Ex.no : 12.

# DECISION TREE CLASSIFICATION.

Date :

AIM:

To classify the social network dataset using decision tree analysis

Source code::

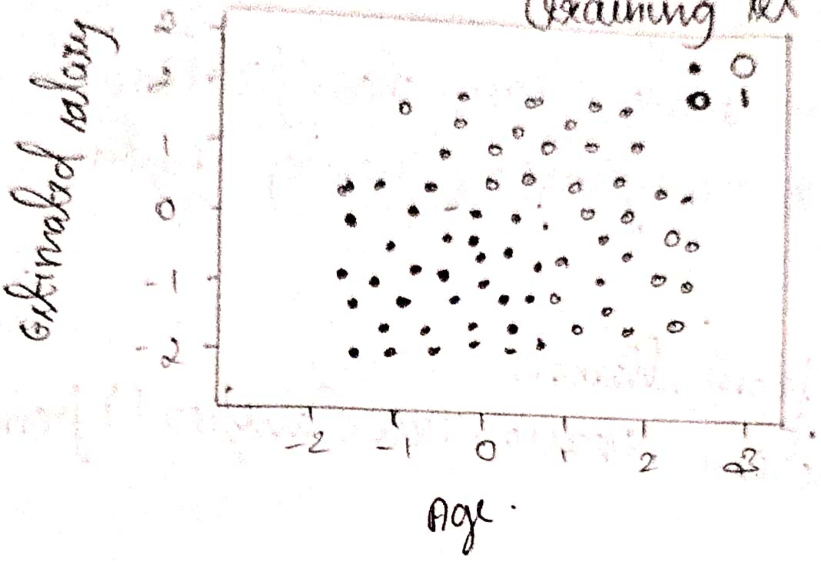
```
from google.colab import drive.
drive.mount("/content/gdrive")
import pandas as pd
import numpy as np.
import matplotlib.pyplot as plt.
dataset = pd.read_csv('/content/gdrive/MyDrive/
              Social_Network_Ads.csv')
 x = dataset.iloc[:, [2,3]].values.
 y = dataset.iloc[:, -1].values.
 from sklearn.model_selection import train_test_
split
x_train, x_test, y_train, y_test = train_test_split
(x, y, test_size = 0.25, random_state = 0)
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
 x_train = sc.fit_transform(x_train)
x_test = sc.transform(x_test)
from sklearn.tree import DecisionTreeClassifier
classifier = DecisionTreeClassifier( criterion = 'entropy',
                    random_state = 0)
```

```
classifier . fit ( x_train , y - train)
y - pred = classifier . predict (x - test ).
from sklearn .metrics import confusion - matrix.
 cm = confusion - matrix ( y - test, y - pred).
 print (cm).
 from sklearn . metrics import confusion_matrix .
 cm ~~x~~_set , y - set = x_train , y - train
 x1 , x2 = np . meshgrid (np . arange ( start =
                              x_set [:, 0] . min () -
 1 - stop = x_set [:, 0] . max () + 1 , step = 0.01),
       np.arrange ( start = x _set [:, 1] . min () -
 1 - stop = x _ set [:, 1] . max () +1, step = 0 . 01))
 plt . contourf (x1 , x2 , classifier . predict (np . array
    ([x1 . ravel () . x2 . ravel ()]) . T) . reshape (
    x1 . shape), alpha = 0.75, cmap =
   ListedColormap (('red', 'green')))
   plt . xlim (x1 . min (), x1 . max ()).
   plt . ylim (x2 . min () , x2 . max )).
  plt . xlabel ('Age')
  plt . ylabel ('Purchase')
  plt . legend ()
   plt . show ()
```

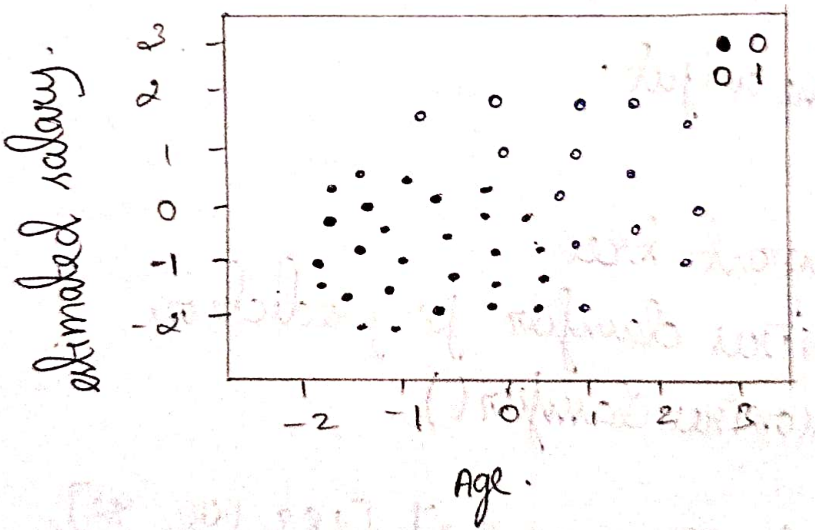
OUTPUT :

```
[[62  6]
 [3  29]].
```

Decision tree classification (Training set).



Decision tree classification (Test set).



RESULT :

The program was successfully executed and the output was verified.