

## Experiment No.: 12 Decision Tree Classification

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/My Drive/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 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.arange(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())
```

```
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)')
```

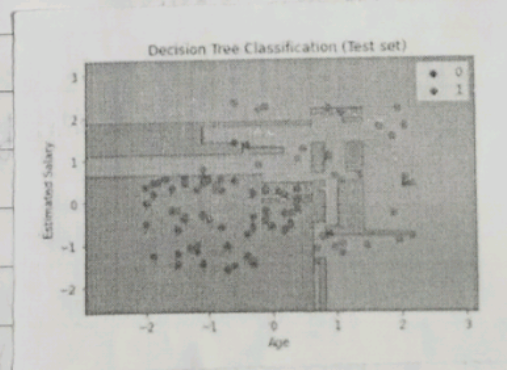
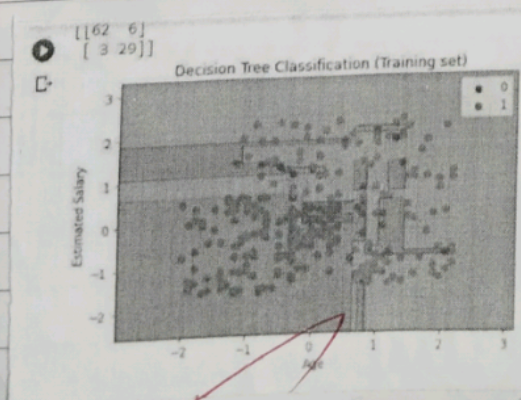
```
plt.xlabel('Age')
```

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

```
plt.legend()
```

```
plt.show()
```

Output:



Result: Thus the program for decision tree classification is successfully executed and the output is verified.