# Soumya Mukherjee | CH24M571 |

# Lab Report Question 1

**1.0 Preface**

The question 1 of the assignment asks us to do the following on the diabetes data set :

1 . Fit a decision tree classifier

2. Optimise the tree by tuning parameters like criterion, max\_depth, and min\_samples\_leaf

3. Check the accuracy of the normal and tuned tree and do the confusion matrix

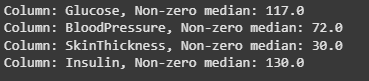
**2.0 Methodology :**

The dataset was loaded from the diabetes.csv file. It contains features such as Pregnancies, Glucose, BloodPressure, SkinThickness, Insulin, BMI, DiabetesPedigreeFunction, and Age, with the target variable being Outcome (1 for diabetic, 0 for non-diabetic).

**2.0.1 Data Pre-processing :**

The dataset was checked for null and NaN values . We visualized the data using histograms which indicated there were ‘0’ values for certain features which could not be true in real life .

This values were handled by imputation (median for numerical features).



The before and after statistics were calculated , The mean and standard deviation changed for the columns above .

Data was split in 70:30 as asked in the question

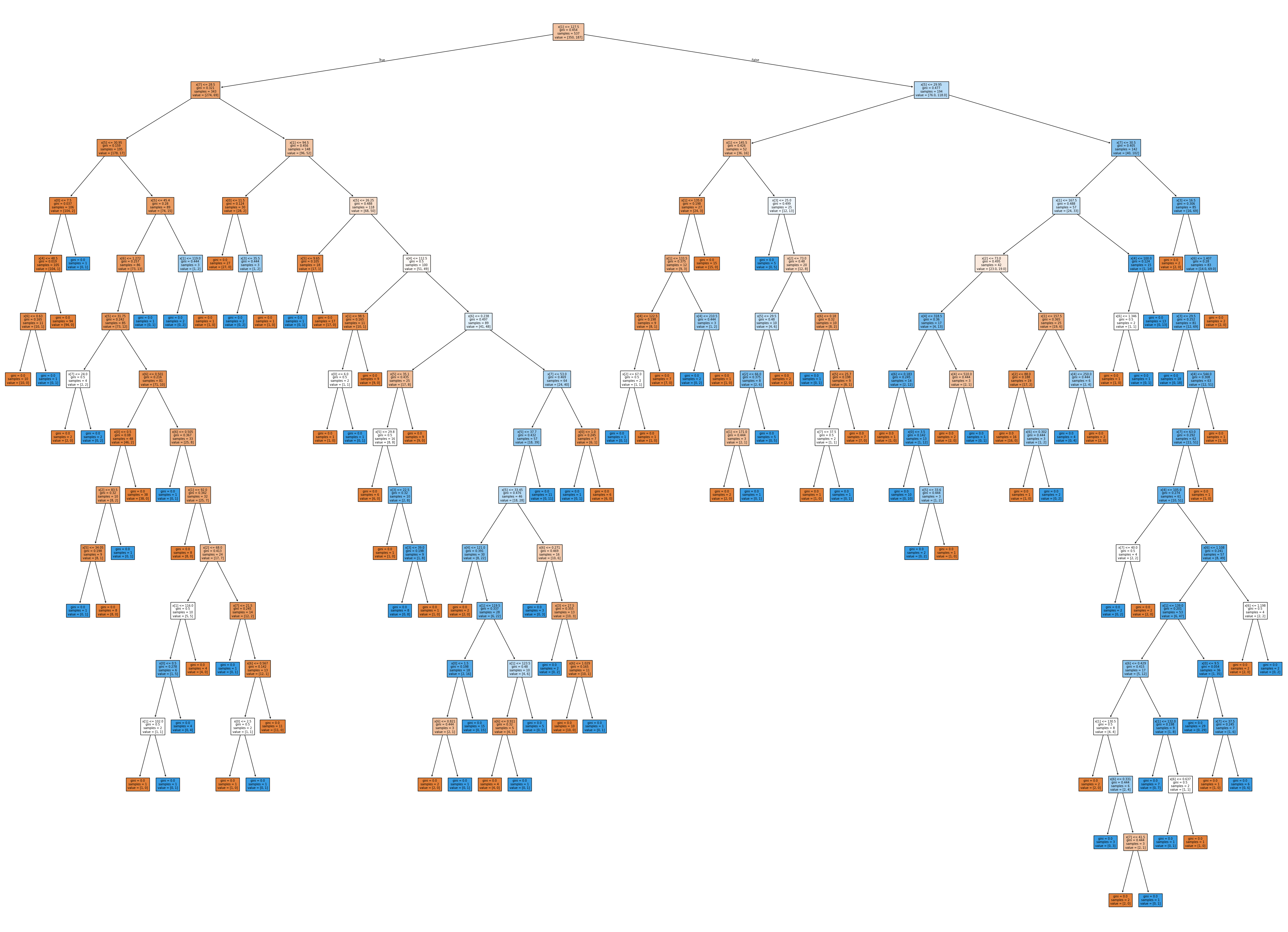
**2.0.2 Model Implementation**

**2.0.2a**

A decision tree classifier was implemented using DecisionTreeClassifier with default parameters. The model was trained on the imputed training set and the accuracy of the model was evaluated on the test set.

**2.0.3 Visualisation**

The tree was visualised using plot\_tree from sklearn . Font-size was kept to 7 .



The accuracy of the model without any tuning came upto be

* Training : 100 %
* Testing : 77.05%

**2.0.2b**

The performance of the decision tree was optimized using GridSearchCV to tune the following parameters:

Criterion: Tested gini index and entropy.

max\_depth: Tested different depths to avoid overfitting.

