

## EXP 8

# IMPLEMENTING ARTIFICIAL NEURAL NETWORKS FOR AN APPLICATION USING PYTHON - CLASSIFICATION (7)

AIM :-

To implement artificial neural networks for an application in classification using python.

PSEUDOCODE :-

1. Import necessary libraries (sklearn, numpy, matplotlib, seaborn)
2. Generate circular datasets for training (100 samples) and testing (100 samples) with noise.
3. Plot the training data using a scatter plot:
  - x-axis: Feature 1
  - y-axis: Feature 2
  - color by label (hue)
4. Initialize an MLP classifier with max iteration set to 1000
5. Train the classifier on the training data ('X\_train', 'y\_train').
6. Compute and display the R2 scores for:
  - Training data
  - Test data
7. Predict labels for the test data ('X\_test').
8. Create 2 subplots:
  - a. Plot predicted labels for test data
  - b. Plot actual label for test data
9. Display the subplots for comparison.

CODE 2

~~###~~

```
from sklearn.model_selection import train_test_split
from sklearn.datasets import make_circles
from sklearn.neural_network import MLPClassifier
import numpy as np
import matplotlib.pyplot as plt.
import seaborn as sns
```

```
%matplotlib inline
```

```
x_train, y_train = make_circles(n_samples=700,
                                noise=0.05)
```

```
x_test, y_test = make_circles(n_samples=400,
                               noise=0.05)
```

```
sns.scatterplot(x=x_train[:,0], y=x_train[:,1],
                hue=y_train)
```

```
plt.title("Train Data")
```

```
plt.show()
```

```
clf = MLPClassifier(max_iter=1000)
```

```
clf.fit(x_train, y_train)
```

```
print(f"R4 R2 score for training data = {clf.score(x_train, y_train)}")
```

```
print(f"R4 R2 score for testing data = {clf.score(x_test, y_test)}")
```

```
clf = MLPClassifier(max_iter=1000)
```

```
clf.fit(x_train, y_train)
```

# Prediction below

```
y_pred = clf.predict(X_test)
```

```
fig, ax = plt.subplots(1, 2, figsize=(12, 5))
```

```
sns.scatterplot(x=X_test[:, 0], y=X_test[:, 1],
```

```
hue=y_pred, ax=ax[0])
```

```
ax[0].title.set_text("Predicted data")
```

```
sns.scatterplot(x=X_test[:, 0], y=X_test[:, 1],
```

```
hue=y_test, ax=ax[1])
```

```
ax[1].title.set_text("Test Data")
```

```
plt.show()
```

OUTPUT:-

Result:- Program is successfully executed  
verified

