**Program 9**

**9. Implement the non-parametric Locally Weighted Regression algorithm in order to fit data points. Select appropriate data set for your experiment and draw graphs. LOCALLY WEIGHTED REGRESSION**

from math import ceil

import numpy as np

from scipy import linalg

def lowess(x, y, f=2./3., iter=3):

n = len(x)

r = int(ceil(f\*n))

h = [np.sort(np.abs(x - x[i]))[r] for i in range(n)]

w = np.clip(np.abs((x[:,None] - x[None,:]) / h), 0.0, 1.0)

w = (1 - w\*\*3)\*\*3

yest = np.zeros(n)

delta = np.ones(n)

for iteration in range(iter):

for i in range(n):

weights = delta \* w[:,i]

b = np.array([np.sum(weights\*y), np.sum(weights\*y\*x)])

A = np.array([[np.sum(weights), np.sum(weights\*x)],

[np.sum(weights\*x), np.sum(weights\*x\*x)]])

beta = linalg.solve(A, b)

yest[i] = beta[0] + beta[1]\*x[i]

residuals = y - yest

s = np.median(np.abs(residuals))

delta = np.clip(residuals / (6.0 \* s), -1, 1)

delta = (1 - delta\*\*2)\*\*2

return yest

if \_\_name\_\_ == '\_\_main\_\_':

import math

n = 100

x = np.linspace(0, 2 \* math.pi, n)

print("==========================values of x=====================")

print(x)

y = np.sin(x) + 0.3\*np.random.randn(n)

print("================================Values of y===================")

print(y)

f = 0.25

yest = lowess(x, y, f=f, iter=3)

import pylab as pl

pl.clf()

pl.plot(x, y, label='y noisy')

pl.plot(x, yest, label='y pred')

pl.legend()

pl.show()

**output**

==========================values of x=====================

[0. 0.06346652 0.12693304 0.19039955 0.25386607 0.31733259

0.38079911 0.44426563 0.50773215 0.57119866 0.63466518 0.6981317

0.76159822 0.82506474 0.88853126 0.95199777 1.01546429 1.07893081

1.14239733 1.20586385 1.26933037 1.33279688 1.3962634 1.45972992

1.52319644 1.58666296 1.65012947 1.71359599 1.77706251 1.84052903

1.90399555 1.96746207 2.03092858 2.0943951 2.15786162 2.22132814

2.28479466 2.34826118 2.41172769 2.47519421 2.53866073 2.60212725

2.66559377 2.72906028 2.7925268 2.85599332 2.91945984 2.98292636

3.04639288 3.10985939 3.17332591 3.23679243 3.30025895 3.36372547

3.42719199 3.4906585 3.55412502 3.61759154 3.68105806 3.74452458

3.8079911 3.87145761 3.93492413 3.99839065 4.06185717 4.12532369

4.1887902 4.25225672 4.31572324 4.37918976 4.44265628 4.5061228

4.56958931 4.63305583 4.69652235 4.75998887 4.82345539 4.88692191

4.95038842 5.01385494 5.07732146 5.14078798 5.2042545 5.26772102

5.33118753 5.39465405 5.45812057 5.52158709 5.58505361 5.64852012

5.71198664 5.77545316 5.83891968 5.9023862 5.96585272 6.02931923

6.09278575 6.15625227 6.21971879 6.28318531]

================================Values of y===================

[ 0.1240189 0.664614 0.06400676 0.2207773 -0.00313679 -0.03981776

0.11318095 0.5432104 0.18210662 0.43932693 0.32870882 0.7155147

0.80598575 0.49379539 0.42596786 0.75549759 0.82726288 0.59627161

0.70823627 0.90008044 0.79967111 1.07362763 1.50742128 1.34864406

0.89823727 0.69453562 1.04908341 1.30509424 1.21366036 0.84916839

1.2434968 0.72447522 1.35045102 0.24776388 0.68830294 1.11454485

0.18204808 0.86140467 1.001109 0.85736372 0.61299745 0.70543754

0.5123533 0.45805346 0.62181211 0.54312735 -0.24334421 -0.02624053

-0.55951705 -0.13337209 -0.12525381 -0.30316107 -0.37886905 -0.25203055

-0.61803269 -0.59939626 -0.11006446 -0.55773841 -0.3182158 -0.50232376

-0.25540226 -0.31646237 -0.81352062 -1.29218533 -0.65185225 -0.7434856

-1.01770899 -0.99719141 -1.20387864 -1.05627416 -1.59662351 -1.46003532

-1.06858937 -0.87968458 -0.83224566 -1.56871982 -0.8951245 -0.76804187

-1.88884092 -0.46843078 -0.96196433 -1.11454782 -0.98912405 -0.91100319

-0.47704966 -0.82176946 -1.11817551 -0.75720119 -1.43230053 0.04558205

-0.0967495 -0.72741239 -0.38869797 -0.47947553 -0.63366026 -0.72990181

-0.07260149 -0.06519691 -0.23719697 0.47426349]