min\_samples\_leaf: Tested different values to control the minimum number of samples required to be at a leaf node.

The best parameters found were {'criterion': 'entropy', 'max\_depth': 5, 'min\_samples\_leaf': 20}**.**The accuracy of the optimised model rose to 79.22%

**2.0.2c**

It was noticed that there were much more 0 Outcomes in the data set as compared to 1 . 500 vs 268 . We performed SMOTE ( Synthetic Minority Over-sampling technique ) on the hyper parameter tuned model to address this .

We used a pipeline from imblearn package instead of sklearn , to account for the SMOTE .

We used 3 , 5 and 7 from the probable K-neighbours as the parameters for SMOTE and gini and entropy as crietrions . Max depth and min samples leaf were same as above .

The decision tree was retained on the SMOTE augmented dataset .

**3.0 Evaluation**

Two parameters were used for evaluation

* Accuracy: The accuracy of the optimized model was calculated on the test set.
* Confusion Matrix and Classification Report: The confusion matrix and classification report were generated to evaluate the model's performance in terms of precision, recall, and F1-score.

**4.0 Results and Conclusion**

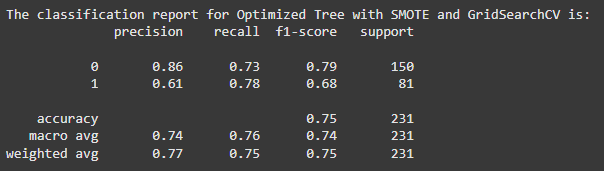
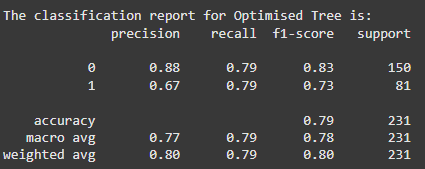
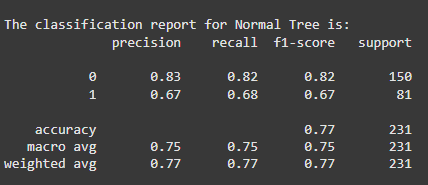
The accuracy of the decision tree model with default parameters was 77.06% on the test set.

And that of Hyperparameter - tuned tree was 79.22% with best combination of parameters being

* criterion: entropy
* max\_depth: 5
* min\_samples\_leaf: 20

For the SMOTE tuned tree the accuracy dropped to 74.89%

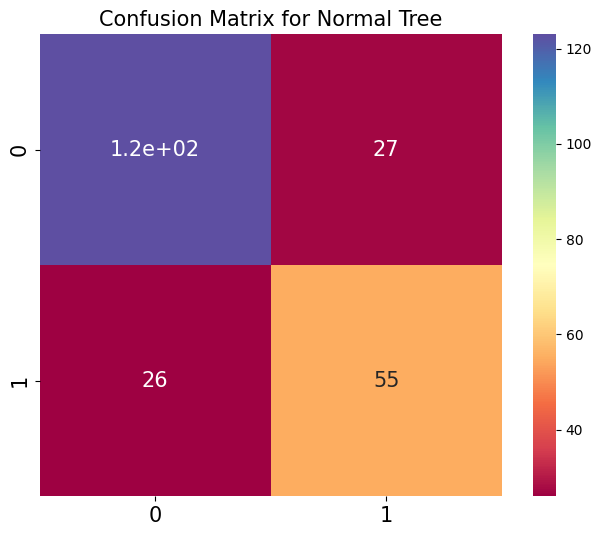
The classification reports are as follows :



Applying SMOTE to handle class imbalance resulted in a slight decrease in accuracy (74.89%), but it improved the recall for the minority class (diabetic patients), indicating better performance in identifying positive cases.

