**MLDP Project Report**

**i. Cover Page**

STUDENT NAME: CHEN XIN

MATRIX NO: 2305105E  
CLASS: PC04

**ii. Introduction**

The project focuses on developing a machine learning model to predict obesity levels using various health and lifestyle factors. The dataset contains information about individuals' physical characteristics, dietary habits, physical activity, and other relevant health indicators. The primary goal is to create a predictive model that can classify an individual's obesity level based on their personal health profile.

**Overview:**  
This dataset include data for the estimation of obesity levels in individuals from the countries of Mexico, Peru and Colombia, based on their eating habits and physical condition.  
The data contains 17 attributes and 2111 records, the records are labelled with the class variable NObesity (Obesity Level), that allows classification of the data using the values of Insufficient Weight, Normal Weight, Overweight Level I, Overweight Level II, Obesity Type I, Obesity Type II and Obesity Type III.

**Data Details:**

* Gender: Gender
* Age: Age
* Height : in metres
* Weight : in kgs
* family\_history : Has a family member suffered or suffers from overweight?
* FAVC : Do you eat high caloric food frequently?
* FCVC : Do you usually eat vegetables in your meals?
* NCP : How many main meals do you have daily?
* CAEC : Do you eat any food between meals?
* SMOKE : Do you smoke?
* CH2O : How much water do you drink daily?
* SCC : Do you monitor the calories you eat daily?
* FAF: How often do you have physical activity?
* TUE : How much time do you use technological devices such as cell phone, videogames, television, computer and others?
* CALC : How often do you drink alcohol?
* MTRANS : Which transportation do you usually use?
* Obesity\_level (Target Column) : Obesity level

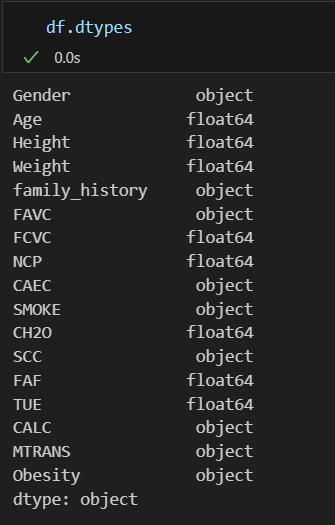
**iii. Data Exploration and Pre-processing**

I loaded the dataset using pandas and used .head(), .describe(), .isnull() to explore the first few records, statistical summaries, and check for missing values.

# Load the data

df = pd.read\_csv('Obesity\_prediction.csv')

df.head()



I then use df.types in order to see the dataset that has to be transformed or encoded.

Here is what I have come up with the plan on changing

Age = round.

gender = binary.

smoke = binary.

family history = binary

FAVC = binary

OBESITY = numeric

CAEC = one hot encoding

SCC = binary

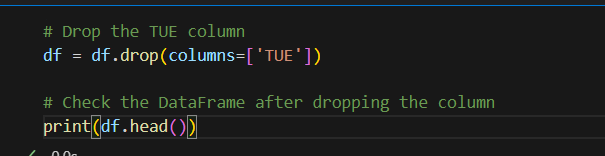
CALC = one hot encoding

MTRANS = one hot encoding

TUE = drop

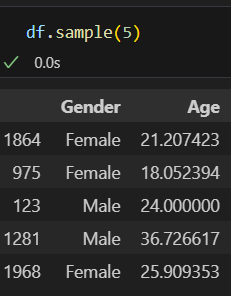
1. TUE (Drop)

The reason why I drop TUE is because this analysis is related to health, fitness and obesity related factors. The main purpose of TUE is about the time spend on electronic device which might not provide valuable information compared to other variables like physical activities, eating habits or family history. Therefore, I have decided to drop TUE.



1. Age (Round)

I have noticed from the dataset for the age it has decimals. Therefore, I have use round to make it in whole number.

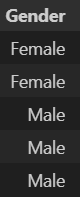
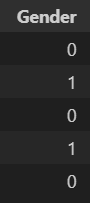






1. Gender (binary)

I have converted the gender into binary numerical format where Male = 1 and Female = 0.



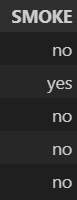
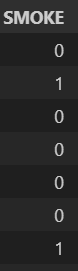
BEFORE

AFTER

1. SMOKE (binary)

I have converted the smoke into binary numerical format where yes = 1 and no = 0.



