

Exp No : 12

Date : 19/10/24

DECISION TREE ANALYSIS

AIM:

To classify the social network datasets using the decision tree analysis.

PROGRAM:

```
from google.colab import drive
drive.mount("/content/gdrive")

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

from sklearn.model_selection import
    train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import DecisionTreeClassifier
from sklearn.tree import confusion_matrix
from matplotlib.colors import ListedColormap

dataset = pd.read_csv("/content/gdrive/MyDrive/
colab Datasets /social_Network_Ads.csv")

X = dataset.iloc[:, [2, 3]].values
Y = dataset.iloc[:, -1].values

X_train, X_test, Y_train, Y_test = train_test_split
    (X, Y, test_size=0.25, random_state=0)

sc = StandardScaler()
```

```
X_train = sc.fft_transform(X_train)
```

```
X_test = sc.transform(X_test)
```

```
classifier = DecisionTreeClassifier(criterion='entropy',  
                                   random_state=0)
```

```
classifier.fit(X_train, Y_train)
```

```
Y_pred = classifier.predict(X_test)
```

```
cm = confusion_matrix(X_test, Y_pred)
```

```
print("Confusion Matrix:")
```

```
print(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.arange(start=X_set[:,1].min()-1, stop=X_  
max()+1, step=0.01))
```

```
plt.figure(figsize=(10,6))
```

```
cmap_background = ListedColormap(['#FFAAAA', '#AAFFAA'])
```

```
cmap_points = ListedColormap(['red', 'green'])
```

```
plt.contourf(X1, X2, classifier.predict(np.array([X1.ravel()  
X2.ravel()])).T,
```

```
    reshape(X1.shape), alpha=0.75, cmap=cmap_background)
```

```
plt.xlim(X1.min(), X1.max())
```

```
plt.ylim(X2.min(), X2.max())
```

for c, j in enumerate(np.unique('y-set')):

plt.scatter(x-set['y-set == j, 0'], x-set['y-set == j, 1'])

c = cmap = points(i), label=j)

plt.title('Decision Tree Classification (Training sets)')

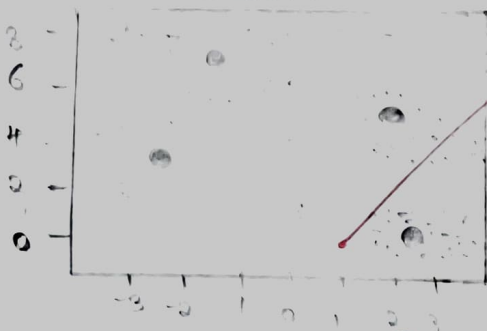
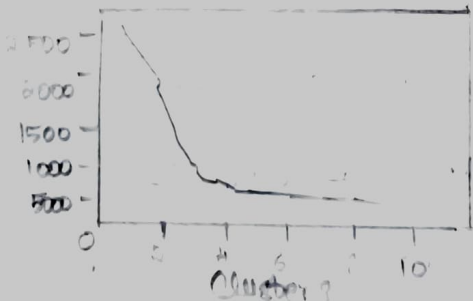
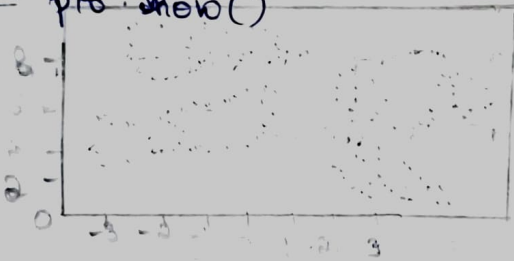
plt.xlabel('Age')

plt.ylabel('Estimated Salary')

plt.legend()

OUTPUT:

plt.show()



RESULT:

Thus, the decision tree implementation is created successfully and output is verified.