

Date: 15/06/2021

Question 5

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

Program:

```
from numpy import *
import operator
from os import listdir
import matplotlib
import matplotlib.pyplot as plt
import pandas as pd
from numpy.linalg import *

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

from google.colab import drive
drive.mount('/content/drive')
```

```

data = pd.read_csv("/content/drive/MyDrive/dataset.csv")
x=array(data.X)
y=array(data.Y)
mx=mat(x)
my=mat(y)
m=shape(mx)[1]
one = mat(ones(m))
X=hstack((one.T,mx.T))

# set k here
ypred = localWeightRegression(X, my, 10)
SortIndex = X[:, 1].argsort(0)
xsort = X[SortIndex][:, 0]
fig = plt.figure()
ax = fig.add_subplot(1, 1, 1)
ax.scatter(x,y, color="green")
ax.plot(xsort[:, 1], ypred[SortIndex], color="red", linewidth=5)
plt.xlabel("X")
plt.ylabel("Y")
plt.show()

print("Predicted Value of Y for X=55 ",ypred)

```

Dataset:

A	B	C
Sub	X	Y
1	43	95
2	21	65
3	25	79
4	42	75
5	57	87
6	59	81