15/06/2021 SUB: MACHINE LEARNING NAME: PRATHIBHA.R

Lab test - 2 CLASS: 6B.

USN: 1BM18CS074

5. Implement the non parametric locally weighted regression algorithm in order to fit data points. Select appropriate dataset 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.
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.
```

15/06/2021 Prathibha R - 1BM18CS074.

```
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("data.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.
 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], ypread.[SortIndex], color = "red", linewidth
                                                              = 5)
  plt.xlabel("x")
  plt.ylabel("y")
  plt.show().
```