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

from numpy import \*

import operator

from os import listdir

import matplotlib

import matplotlib.pyplot as plt

import pandas as pd

import numpy.linalg

from scipy.stats.stats import pearsonr

def kernel(point,xmat,k):

m,n=shape(xmat)

weights=mat(eye((m)))

for j in range(m):

diff=point-X[j]

weights[j,j]=exp(diff\*diff.T/(-2.0\*k\*\*2))

return weights

def localWeight(point,xmat,ymat,k):

wei=kernel(point,xmat,k)

W=((X.T\*wei\*X)).I\*(X.T\*(wei\*ymat.T))

return W

def localWeightRegression(xmat,ymat,k):

m,n=shape(xmat)

ypred=zeros(m)

for i in range(m):

ypred [i]=xmat[i]\*localWeight(xmat[i],xmat,ymat,k)

return ypred

data=pd.read\_csv('tips.csv')

bill=array(data.bill)

tip=array(data.tip)

mbill=mat(bill)

mtip=mat(tip)

m=shape(mbill)[1]

one=mat(ones(m))

X=hstack((one.T,mbill.T))

ypred=localWeightRegression(X,mtip,0.5)

SortIndex=X[:,1].argsort(0)

xsort=X[SortIndex][:,0]

fig=plt.figure()

ax=fig.add\_subplot(1,1,1)

ax.scatter(bill,tip,color='green')

ax.plot(xsort[:,1],ypred[SortIndex],color='red',linewidth=5)

plt.xlabel('totalbil')

plt.ylabel('tip')

plt.show();

**tips.csv**

bill,tip

3000,30

400,20

5000,40

8000,50

**OUTPUT**

