

EXP NO: 13 - DECISION TREE CLASSIFICATION

AIM:

To classify the social network dataset using decision tree analysis.

ALGORITHM:

- 1) Import Libraries and load the dataset
- 2) Define x (features) and y (labels).
- 3) Split data into training and test sets (75% train, 25% test).
- 4) Scale features with standard scaler.
- 5) Initialize and train DecisionTreeClassifier using entropy.
- 6) Predict on test data and generate a confusion matrix.
- 7) Prepare x-set and y-set for visualization.

CODE:

```

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

dataset = pd.read_csv('content/gdrive/my drive/social-network-adh.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 matplotlib.colors import ListedColormap  
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.arrangearray(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()]).  
                                         reshape(x1.shape)), alpha=0.75, cmap=ListedColormap(['red', 'green']))
```

```
plt.xlim(x1.min(), x1.max())
```

```
plt.ylim(x2.min(), x2.max())
```

```
for i, j in enumerate(np.unique(y-set)):
```

```
    plt.scatter(x-set[y-set==j, 0], x-set[y-set==j, 1], c=
```

```
    ListedColormap(['red', 'green'])[i], label=j)
```

```
plt.title('Decision Tree classification (Training set)')
```

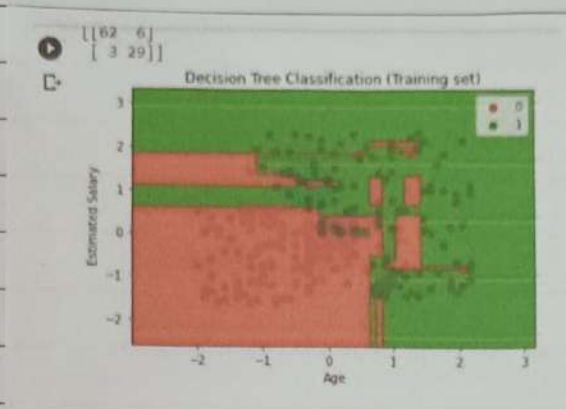
```
plt.xlabel('Age')
```

```
plt.ylabel('Purchase')
```

```
plt.legend()
```

```
plt.show()
```


OUTPUT :



RESULT :

thus ~~decision~~ tree analysis is successfully executed and
~~verified~~