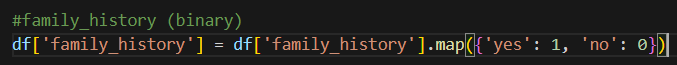


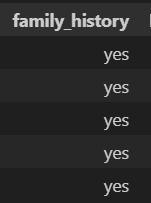
AFTER

BEFORE

1. Family\_history (binary)

I have converted the family\_history into binary numerical format where yes = 1 and no = 0.



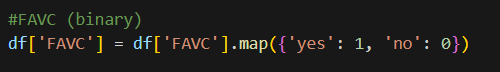


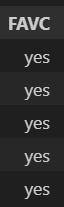
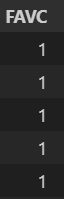
AFTER

BEFORE

1. FAVC (binary)

I have converted the FAVC into binary numerical format where yes = 1 and no = 0.



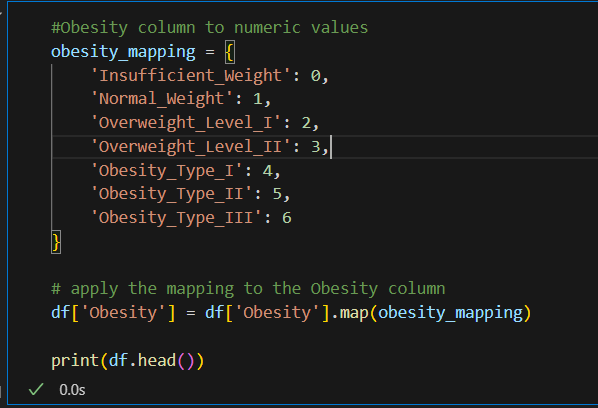


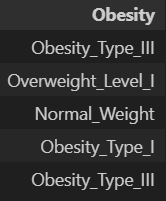
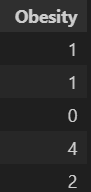
BEFORE

AFTER

1. Obesity (numeric)

I have changed obesity into numeric eg. Insufficient\_weight = 0, normal\_weight = 1.



****

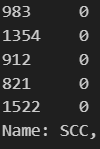
AFTER

BEFORE

1. SCC (binary)

I have converted the SCC into binary numerical format where yes = 1 and no = 0.

