Name – Harsh
Roll Number – 36
Registration Number – 201800469
Semester – 7th
Section – C

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

1. Submit pdf with code and graph.

2. Perform the experiment for at least two data sets.

3. Analyze with different test cases.

Solution –

ALGORITHM:

1. Start.
2. Read the Given Data Sample to X and the curve to Y.
3. Set the value for Smoothening parameter or Free parameter say τ.
4. Set the bias /Point of interest set x0 which is a subset of X.
5. Determine the weight matrix using:

$$w(x, x_o) = e^{-\frac{(x-x_o)^2}{2\tau^2}}$$

6. Determine the value of model term parameter β using:

$$\hat{\beta}(x_o) = (X^T W X)^{-1} X^T W y$$

7. Prediction = x0*β.
8. Stop.

Code:

1. **Dataset used – '**10-dataset.csv'

```
In [6]: data
```

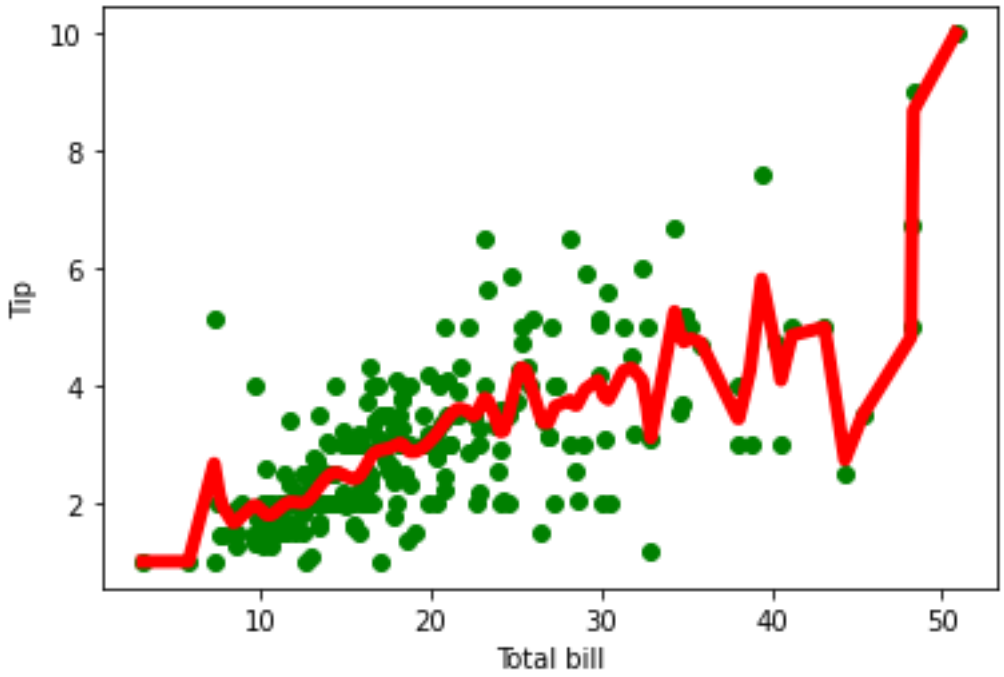
Out[6]:

	total_bill	tip	sex	smoker	day	time	size
0	16.99	1.01	Female	No	Sun	Dinner	2
1	10.34	1.66	Male	No	Sun	Dinner	3
2	21.01	3.50	Male	No	Sun	Dinner	3
3	23.68	3.31	Male	No	Sun	Dinner	2
4	24.59	3.61	Female	No	Sun	Dinner	4
...	...	...	...	...	...	...	...
239	29.03	5.92	Male	No	Sat	Dinner	3
240	27.18	2.00	Female	Yes	Sat	Dinner	2
241	22.67	2.00	Male	Yes	Sat	Dinner	2
242	17.82	1.75	Male	No	Sat	Dinner	2
243	18.78	3.00	Female	No	Thur	Dinner	2

