Name: Jigar Siddhpura **SAPID:** 60004200155

DIV: C/C2 **Branch:** Computer Engineering

ML - Experiment 4

Jigar Siddhpura
60004210155
C22

ML - Experiment 4

Aim: To implement CART decision tree algorithm

Theory:
CART (classification and Regression Tax) is a variation of the decision tree algorith. It can handle both classification and regression tasks. It is a pro 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 pre two by considering the best attribute and threshold value. Further, the subsets are also split using 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 trees.
CART algorithm uses Ginni Impurity to split the dataset into a decision tree. It does that by searching for the best homogenity for the sub nodes with the help of Ginni index critriation

Ginni Index / Gini impurity
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 Ginni co-efficient. 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

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

where p_i is the probability of an object being classified to a particular

Advantages of CART

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

Disadvantages

- Overfitting
- High variance
- The tree structure may be unstable

Jigar Siddhpura

60004210155

C22

Gini Index $\alpha(T) = 1 - \sum_{i=1}^{n} p_i^2$

Gini Index $(T, n) = \left|\frac{S_1}{T}\right| \text{Gini}(S_1) + \left|\frac{S_2}{T}\right| \text{Gini}(S_2)$

Step 1:- Gini Index of Target

Gini Index $(T) = 1 - \left(\frac{7}{10}\right)^2 - \left(\frac{3}{10}\right)^2 = 1 - 0.49 - 0.09$

Gini-Index $(T) = 0.42$

Step 2:- Gini Index for each of the attribute and each at the subset in the attribute.

$\boxed{\text{CGPA}}$

CGPA	Job offer = yes	Job offer = No
≥ 9	3	1
≥ 8	4	0
< 8	0	2

1) Gini - Index $(T, \text{CGPA} \in \{\geq 9, \geq 8\}) = 1 - \left(\frac{7}{8}\right)^2 - \left(\frac{1}{8}\right)^2$

$= 0.2194$

Gini - Index $(T, \text{CGPA} \in \{< 8\}) = 1 - \left(\frac{0}{2}\right)^2 - \left(\frac{2}{2}\right)^2 = 0$

Gini Index $(T, \text{CGPA} \in \{\geq 9, \geq 8, < 8\}) = \frac{8}{10} \times 0.2194 + \frac{2}{10} \times 0$

$= 0.17552$

FOR EDUCATIONAL USE

2) Gini Index $(T, CGPA = \{\geq 9, <8\}) = 1 - (\frac{3}{6})^2 - (\frac{3}{6})^2 = 0.5$

Gini Index $(T, CGPA = \{\geq 8\}) = 1 - (\frac{4}{4})^2 - (\frac{0}{4})^2 = 0$

Gini Index $(T, CGPA = \{(\geq 9, <8) \geq 8\} = \frac{6}{10} \times 0.5 + \frac{4}{10} \times 0 = 0.3$

3) Gini-Index $(T, CGPA \{\geq 8, <8\}) = 1 - (\frac{4}{6})^2 - (\frac{2}{6})^2 = 0.445$

Gini Index $(T, CGPA = \{\geq 9\}) = 1 - (\frac{3}{4})^2 - (\frac{1}{4})^2 = 0.375$

Gini Index $(T, CGPA \{\geq 8, <8\} \geq 9\}) = \frac{6}{10} \times 0.445 + 4/10 \times 0.375$

Subsets		Gini Index	
≥9, ≥8	<8	0.1755	→ best splitting subsets
≥9, <8	≥8	0.3	~~for~~ CGPA
≥8, <8	≥9	0.417	

Gini (CGPA) = Gini (T) − Gini (T, CGPA)
$= 0.42 - 0.1755 = 0.2445$

~~Inte~~ Interactiveness

Interactiveness	Job offer = yes	Job offer = No
yes	5	1
No	2	2

FOR EDUCATIONAL USE

Gini Index (T, Interactiveness {yes, No}) = $\frac{6}{10}(0.28) + \frac{4}{10}(0.5) = 0.368$

Gini Index (T, Interactiveness = {yes}) = $1 - \left(\frac{5}{6}\right)^2 - \left(\frac{1}{6}\right)^2 = 0.28$

Gini Index (T, Interactiveness = {No}) = $1 - \left(\frac{2}{4}\right)^2 - \left(\frac{1}{4}\right)^2 = 0.5$

Δ Gini (Interactiveness) = Gini (T) - Gini (T, Interactiveness)
= 0.42 - 0.368
= 0.052

- Practical knowledge

Practical Knowledge	Job offer = yes	Job offer = No
Very good	2	0
Good	4	1
Average	1	2

Subsets		Gini-Index	
(very good, Good)	Average	0.3054	→ best splitting
(very good, average)	Good	0.40	subset for
(Good, Average)	Very good	0.3750	practical knowledge

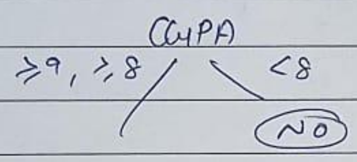
- Δ Gini (Practical knowledge) = Gini (T) - Gini (T, Practical knowledge)
= 0.42 - 0.3054
= 0.1146

Communication skills

Comm skills	Job offer - yes	No
Good	4	1
Moderate	3	0
Poor	0	2

$$\Delta Gini(\text{Comm Skill}) = Gini(T) - Gini(T, \text{Comm Skill})$$
$$= 0.42 - 0.1755$$
$$= 0.2445$$

Attribute	Gini Index	ΔGini
CGPA	0.1755	0.2445
Interactiveness	0.368	0.052
Practical Know	0.3054	0.1146
Comm Skill	0.1755	0.2445



Step 4

Attribute	Gini Index	ΔGini
Interactiveness	0.056	0.1624
Practical know	0.125	0.0934
Comm Skills	0	0.2184

```python
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() # count of each unique attr
    return no_of_types, counts, types
```

```python
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
```

```python
df = pd.read_csv('/content/gdrive/MyDrive/ML/CART.csv')
col = list(df.columns.values.tolist())
```

```python
cl = df.iloc[:, -1]
# cla = [yes,no]
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)
```

```
0.4591836734693877
```

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

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

```python
def plot_decision_tree():
  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)

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)
```

```python
delta_gini = list(map(lambda item : gini - item, gini_a))
print(delta_gini) # highest delta gini wala lenge
index = delta_gini.index(max(delta_gini))
print("\n")
print(col[index], "is the root variable")
```

```
[0.10204081632653056, 0.016326530612244476, 0.09183673469387743, 0.030612244897959162]


Age is the root variable
```

```python
# Call the function to plot the decision tree
plot_decision_tree(gini, gini_a, delta_gini, df)
```

```
tree based on root - Age
student (left)
income (right)
```