Gini Index

Gini Index calculates the amount of probability of a specific feature that is classified incorrectly when selected randomly. If all the elements are linked with a single class then it can be called pure.

It is calculated by subtracting the sum of the squared probabilities of each class from one.

The Gini index varies between values 0 and 1, where 0 expresses the purity of classification, i.e. all the elements belong to a specified class or only one class exists there. And 1 indicates the random distribution of elements across various classes. The value of 0.5 of the Gini Index shows an equal distribution of elements over some classes.

Implementation of Gini Index Calculation

```
import numpy as np

#define function to calculate Gini coefficient
def gini(x):
    total = 0
    for i, xi in enumerate(x[:-1], 1):
        total += np.sum(np.abs(xi - x[i:]))
    return total / (len(x)**2 * np.mean(x))
```

Example of using gini() function

```
#define NumPy array of income values
incomes = np.array([50, 50, 70, 70, 70, 90, 150, 150, 150, 150])

#calculate Gini coefficient for array of incomes
gini(incomes)

0.226
```

CART Algorithm

In the decision tree, the nodes are split into subnodes on the basis of a threshold value of an attribute. The CART algorithm does that by searching for the best homogeneity for the subnodes, with the help of the Gini Index criterion.

The root node is taken as the training set and is split into two by considering the best attribute and threshold value. Further, the subsets are also split using the same logic. This continues till the last pure sub-set is found in the tree or the maximum number of leaves possible in that growing tree. This is also known as Tree Pruning.