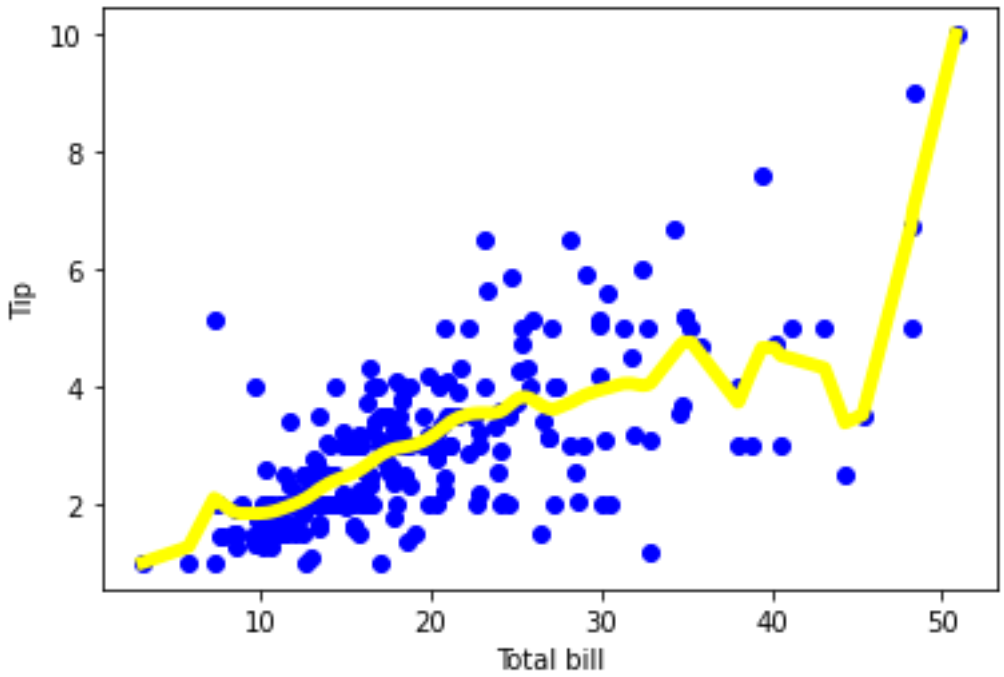
244 rows × 7 columns

```
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np

def kernel(point, xmat, k):
   m,n = np.shape(xmat)
   weights = np.mat(np.eye((m)))
   for j in range(m):
      diff = point - X[j]
      weights[j,j] = np.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 = np.shape(xmat)
   ypred = np.zeros(m)
   for i in range(m):
      ypred[i] = xmat[i]*localWeight(xmat[i],xmat,ymat,k)
   return ypred

# Load data points.

data = pd.read_csv('10-dataset.csv')
bill = np.array(data.total_bill)
tip = np.array(data.tip)

# Preparing and add 1 in bill.

mbill = np.mat(bill)
mtip = np.mat(tip)

m= np.shape(mbill)[1]
one = np.mat(np.ones(m))
X = np.hstack((one.T,mbill.T))

# Set k=0.5.

ypred = localWeightRegression(X,mtip,0.5)
SortIndex = X[:,1].argsort(0)
xsort = X[SortIndex][:,0]

# Set k=1.
Y_pred = localWeightRegression(X,mtip,1)

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('Total bill')
plt.ylabel('Tip')
plt.show();
```




```
fig = plt.figure()
ax = fig.add_subplot(1,1,1)
ax.scatter(bill,tip, color='blue')
ax.plot(xsort[:,1],ypred1[SortIndex], color = 'yellow', linewidth=5)
plt.xlabel('Total bill')
plt.ylabel('Tip')
plt.show();
```



2. **Dataset used –** 'Poly_Dataset.csv'

```
In [6]: data
```

Out[6]:

	Age	Height
0	10	138
1	11	138
2	12	138
3	13	139
4	14	139
...	...	...
66	76	204
67	77	205
68	78	206
69	79	207
70	80	208

