

## Experiment - 8

# Implementation of Artificial Neural Networks for an Application Using Python - Regression

### AIM

To implement artificial neural networks for an application in Regression using Python.

### PROGRAM

```
from sklearn.neural_network import MLPRegressor
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline

X, y = make_regression(n_samples=1000, noise=0.05,
                      n_features=100)

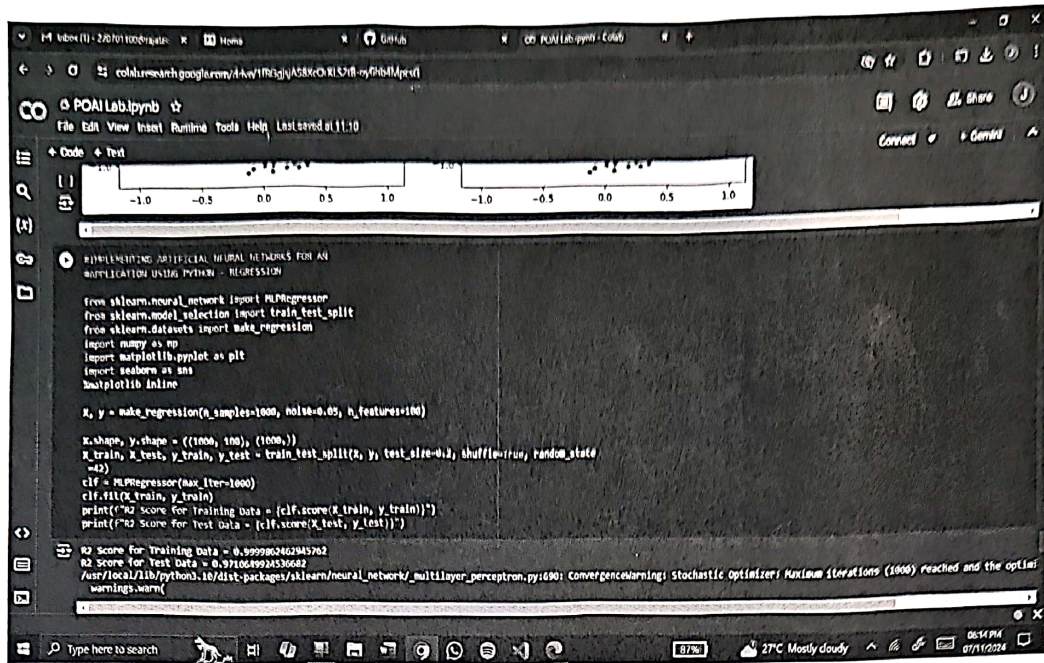
X.shape, y.shape = (1000, 100), (1000,)

X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, shuffle=True, random_state=42)

clf = MLPRegressor(hidden_layer_sizes=(100,))
clf.fit(X_train, y_train)

print(f'R2 score for training data = {clf.score(X_train, y_train)}')
print(f'R2 score for Test data = {clf.score(X_test, y_test)}')
```

## OUTPUT



```
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
matplotlib inline

X, y = make_regression(n_samples=1000, noise=0.05, n_features=10)

X.shape, y.shape = ((1000, 10), (1000,))
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)
print("R2 Score for Training Data = {clf.score(X_train, y_train)}")
print("R2 Score for Test Data = {clf.score(X_test, y_test)}")

R2 Score for Training Data = 0.9999802462945702
R2 Score for Test Data = 0.9710649924536682
/usr/local/lib/python3.10/dist-packages/sklearn/neural_network/multilayer_perceptron.py:690: ConvergenceWarning: Stochastic Optimizer: Maximum iterations (1000) reached and the optimization was terminated.
warnings.warn(
```

## RESULT

thus, the program for ANN using Regression is successfully executed and the output is verified.