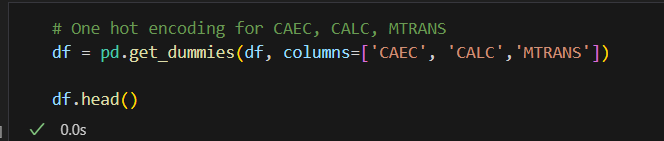


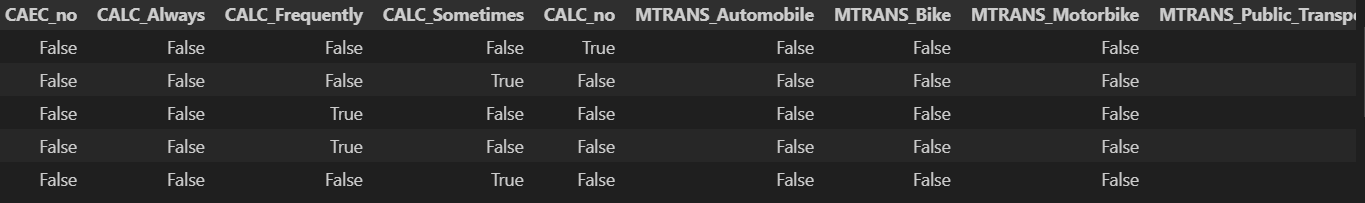
****

AFTER

BEFORE

1. One Hot Encoding (CAEC, CALC, MTRANS)

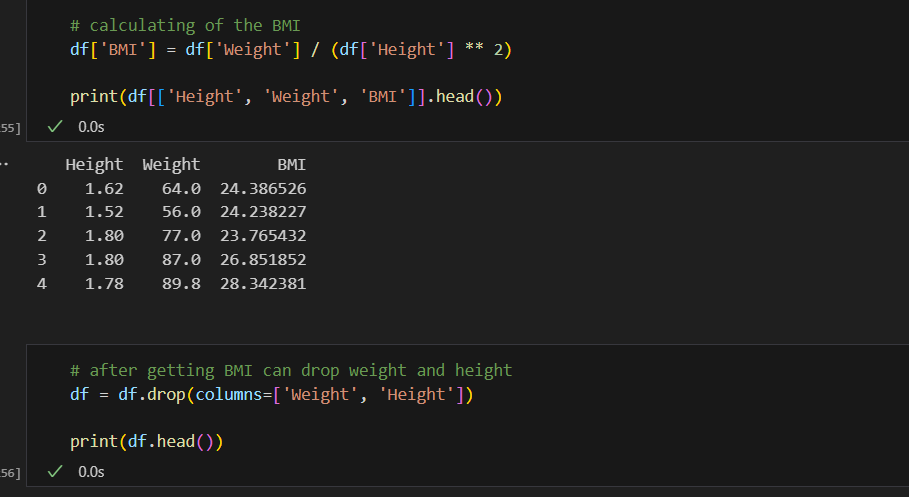


****

1. BMI (Weight and Height)

I have converted weight and height into BMI instead as I feel that it is a more convenient method to see whether the individual has a healthy body weight.

Therefore, I would then drop the weight and height as we do not need it anymore.

