In [1]:
```
import numpy as np
import math as mth
import matplotlib.pyplot as plt
import sklearn.model_selection as ms
import sklearn.neural_network as nn
import sklearn.metrics as met
import scipy.stats as stt
```

In [2]:
```
def F(x1, x2, x3, x4):
    y = 1.4 * x1 - 1.6 * x2 ** 2 + mth.sin(3 * x3) + 2.7 * abs(x4 - 1) + 2
    return y
```

In [3]:
```
np.random.seed(50)
Min = -3
Max = +3
N = 4000
x = np.zeros((N, 4))
y = np.zeros((N))
```

In [6]:
```
for i in range(0, N):
    x[i, 0] = np.random.uniform(Min, Max)
    x[i, 1] = np.random.uniform(Min, Max)
    x[i, 2] = np.random.uniform(Min, Max)
    x[i, 3] = np.random.uniform(Min, Max)
    y[i] = F(x[i, 0], x[i, 1], x[i, 2], x[i, 3])
```

In [7]:
```
xtr, xte, ytr, yte = ms.train_test_split(x, y, train_size = 0.75, random_state = 45)
```

In [8]:
```
NN = nn.MLPRegressor(hidden_layer_sizes = (40, 50),
                     activation = "relu",
                     solver = "adam",
                     learning_rate_init = 0.01,
                     max_iter = 5000,
                     random_state = 50)
```

In [9]:
```
NN.fit(xtr, ytr.ravel())
```

Out[9]:
```
MLPRegressor(hidden_layer_sizes=(40, 50), learning_rate_init=0.01,
             max_iter=5000, random_state=50)
```

In [11]:
```
trPred = NN.predict(xtr)
tePred = NN.predict(xte)

trMSE = met.mean_squared_error(ytr, trPred)
teMSE = met.mean_squared_error(yte, tePred)

trPCC, _ = stt.pearsonr(ytr, trPred)
tePCC, _ = stt.pearsonr(yte, tePred)

trMAPE = met.mean_absolute_percentage_error(ytr, trPred)
teMAPE = met.mean_absolute_percentage_error(ytr, trPred)
```

In [12]:
```
print ("Train MSE : ", trMSE)
print ("Test MSE : ", teMSE)
print ("-" * 35)
print ("Train pcc : ", trPCC)
print ("Test pcc : ", tePCC)
print ("-" * 35)
print ("Train MAPE : ", trMAPE)
print ("Test MAPE : ", teMAPE)
```

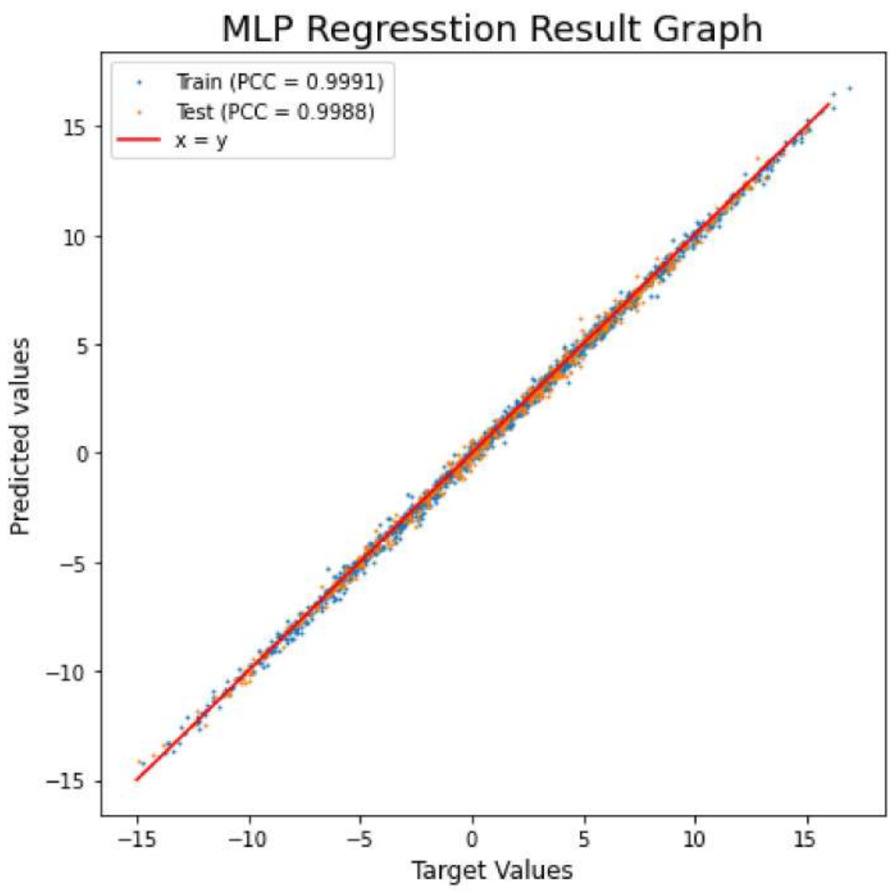
```
Train MSE :  0.06327411488706383
Test MSE :  0.08272028383001365
-----------------------------------
Train pcc :  0.9991207835635876
Test pcc :  0.998801108690069
-----------------------------------
Train MAPE :  0.16313966642168426
Test MAPE :  0.16313966642168426
```

In [16]:

```
plt.figure(figsize = (7, 7))
plt.scatter(ytr[::2], trPred[::2], label = "Train (PCC = {})".format(round(trPCC, 4)),
plt.scatter(yte[::2], tePred[::2], label = "Test (PCC = {})".format(round(tePCC, 4)),
plt.plot([-15, +16], [-15, +16], label = "x = y", c = "r")
plt.title("MLP Regresstion Result Graph", fontsize = 18)
plt.xlabel("Target Values", fontsize = 12)
plt.ylabel("Predicted values", fontsize = 12)
plt.legend()
```

Out[16]:

```
<matplotlib.legend.Legend at 0x17ee2f40a90>
```



In [ ]: