Machine Learning
Experiment 3

SAP ID : 60004200139   Name : Riya Bihani
Division : B   Batch : B1

AIM

To implement CART decision tree algorithm.

THEORY

CART (Classification and Regression Tree) is a variation of the decision tree algorithm. It can handle both classification and regression tasks. It is a predictive algorithm used in Machine learning and it explains how the target variable's values can be predicted based on other matters. It is a decision tree where each fork is split into a predictor variable and each node has a prediction for the target variable at the end.
In the decision tree, nodes are split into sub-nodes based on a threshold value of an attribute. 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.
CART algorithm uses Gini Impurity to split the dataset into a decision tree .It does that by searching for the best homogeneity for the sub nodes, with the help of the Gini index criterion.

<u>Gini index/Gini impurity</u>

The Gini index is a metric for the classification tasks in CART. It stores the sum of squared probabilities of each class. It computes the degree of probability of a specific variable that is wrongly being classified when chosen randomly and a variation of the Gini coefficient. It works on categorical variables, provides outcomes either "successful" or "failure" and hence conducts binary splitting only. The degree of the Gini index varies from 0 to 1.

$$Gini = 1 - \sum_{i=1}^{n} (pi)^2$$

where pi is the probability of an object being classified to a particular class.

<u>Advantages of CART</u>

- Results are simplistic.
- Classification and regression trees implicitly perform feature selection.
- Outliers have no meaningful effect on CART.

Disadvantages of CART

- Overfitting.
- High Variance.
- The tree structure may be unstable.

CODE

CART

```
import pandas as pd
import numpy as np

def variable_count(att):
 types = pd.unique(att)
 no_of_types = len(types)
 counts = att.value_counts()
 return no_of_types, counts, types

def gini_of_attribute(no_of_types, counts, rows, cla, types, att1, cl):
 gini_a = 0
 type_cl_count = 0
 type_count = 0
 gini = []
 div_index = 0

 if no_of_types == 2:
  for i in range(len(types)):
   temp = df.loc[df[att1.name] == types[i]]
   type_count = len(temp)
   p = 1
   for j in range(len(cla)):
    temp = df.loc[(df[att1.name] == types[i]) & (df[cl.name] == cla[j])]
    type_cl_count = len(temp)
    p -= pow((type_cl_count/type_count), 2)
   gini_a += (type_count/rows) * p

 elif no_of_types > 2:
  for i in range(no_of_types):
   temp1 = df.loc[df[att1.name] == types[i]]
   temp2 = df.loc[df[att1.name] != types[i]]
   type_count1 = len(temp1)
   type_count2 = len(temp2)
   p1 = 1
```

```
      p2 = 1
      for j in range(len(cla)):
        temp3 = df.loc[(df[att1.name] == types[i]) & (df[cl.name] == cla[j])]
        type_cl_count1 = len(temp3)
        p1 -= pow((type_cl_count1/type_count1), 2)
        temp4 = df.loc[(df[att1.name] != types[i]) & (df[cl.name] == cla[j])]
        type_cl_count2 = len(temp4)
        p2 -= pow((type_cl_count2/type_count2), 2)
      gini.append((type_count1/rows) * p1 + (type_count2/rows) * p2)
    gini_a = min(gini)
    div_index = gini.index(gini_a)
  return gini_a, div_index

df = pd.read_csv('CART.csv')
col = list(df.columns.values.tolist())

cl = df.iloc[ : , -1]
no_of_types, counts, cla = variable_count(cl)
rows = len(cl)
gini = 1 - pow((counts[0]/rows), 2) - pow((counts[1]/rows), 2)
print(gini)

gini_a = []
div = []
t = []
att = len(df.columns) - 1

for i in range(att):
  att1 = df.iloc[ : , i]
  no_of_types, counts, types = variable_count(att1)
  t.append(types)
  gini_a1, div_index =  gini_of_attribute(no_of_types, counts, rows, cla, types, att1, cl)
  gini_a.append(gini_a1)
  div.append(div_index)

print(gini_a)

delta_gini = list(map(lambda item : gini - item, gini_a))

print(delta_gini)
```

```
index = delta_gini.index(max(delta_gini))
print("\n")
print(col[index], "is the root variable and the variable on one side is ",t[index][div[index]])
```

