**PROGRAM 6 :**

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

*import* matplotlib.pyplot *as* plt

def gaussian\_kernel(*x*, *xi*, *tau*):

*return* np.exp(-np.sum((x - xi) \*\* 2) / (2 \* tau \*\* 2))

def locally\_weighted\_regression(*x*, *X*, *y*, *tau*):

m = X.shape[0]

weights = np.array([gaussian\_kernel(x, X[i], tau) *for* i *in* range(m)])

W = np.diag(weights)

X\_transpose\_W = X.T @ W

theta = np.linalg.inv(X\_transpose\_W @ X) @ X\_transpose\_W @ y

*return* x @ theta

np.random.seed(42)

X = np.linspace(0, 2 \* np.pi, 100)

y = np.sin(X) + 0.1 \* np.random.randn(100)

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

x\_test = np.linspace(0, 2 \* np.pi, 200)

x\_test\_bias = np.c\_[np.ones(x\_test.shape), x\_test]

tau = 0.5

y\_pred = np.array([locally\_weighted\_regression(xi, X\_bias, y, tau) *for* xi *in* x\_test\_bias])

plt.figure(*figsize*=(10, 6))

plt.scatter(X, y, *color*='red', *label*='Training Data', *alpha*=0.7)

plt.plot(x\_test, y\_pred, *color*='blue', *label*=f'LWR Fit (tau={tau})', *linewidth*=2)

plt.xlabel('X', *fontsize*=12)

plt.ylabel('y', *fontsize*=12)

plt.title('Locally Weighted Regression', *fontsize*=14)

plt.legend(*fontsize*=10)

plt.grid(*alpha*=0.3)

plt.show()

