EXP NO8 IMPLEMENTING ARTIFICIAL NEURAL NETWORKS FOR AN APPLICATION USING PYTHON-REGRESSION

AIM:-

To implementing artificial neural networks for an application in Regression using python.

PSEUDO CODE:-

1. Import necessary libraries (sklearn, numpy, matplotlib, seaborn)
2. Generate a regression dataset:
   - 1000 samples
   - 100 features
   - Add noise to the data
3. Split the dataset into training and test sets:
   - Training set: 80% of data
   - Test set: 20% of data.
   - Shuffle the data and set a random seed for reproducibility
4. Initialize an MLP regressor with maximum iterations set to 1000
5. Train the regressor on the training data ('X_train', 'y_train').
6. Compute and display the R2 scores for:
   - Training data
   - Test data.

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
% matplotlib inline

X, y = make_regression (n_samples = 1000, noise = 0.05, 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 (max_iter = 1000)
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:-

R2Score for Training Data = 0.9999893761154967
R2Score for Test Data = 0.9699273107315 87

RESULT:-

Therefore, the program is successfully executed & output is verified.