la: No: 11 Regression (f(tal. y tad.)))) Date : 4- pred = self. predict (x seet) pit. figure (figure = (86 6)) Program lode: pit . plat (1 y. new () J. max()]. I Import python libraries. ox - notax, lirent & 1. matplotlib whine pht little (" true us pradicted value X, y = make _ regression (n_ samples = 1000, hoise = 0.05, h_features = 100) Alt. upaled 1 Bredited va print (f" x shape: {x. shape }, y shape: { y. Shape }") x_train, x_test, y_train, y_test = train_test_ Split (x, y, test_size = 0.2, shuffle = Irue, random_state = 42) MLP Regressor (max_iter = 1000) elf. fet (x-train, y-train) print (f" R2 Score for Training Data = { if. Score (x_ train, y-train)}")

print (1" R2 Score for Test Data = { clf. Score (x_test, y_test)3') y-pred = ill. predict (x-test) plt, figure (figsize = (8,6)) plt. plot ([y. min (), y. max()], (y. min(), y. max ()], rolor = 'red', lu = 2) plt. title ('Irue vs predicted values') plt. slabel ('True values') plt. y label (' Predicted values') print (+" x shape (x. Shape) plt. Show () (y shupe)") output: train x text, y train, y text Shapes of X and y: (1000, 100) (1000,) R2 Score for Training Data = 1.00 R2 Score for Test Data = 0.97 MLP REQUESTA (max - ster = 1000 if get (x train, y train) print (f' R2 Surve , low Mathing diala program is successfully executed & the output is verified.