# CA03 – Decision Tree Algorithm

**Q.1.1 Why does it makes sense to discretize columns for this problem?**

Every split in a decision tree is based on a feature. If the feature is categorical, the split is done with the elements belonging to a particular class. If the feature is continuous, the split is done with the elements higher or lower than a threshold. At every split, the decision tree will take the best variable at that moment. Decision trees categorize continuous variables by creating binary regions with the threshold.

Thus, it makes sense to discretize the continuous data columns in the dataset.

**Q.1.2 What might be the issues (if any) if we DID NOT discretize the columns.**

Since, we are building a CART classification decision tree model which works only for the categorical dataset. While there is another CART Regression tree which works for the continuous data as well. But in this assignment, we have to do a classification of people predicting whether their income will be greater or less than 50k. Therefore, we will convert our categorical columns into bins.

CART and C4.5 supports the processing of data features as continuous variables by using binary segmentation to process continuous variables. If the feature value is greater than the split value, the left tree is taken or the right

**Q.7.1 Decision Tree Hyper-parameter variation vs. performance**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Decision Tree Hyperparameter Variations Vs. Tree Performance** | | | | | | | |
| **=============== Complete the following table ==============** | | | | | | | |
|  | | | | | | | |
| **Hyperparameter Variations** | | | | **Model Performance** | | | |
| **Split Criteria (Entropy or Gini)** | **Minimum Sample Split** | **Minimum Sample Leaf** | **Maximum Depth** | **Accuracy** | **Recall** | **Precision** | **F1 Score** |
| ***Entropy*** | ***7*** | ***5*** | ***10*** | **0.84** | **0.57** | **0.71** | **0.63** |
|  | ***10*** | ***10*** | ***15*** | **0.84** | **0.57** | **0.7** | **0.63** |
|  | ***20*** | ***15*** | ***6*** | **0.83** | **0.59** | **0.68** | **0.63** |
|  | ***15*** | ***20*** | ***7*** | **0.841** | **0.57** | **0.7** | **0.63** |
|  |  |  |  |  |  |  |  |
| ***Gini Impurity*** | ***7*** | ***5*** | ***10*** | **0.84** | **0.57** | **0.71** | **0.63** |
|  | ***10*** | ***10*** | ***15*** | **0.83** | **0.56** | **0.7** | **0.62** |
|  | ***20*** | ***15*** | ***6*** | **0.84** | **0.59** | **0.69** | **0.63** |
|  | ***15*** | ***20*** | ***7*** | **0.84** | **0.59** | **0.7** | **0.64** |

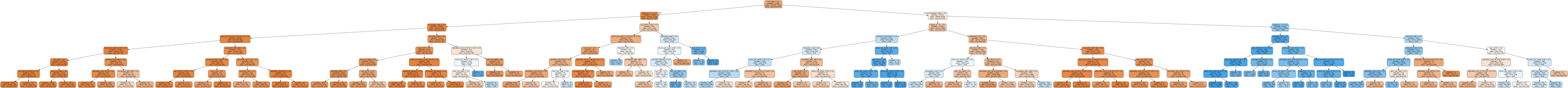
**Q.8.1 How long was your total run time to train the model?**

Total run time **-** 89 ms

**Q.8.2 Did you find the BEST TREE?**

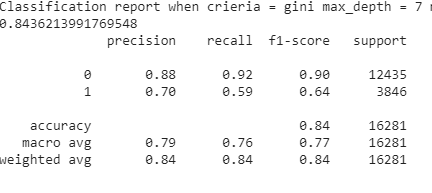
The hyperparameters entered above are tuned and we select the tree with best accuracy, f score, precision and recall. The best tree is highlighted in yellow in the above table

**Q.8.3 Draw the Graph of the BEST TREE Using GraphViz**



**Q.8.4 What makes it the best tree?**

The best decision tree is decided on the basis of several performance metrics score like accuracy, precision, recall, f1 score, etc. Below are the performance metrics score I got for the best tree selected.



**Q.10.1 What is the probability that your prediction for this person is accurate?**

The prediction is accurate up to 84% when we predict the income of this person with the best decision tree built. The income predicted for this person is more than 50K.