**Experiment – 05**

Name: Akash Kumar

PRN: S1032233339

**Aim**: Implement Decision tree to classify the given data set.

**Objective**: Develop practical skills in implementing and evaluating Decision Tree classifiers for given dataset using Python on Jupyter Notebook.

**Implement an application of Decision Tree algorithm:**

DataSet: Indians Diabetes from kaggle.

<https://www.kaggle.com/datasets/uciml/pima-indians-diabetes-database>

This dataset contains health conditions and different parameters for get analysis from data.

**Theory:**

**Decision Trees:** Decision trees are a supervised learning algorithm used for both classification and regression tasks. They work by recursively splitting the data based on feature values to make decisions. Here's a brief overview of the theory behind decision trees:

🡪Tree Structure: A decision tree is a hierarchical structure consisting of nodes and branches. Each internal node represents a decision based on the value of a particular feature, and each leaf node represents the predicted class or value.

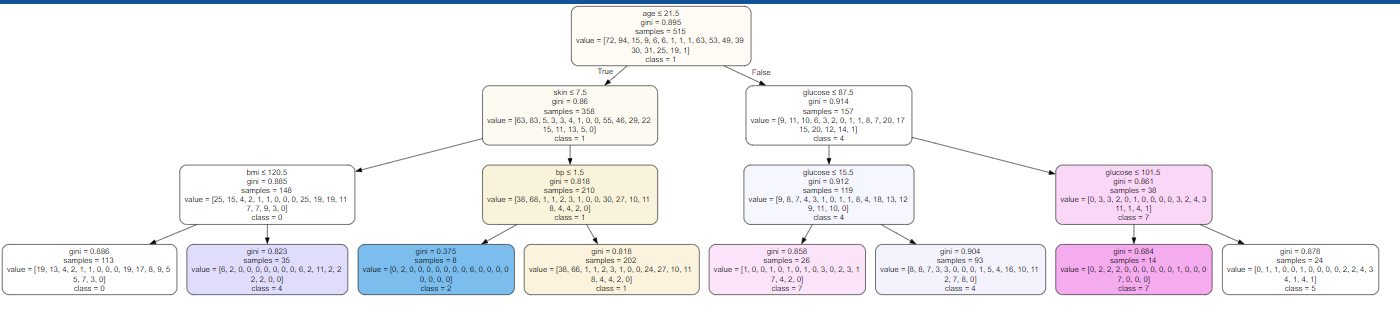
🡪Splitting Criteria: Decision trees split the data at each node based on a criterion that maximizes the homogeneity of the target variable within each subset. Common splitting criteria include Gini impurity and information gain (entropy).

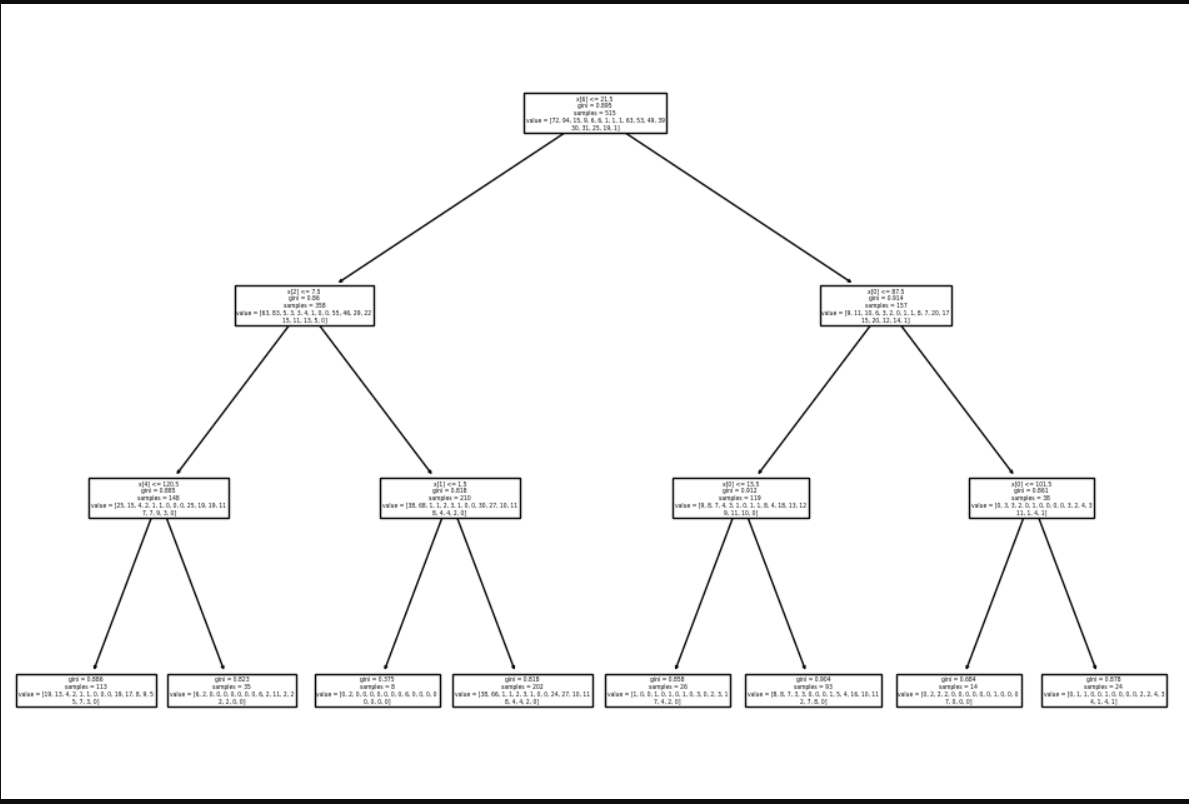
🡪Building the Tree: Decision trees are typically built using a top-down, greedy approach. At each step, the algorithm selects the best feature to split the data and recursively builds the tree until a stopping criterion is met (e.g., maximum depth, minimum samples per leaf).

🡪Pruning: To prevent overfitting, decision trees can be pruned after construction. Pruning involves removing nodes from the tree that do not significantly improve its predictive performance on a validation set.

**Code:** <https://github.com/Akash09kumar/Decision-Tree-algorithm/blob/main/DT_diabetes.ipynb>

**Result**

****

****

**Conclusion:**

1. In this experiment, we build a Decision-Tree Classifier model to predict the diabetes. We build two models, one with criterion gini index and another one with criterion entropy.
2. In the model with criterion gini index, the training-set accuracy score is 0.2641 while the test-set accuracy to be 0.2008. These two values are quite comparable. So, there is no sign of overfitting.
3. In both the cases, the training-set and test-set accuracy score is the same. It may happen because of small dataset. In my dataset Model Accuracy, how often is the classifier correct was checked and accuracy came to be 0.086614.
4. The confusion matrix and classification report yield good model performance.

**GitHub Link:** <https://github.com/Akash09kumar/Decision-Tree-algorithm>