The classification reports show that the optimized model (without SMOTE) achieved a better balance between precision and recall compared to the default model.

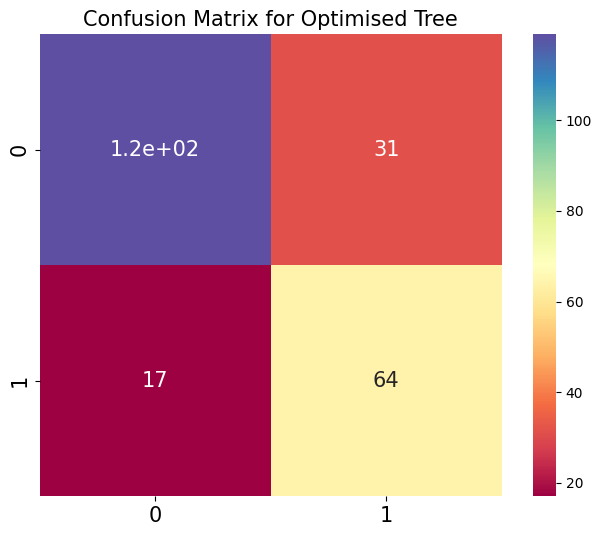
Based on the confusion matrices , we can have the following conclusions :



The default model has a high number of false negatives (55), indicating poor performance in identifying diabetic patients.

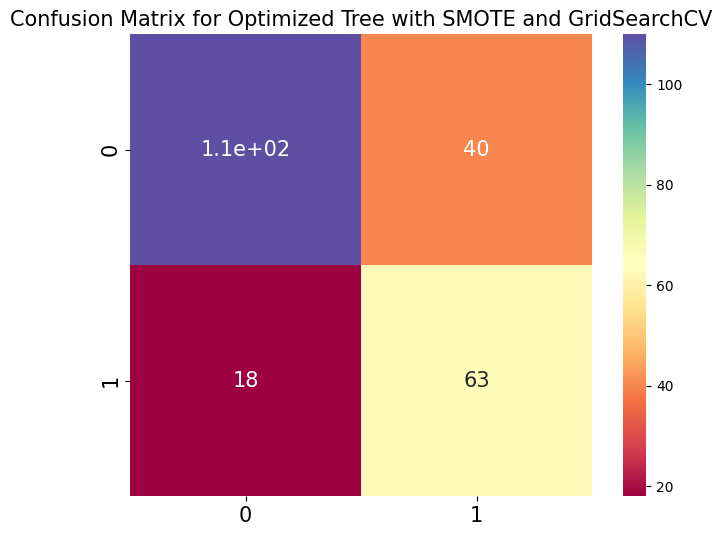
The overall accuracy is 77.06%, but the model is biased toward the majority class (non-diabetic patients).

Optimized Decision Tree (GridSearchCV):



Hyperparameter tuning significantly improved the model's performance. The number of false negatives (FN) decreased from 55 to 31, and the number of true positives (TP) increased from 25 to 64. The overall accuracy improved to 79.22%, and the model achieved a better balance between precision and recall for both classes.

Optimized Decision Tree with SMOTE and GridSearchCV:



Applying SMOTE further improved the model's ability to identify diabetic patients, reducing false negatives (FN) to 18. However, this came at the cost of increased false positives (FP), which rose to 40. The overall accuracy decreased slightly to 74.89%, but the model's recall for the minority class (diabetic patients) improved significantly.

# Lab Report Question 2

**1.0 Preface**

The question 2 of the assignment asks us to do the following :

* Classify the dataset using Logistic regression and find summary statistics
* Implement the LDA and QDA and evaluate the model performance
* Implement stochastic gradient descent classifier from scratch

**2.0 Methodology :**

**2.0.1 Data Pre-processing :**

**2.0.2 Feature Engineering :**

**Feature Importance Analysis:**

**Summary Statistics for Important Features**:

.

**3.0 Results and Analysis**

**Conclusion:**