**Pgm 10:** Implement the non-parametric **Locally Weighted Regression algorithm** in order to fit data points. Select appropriate data set for your experiment and draw graphs.

import numpy as np

from bokeh.plotting import figure, show, output\_notebook

from bokeh.layouts import gridplot

from bokeh.io import push\_notebook

def local\_regression(x0, X, Y, tau):

x0 = np.r\_[1, x0]

X = np.c\_[np.ones(len(X)), X]

xw = X.T \* radial\_kernel(x0, X, tau)

beta = np.linalg.pinv(xw @ X) @ xw @ Y

return x0 @ beta

def radial\_kernel(x0, X, tau):

return np.exp(np.sum((X - x0) \*\* 2, axis=1) / (-2 \* tau \* tau))

n = 1000

X = np.linspace(-3, 3, num=n)

print("The Data Set ( 10 Samples) X :\n",X[1:10])

Y = np.log(np.abs(X \*\* 2 - 1) + .5)

print("The Fitting Curve Data Set (10 Samples) Y :\n",Y[1:10])

X += np.random.normal(scale=.1, size=n)

print("Normalised (10 Samples) X :\n",X[1:10])

domain = np.linspace(-3, 3, num=300)

print(" Xo Domain Space(10 Samples) :\n",domain[1:10])

def plot\_lwr(tau):

prediction = [local\_regression(x0, X, Y, tau) for x0 in domain]

plot = figure(plot\_width=400, plot\_height=400)

plot.title.text='tau=%g' % tau

plot.scatter(X, Y, alpha=.3)

plot.line(domain, prediction, line\_width=2, color='red')

return plot

show(gridplot([

[plot\_lwr(10.), plot\_lwr(1.)],

[plot\_lwr(0.1), plot\_lwr(0.01)]]))

**Output**

The Data Set ( 10 Samples) X :

[-2.99399399 -2.98798799 -2.98198198 -2.97597598 -2.96996997 -2.96396396

-2.95795796 -2.95195195 -2.94594595]

The Fitting Curve Data Set (10 Samples) Y :

[2.13582188 2.13156806 2.12730467 2.12303166 2.11874898 2.11445659

2.11015444 2.10584249 2.10152068]

Normalised (10 Samples) X :

[-2.89750158 -3.01975929 -3.03154001 -2.75326359 -2.83690136 -3.11105613

-3.02689842 -3.01589928 -2.89973361]

Xo Domain Space(10 Samples) :

[-2.97993311 -2.95986622 -2.93979933 -2.91973244 -2.89966555 -2.87959866

-2.85953177 -2.83946488 -2.81939799]