<u>CART using in-built function</u>

```
import pandas as pd
from sklearn import tree
from sklearn.model_selection import train_test_split
from sklearn.metrics import confusion_matrix, accuracy_score

df = pd.read_csv('CART.csv')

df['Age']=df['Age'].apply(lambda x: 1 if x =='youth' else (2 if x == 'middle' else 3))
df['Income']=df['Income'].apply(lambda x: 1 if x =='low' else (2 if x == 'medium' else 3))
df['Student']=df['Student'].apply(lambda x: 1 if x=='no' else 2)
df['Credit_Rating']=df['Credit_Rating'].apply(lambda x: 1 if x=='fair' else 2)
df['Buys_Computer']=df['Buys_Computer'].apply(lambda x: 1 if x=='no' else 2)

X = df.iloc[ : , 0 : 3]
y = df.iloc[ : , -1]
# X_train, X_test, y_train, y_test = train_test_split(X, y)

clf = tree.DecisionTreeClassifier()
clf.fit(X, y)

tree.plot_tree(clf)
```

OUTPUT

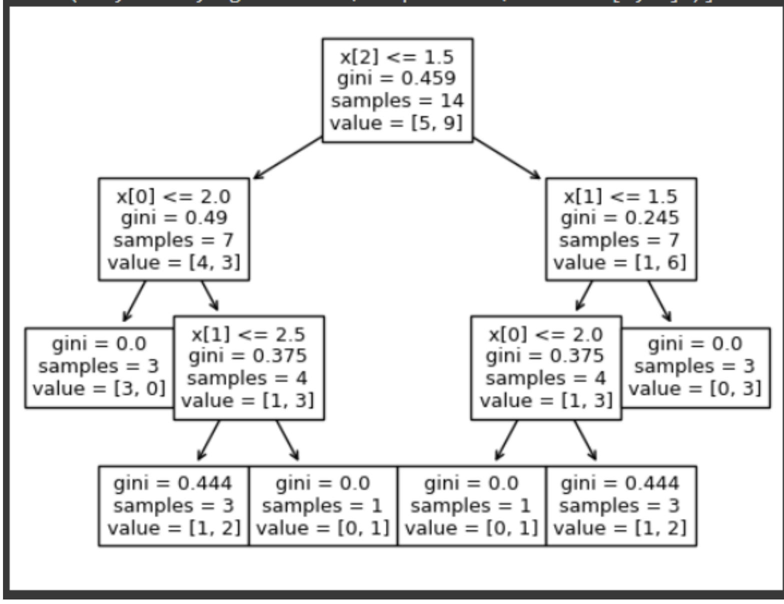
<u>CART</u>

```
0.4591836734693877
[0.35714285714285715, 0.44285714285714295, 0.3673469387755103, 0.428571428571428555]
[0.10204081632653056, 0.01632653061224476, 0.09183673469387743, 0.030612244897959162]


Age is the root variable and the variable on one side is  middle-aged
```

CART using in-built function

```
[Text(0.5, 0.875, 'x[2] <= 1.5\ngini = 0.459\nsamples = 14\nvalue = [5, 9]'),
 Text(0.2, 0.625, 'x[0] <= 2.0\ngini = 0.49\nsamples = 7\nvalue = [4, 3]'),
 Text(0.1, 0.375, 'gini = 0.0\nsamples = 3\nvalue = [3, 0]'),
 Text(0.3, 0.375, 'x[1] <= 2.5\ngini = 0.375\nsamples = 4\nvalue = [1, 3]'),
 Text(0.2, 0.125, 'gini = 0.444\nsamples = 3\nvalue = [1, 2]'),
 Text(0.4, 0.125, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.8, 0.625, 'x[1] <= 1.5\ngini = 0.245\nsamples = 7\nvalue = [1, 6]'),
 Text(0.7, 0.375, 'x[0] <= 2.0\ngini = 0.375\nsamples = 4\nvalue = [1, 3]'),
 Text(0.6, 0.125, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.8, 0.125, 'gini = 0.444\nsamples = 3\nvalue = [1, 2]'),
 Text(0.9, 0.375, 'gini = 0.0\nsamples = 3\nvalue = [0, 3]')]
```



CONCLUSION

Thus, we have successfully implemented CART from scratch and using the in-built functions.