HW #3

In this homework, you have to implement the decision tree algorithm. Please use the following data table, where each row is one data point. The last column is class label, and the first two columns are features.

[[2.2343124,1.123123,0],

[1.43523,1.54245,0],

[3.53467889,2.234987,0],

[3.1249876,2.09237512893,0],

[2.1238756,9.3253154,1],

[7.0981274,3.89074,1],

[1.129875,3.0987234,0],

[7.0897345,0.089745,1],

[6.0987214,3.0978214,1],

[6.1325,3.98763,1],

[1.35765,2.43663,0],

[2.345,3.3456,0],

[0.2345,1.4356,0],

[2.4356,5.67534,0],

[5.234,5.23465,1],

[4.12346,2.975,1],

[2.5467,4.72345,0],

[8.4612,1.6269,1],

[5.215690,2.5362,1],

[4.762,1.76567,1]]

While implementing the decision tree algorithm, you have to follow the following guidance:

1. Each node has (COLUMN, VALUE, DATA) and its left branch represents a subset of DATA that meet “COLUMN < VALUE” and the right branch represents “COLUMN >= VALUE”.
2. For the best node split (i.e., deciding the best COLUMN and the best VALUE), you have to use the Gini impurity. You should choose the minimum Gini impurity split. Gini impurity is also known as Gini index. They are the same.
3. If tie for the Gini impurity, when splitting a node, you should choose the COLUMN and the VALUE that lead to the most balanced split, i.e., the absolute different between size(DATA) of the left child and size(DATA) of the right child should be minimized. If tie again for the balance, you can choose any.
4. You should print the trained tree in the depth-first manner. Please refer to the lecture slides about this. You should fill out three functions, gini\_impurity, recursive\_split, and my\_tree in the released Jupyter Notebook. While implementing, you can define your own additional functions which you may want to call from the three main functions, but the key algorithm should be implemented in the three functions.
5. Your code should be enough flexible to consider max\_depth and min\_samples\_split. Please refer to the following web page describing the two hyperparameters: <https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html>.
6. You should submit your Jupyter Notebook runnable in Colab and a report describing your tests with various (max\_depth, min\_samples\_split) configurations. I require (max\_depth = 1, min\_samples\_split = 2), (max\_depth = 2, min\_samples\_split = 2), (max\_depth = 2, min\_samples\_split = 10), and (max\_depth = 3, min\_samples\_split = 2). You should put your depth-first trees in the report. I will share my result with (max\_depth = 1, min\_samples\_split = 2) as follows, which means the first column X1 and a threshold of 4.12346 had been chosen (you may see a different table because of the random tie breaking).

(X1, 4.12346)

(0)

(1)

Note that 4.12346 can be observed in our data table. So, when splitting a node, the candidates of VALUE are already in the table. The leaf nodes mean class labels.

1. Note that I do not require pruning and any other steps. You need to implement the tree growing algorithm only.
2. 100 pts if all correct. I will check the following points:
   1. -10 if the Gini impurity calculation is incorrect.
   2. -10 if you do not choose the best split following the guidance.
   3. -10 if you do not print in the depth-first manner.
   4. -10 if you do not support max\_depth.
   5. -10 if you do not support min\_samples\_split.
   6. -10 if an incorrect tree is built for each of (max\_depth = 1, min\_samples\_split = 2), (max\_depth = 2, min\_samples\_split = 2), (max\_depth = 2, min\_samples\_split = 10), and (max\_depth = 3, min\_samples\_split = 2).