71 rows × 2 columns

```
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np

def kernel(point, xmat, k):
    m,n = np.shape(xmat)
    weights = np.mat(np.eye((m)))
    for j in range(m):
        diff = point - X[j]
        weights[j,j] = np.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 = np.shape(xmat)
    ypred = np.zeros(m)
    for i in range(m):
        ypred[i] = xmat[i]*localWeight(xmat[i],xmat,ymat,k)
    return ypred

# Load data points.

data = pd.read_csv('Poly_Dataset.csv')
```

```
Age = np.array(data.Age)
Height = np.array(data.Height)
```

 # Preparing and add 1 in Age.

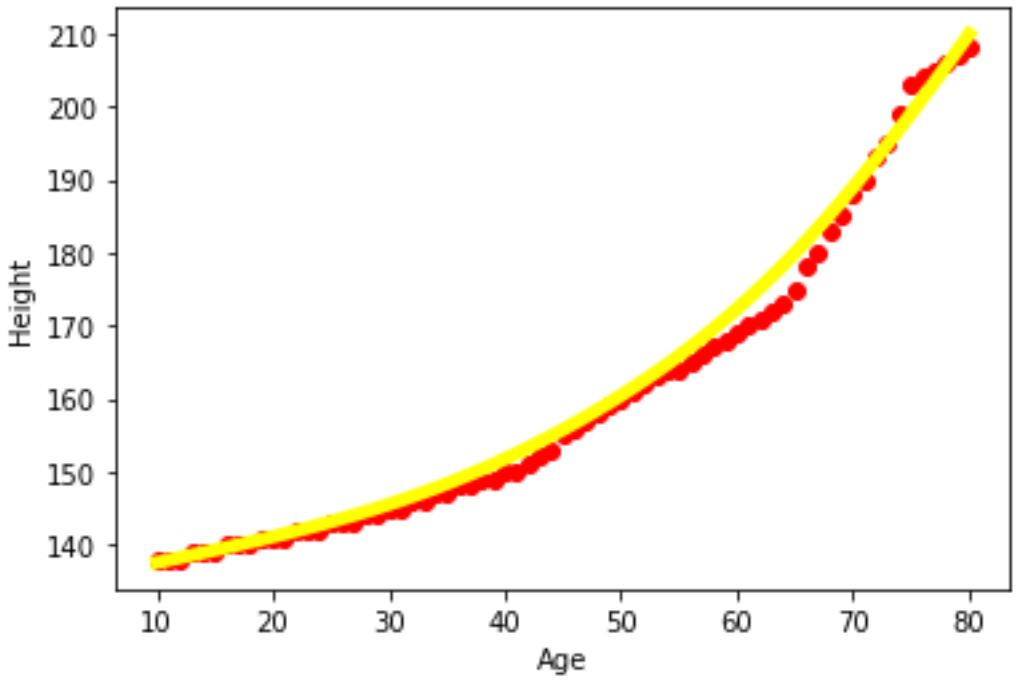
```
mAge = np.mat(Age)
mHeight = np.mat(Height)

m= np.shape(mAge)[1]
one = np.mat(np.ones(m))
X = np.hstack((one.T,mAge.T))
```

 # Set k=10.

```
ypred = localWeightRegression(X,mHeight,10)
SortIndex = X[:,1].argsort(0)
xsort = X[SortIndex][:,0]
```

 # Set k=20.

```
ypred1 = localWeightRegression(X,mHeight,20)

fig = plt.figure()
ax = fig.add_subplot(1,1,1)
ax.scatter(Age,Height, color='red')
ax.plot(xsort[:,1],ypred[SortIndex], color = 'yellow', linewidth=5)
plt.xlabel('Age')
plt.ylabel('Height')
plt.show();
```




```
fig = plt.figure()
ax = fig.add_subplot(1,1,1)
```

```
ax.scatter(Age,Height, color='green')
ax.plot(xsort[:,1],ypred1[SortIndex], color = 'yellow', linewidth=5)
plt.xlabel('Age')
plt.ylabel('Height')
plt.show();
```