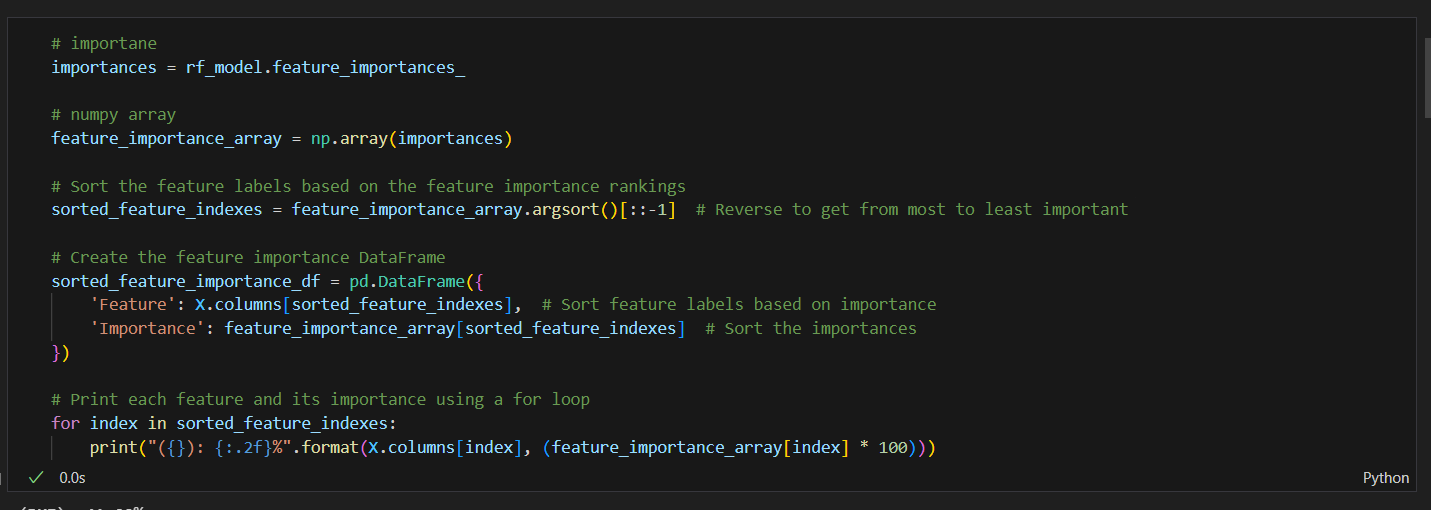


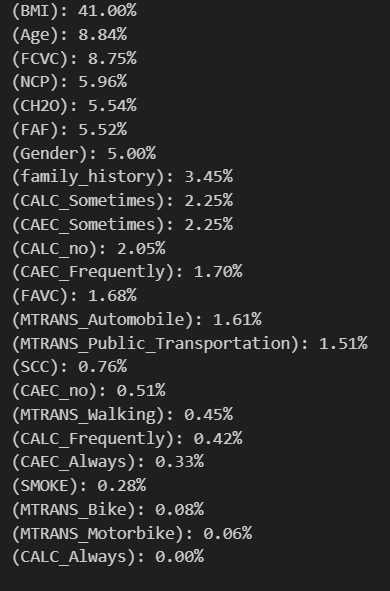
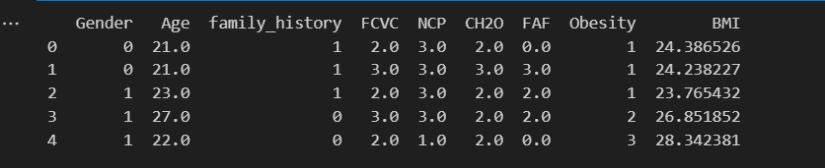
**iv. Methods and Improvements**

1. **Importance**

I feel that feature importance would be good to use here as I would be able to improve the model. Firstly, it can help to remove any unimportant features and reduces the complexity of the model. Secondly, it can prevent overfitting by only focusing on important features. Lastly, it can help to improve the accuracy by using important features which results in better performance.

After using importance, I can notice that there are some columns have very low importance which means that it is not contributing a lot to the prediction accuracy. Therefore, I would remove them.



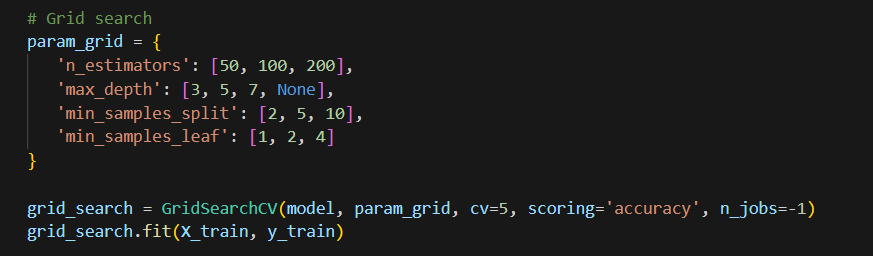


Through this, I will be keeping Gender, FAF, CH20, NCP, FCVC, Age and BMI only and I

will drop the rest.

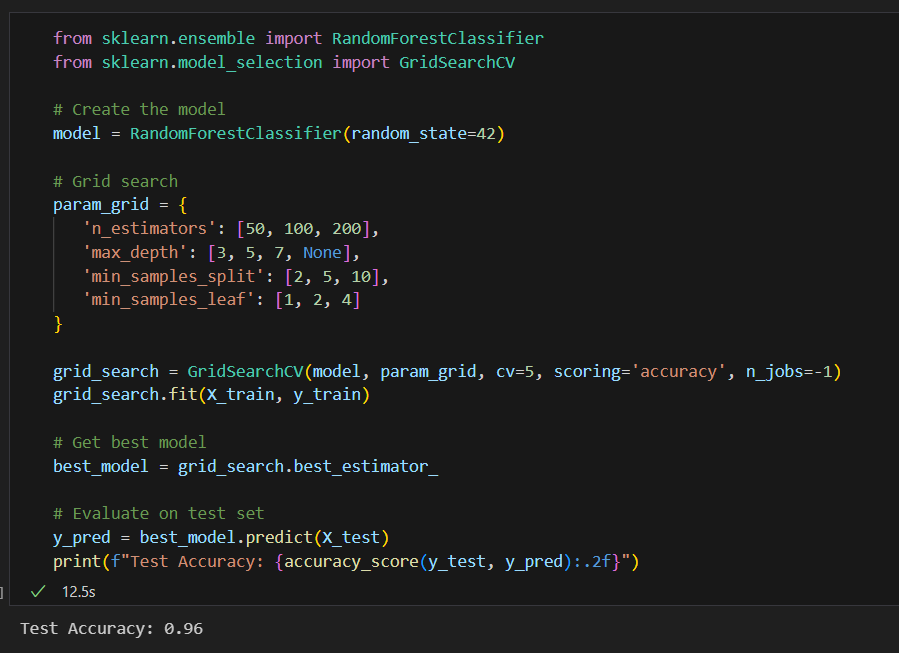
1. **Hyper Parameter Tuning**

I have applied hyperparameter tuning to optimize the performance of the classifier. It controls the model behaviour during the training process and their values would affect the model accuracy. Instead of selecting the hyperparameter manually, I used grid search to automate the process and to search for the best combinations.

