

EX.NO: 5

DATE:11/04/2022

DECISION TREE

AIM

To study and implement Decision tree Algorithm.

Procedure:

Step 1: Importing necessary python libraries.

Step 2: Loading the dataset and pre-processing.

Step 3: Splitting the dataset for x and y values.

Step 4: Splitting the dataset for training and testing.

Step 5: Creating the model using DecisionTreeClassifier with CART algorithm.

Step 6: Fitting the model using training data

Step 7: Predicting the values and testing.

Step 8: Visualizing the decision tree.

Program

Decision Tree using CART algorithm

Importing necessary modules

```
In [13]: import pandas as pd
import numpy as np
from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
import graphviz
from sklearn.tree import export_graphviz
```

Loading the dataset

```
In [2]: data = pd.read_csv('data.csv')
original_data = data
# removing the id column
data = data.drop('ID', axis=1)
data.head()
```

Out[2]:

	AGE	SALARY	GRADUATE	CREDIT	LAPTOP
0	YOUTH	HIGH	NO	AVERAGE	NO
1	YOUTH	HIGH	NO	EXCELLENT	NO
2	MIDDLE	HIGH	NO	AVERAGE	YES
3	SENIOR	MEDIUM	NO	AVERAGE	YES
4	SENIOR	LOW	YES	AVERAGE	YES

preprocessing

```
In [3]: le = LabelEncoder()
dataset = data.iloc[:, :]

for i in dataset:
    dataset[i] = le.fit_transform(dataset[i])
```

```
In [4]: data.head()
```

```
Out[4]:
```

	AGE	SALARY	GRADUATE	CREDIT	LAPTOP
0	2	0	0	0	0
1	2	0	0	1	0
2	0	0	0	0	1
3	1	2	0	0	1
4	1	1	1	0	1

```
In [5]: original_data.head()
```

```
Out[5]:
```

	ID	AGE	SALARY	GRADUATE	CREDIT	LAPTOP
0	1	YOUTH	HIGH	NO	AVERAGE	NO
1	2	YOUTH	HIGH	NO	EXCELLENT	NO
2	3	MIDDLE	HIGH	NO	AVERAGE	YES
3	4	SENIOR	MEDIUM	NO	AVERAGE	YES
4	5	SENIOR	LOW	YES	AVERAGE	YES

```
In [6]: X = data.iloc[:, :4].values
y = data.iloc[:, 4].values
```

train-test-split

```
In [7]: X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=1, random_state=2)
```

Creating model

```
In [8]: model=DecisionTreeClassifier(criterion='gini')
model.fit(X_train,y_train)
```

```
Out[8]: DecisionTreeClassifier()
```

prediction

```
In [9]: youth = 2
middle = 0
senior = 1
high = 0
medium = 2
low = 1
no = 0
yes = 1
avg = 0
exe = 1
```

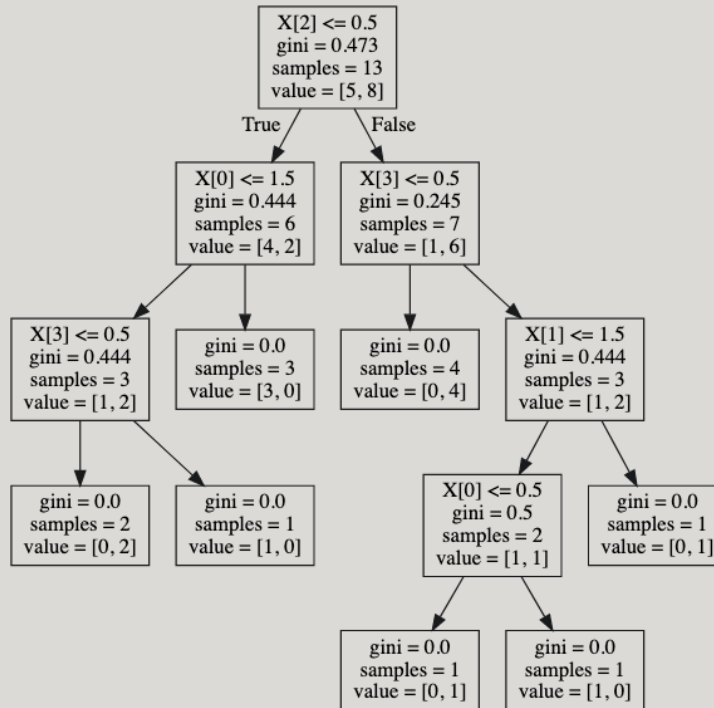
```
In [10]: if model.predict([[middle,high,no,avg]]) == 1:
    print("laptop can be provided")
else:
    print("no laptop")
```

laptop can be provided

Visualizing the decision tree

```
In [14]: dot = export_graphviz(model, out_file=None)
graph = graphviz.Source(dot)
graph
```

Out[14]:



RESULT

Decision tree Algorithm has been studied and implemented successfully.