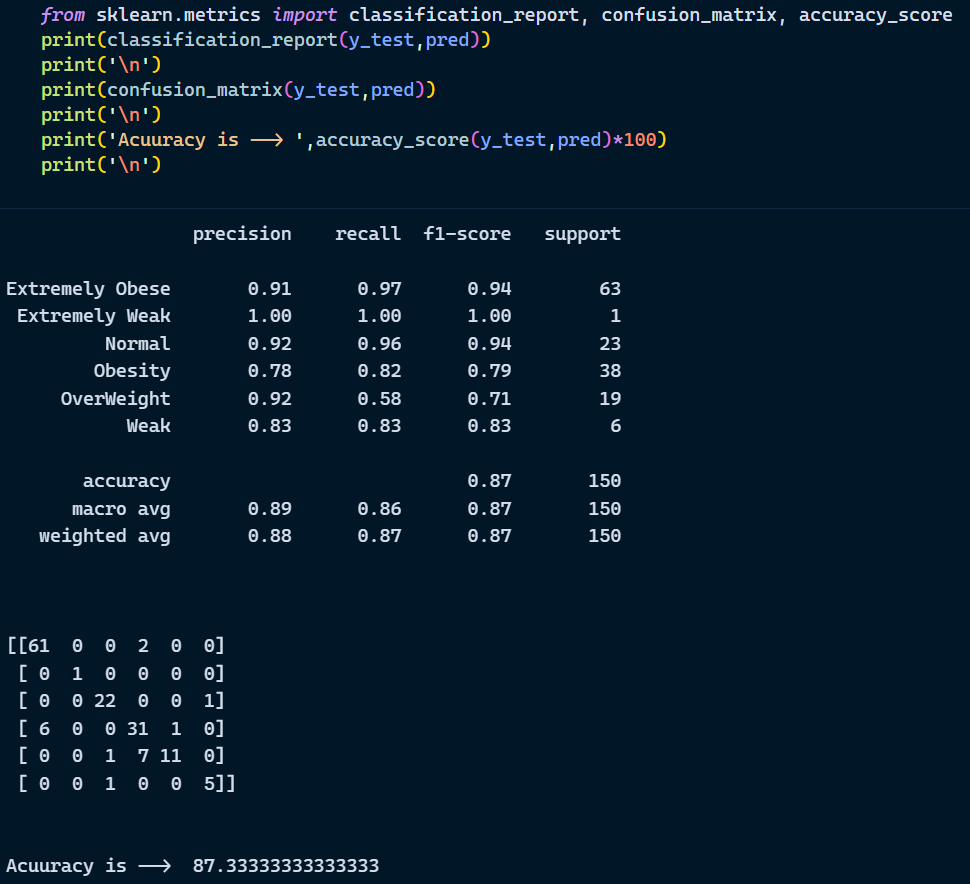
1. <https://scikit-learn.org/stable/modules/generated/sklearn.metrics.classification_report.html>
2. <https://www.geeksforgeeks.org/confusion-matrix-machine-learning/>

***Results***

1. ***Random Forest Classifier***

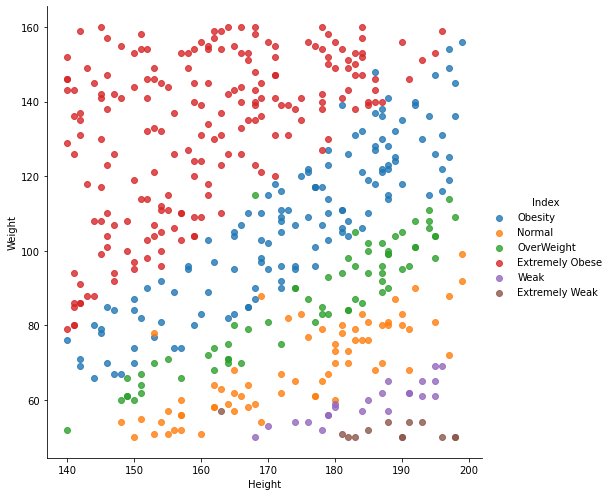


******

******

***Exploratory Data Analysis***

***Scatter Plot***

******