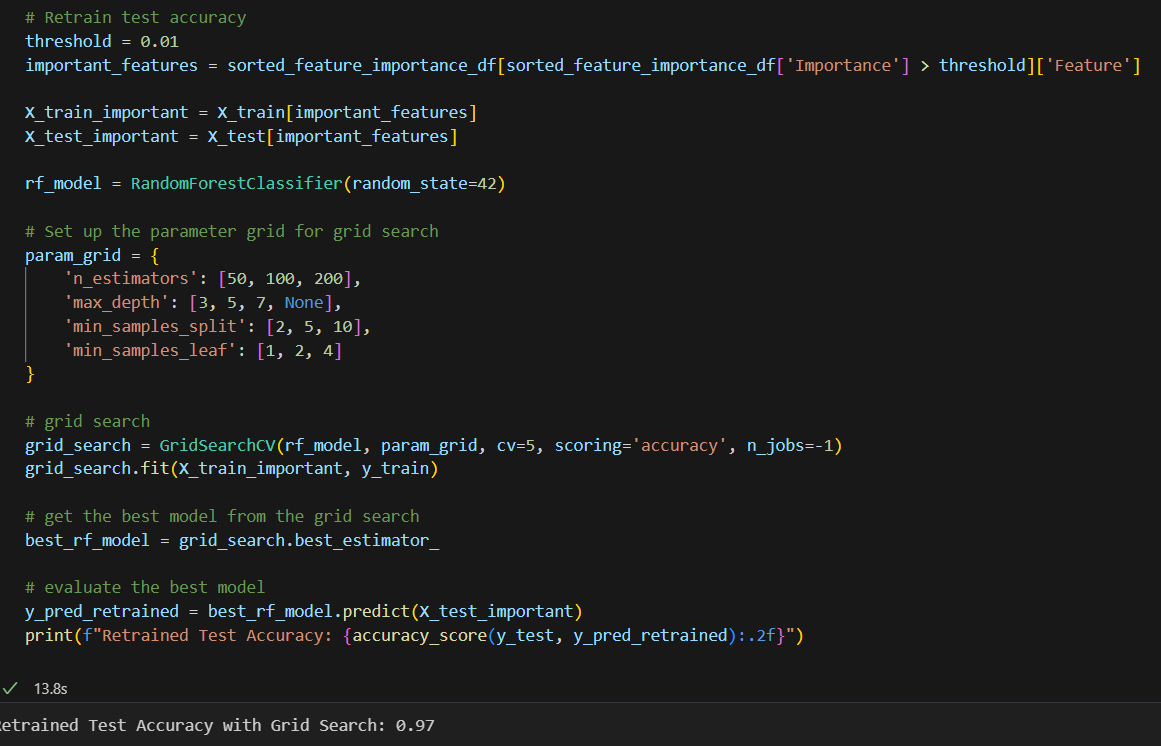


**v. Results and Analysis**

**Random Forest Classifier**



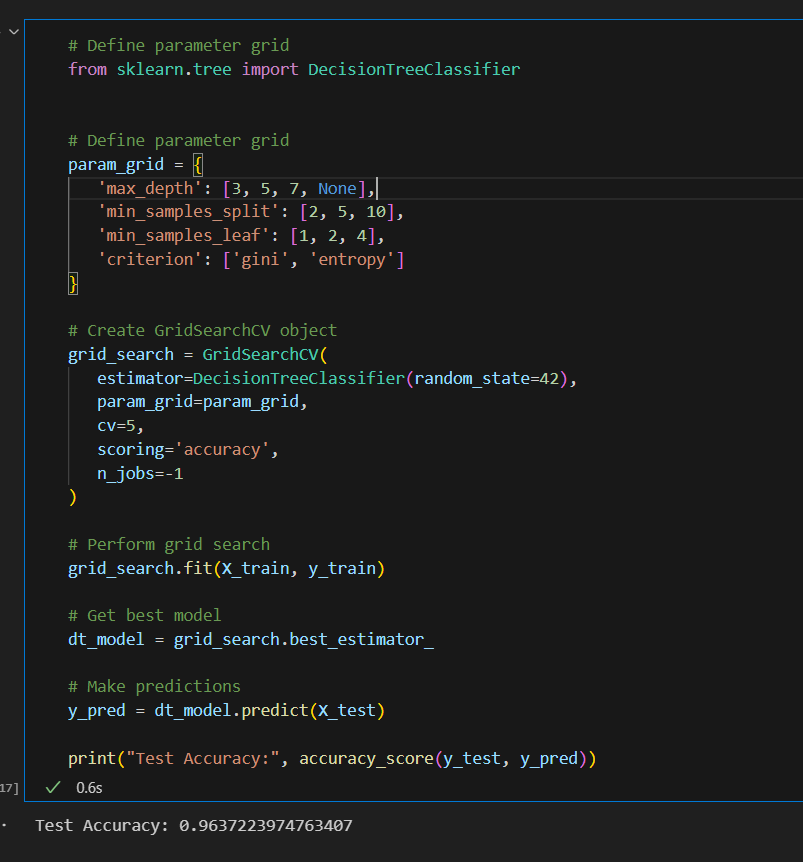
Original



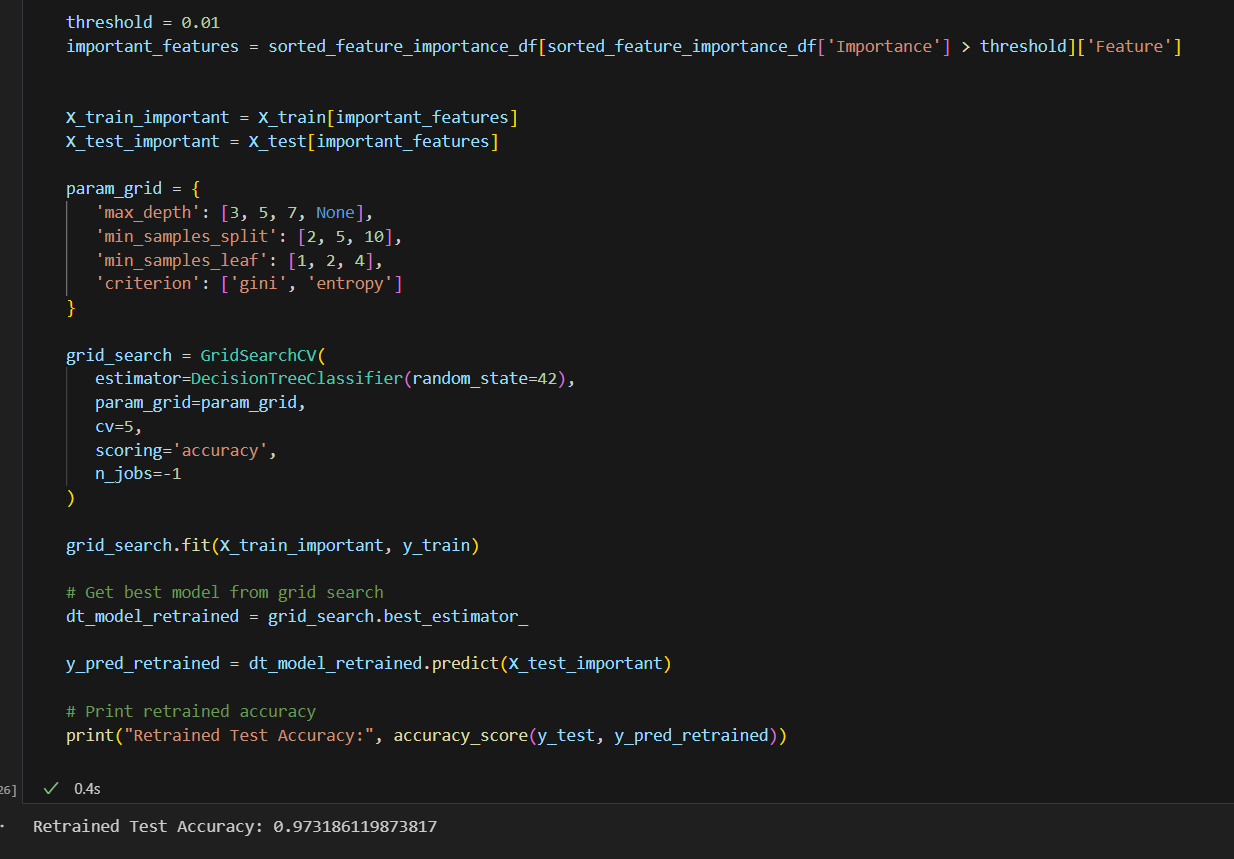
Retrained model

From the Random Forest Classifier, I observed that the retrained model, which focuses only on features with importance, achieves a higher accuracy compared to the original model that uses all features. Both models use grid search for hyperparameter tuning, but the retrained model benefits from reduced complexity and focuses on the most relevant features, improving performance and efficiency.

