

Exp: No: 11

Implementing artificial neural networks for an application using python - Regression.

Aim:- To Implementing artificial neural networks for an application in regression using python.

Source Code:-

```
from sklearn.neural_network import MLPRegressor
from sklearn.model_selection import train_test_split
from sklearn.datasets import make_regression
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import matplotlib
%matplotlib inline
x, y = make_regression(n_samples=1000, noise=0.05, n_features=100)
x.shape, y.shape = (1000, 100), (1000, 1)
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2, shuffle=True, random_state=42)
clf = MLPRegressor(max_iter=1000)
clf.fit(x_train, y_train)
```

O/P

R₂ score for test Data = 0.9686558466
21529

Result:-

The program was successfully executed

Aim: To classify the social network data 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(' ')
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)
sc = StandardScaler()
x_train = sc.fit_transform(x_train)
x_test = sc.transform(x_test)
x1, x2 = np.meshgrid(np.arange(
    start = x_set[:, 0].min(),
    stop = x_set[:, 0].max() + 1, step = 0.01,
    np.arange(start = x_set[:, 1].min(),
    stop = x_set[:, 1].max() + 1, step = 0.01,

```

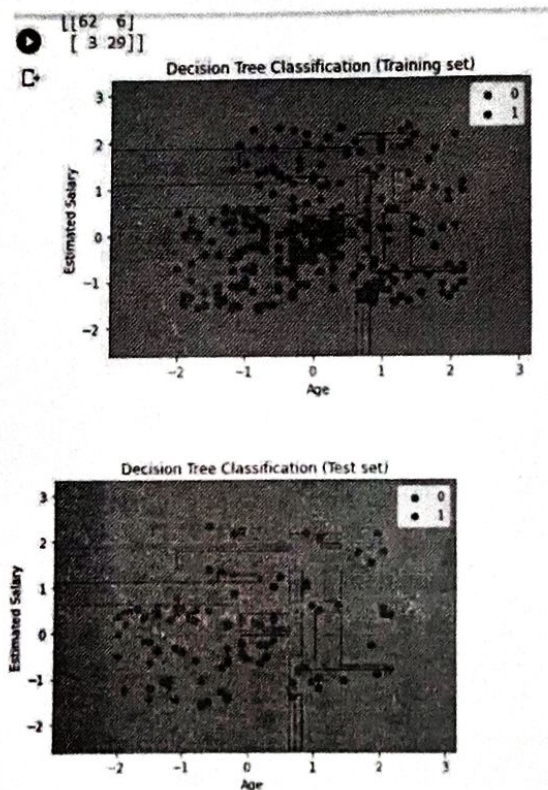
```

plt.title('Decision Tree classification  
(Training Set)')
plt.xlabel('Age')
plt.ylabel('Purchase')
plt.legend()
plt.show()

```

Output:

OUTPUT:



Result:

The program was executed successfully, and the o/p is verified.