

The DecisionTree class create a decision tree with class BinTree as a node, which has yesBranch and noBranch to represent left child node and right child node, respectfully. Each node has mainly the following attributes:

nodeID: the node's ID.

nodeType: the node's type. 0: the leaf node, 1: binary node, 2: categorical node, 3: continuous node.

nodeInfo: the node's information. Used to classify the records.

When we want to check a record's predict class label, simply draws a route from the root node to the leaf node. Check if the record satisfies the split node. If so, the record goes down to the left child node. Otherwise, it goes down to the right child node.

The decision tree can be printed to file as the following format:

```
capital-gain > 5013:
|   capital-gain <= 6849:
|   |   capital-gain <= 6514:
|   |   |   hours-per-week <= 38:
|   |   |   |   age <= 45: <=50K
|   |   |   |   age > 45: >50K
|   |   |   hours-per-week > 38: >50K
|   |   capital-gain > 6514: <=50K
|   capital-gain > 6849:
|   |   age <= 18: <=50K
|   |   age > 18:
|   |   |   age <= 60: >50K
|   |   |   age > 60:
|   |   |   |   age <= 62: <=50K
|   |   |   |   age > 62: >50K
```

The rows without a classification on the right side are the split nodes. The first row is the root node, then the second row is the first layer node and so forth. Nodes on the same layer will have the same indent.

The tree we built has an error rate of 0.2452 to classify the testing data. The report of the classification results is in the file "decision_tree_result.txt".