**Decision Tree classifier**



Original

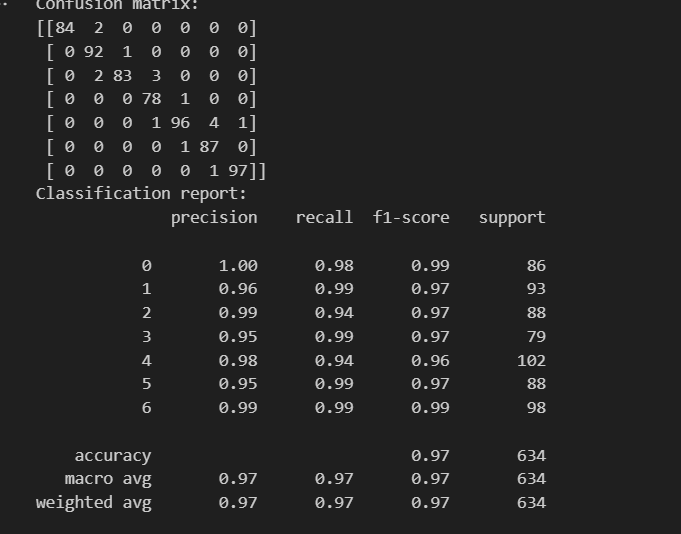


Retrained model

When comparing the decision tree classifier to the random forest classifier, I have observed that the decision tree has a higher accuracy score. While both models used hyperparameters ensuring that the best configuration was used for each. I could conclude that the decision tree would be a better choice.

**Confusion Matrix**

The confusion matrix shows how well my model predicts each class compared to the actual labels. Each of the row would represents the true class, and each column represents the predicted class.



* Class 0 and Class 6 have near perfect predictions.
* Class 4 has some confusion, with 1 misclassified as Class 3 and 4 misclassified as Class 5. This shows that the model struggles slightly to distinguish between these neighbouring classes.
* The overall distribution indicates strong performance, with most of the predictions concentrated along the diagonal.

METRICS:

1. **Precision**:

* For Class 0: Precision = 1.00 (100% of predicted Class 0 samples were correct).
* For Class 4: Precision = 0.98 (98% of predicted Class 4 samples were correct).

1. **Recall**:

* For Class 0: Recall = 0.98 (98% of true Class 0 samples were correctly identified).
* For Class 4: Recall = 0.94 (94% of true Class 4 samples were correctly identified).

1. **F1-Score**:

* Most classes have F1-scores close to or above 0.96, indicating high precision and recall.

1. **Support**:

* Class 0 has 86 samples, Class 1 has 93.

PERFORMANCE:

1. **Accuracy**: The model achieved an overall accuracy of 97%, meaning 97% of all predictions were correct.
2. **Macro Avg**: Average performance across all classes without considering class imbalance. Here, it’s 0.97 for precision, recall, and F1-score.
3. **Weighted Avg**: Average performance weighted by the number of samples in each class. This is also 0.97, indicating strong consistency across all classes.

**vi. Conclusion**

In this project, I have worked on a database related to obesity prediction where I have taken various steps to build and evaluate a machine learning model. Starting with data cleaning and preprocessing, we ensured that the dataset was prepared for analysis by handling missing values and scaling. Afterwards, I have used methods like feature importance and hyperparameter tuning to optimize the model and improve its accuracy. I then test out different models such as random forest and decision tree classifiers to demonstrate robust performance as evidence by metrics such as precision, recall, accuracy and confusion matrix. This project not only has highlighted the importance of systematic machine learning workflows but also provided a clear understanding of how model improvements can be implemented.

LINK: <https://mldp-project-axynuaecue2acyw4jjt8km.streamlit.app/>

**vii. References**

1. TP LMS
2. https://www.kaggle.com/datasets/ruchikakumbhar/obesity-prediction/data