**U18ISI6204 – Machine Learning Techniques**

**LAB- EXPERIMENT 10**

|  |  |
| --- | --- |
| NAME | Navina G A |
| ROLL NO | 20BIS028 |

# Name : SANDHIYA J

# Roll no : 20BIS036

**INTRODUCTION**

In this experiment, we have to perform non parametric locally weighted regression on the restaurant dataset.

Loess regression is a nonparametric technique that uses local weighted regression to fit a smooth curve through points in a scatter plot.

A picture containing line, plot, diagram, text

Description automatically generated

# OBJECTIVE OF THE EXERCISE/EXPERIMENT

To perform non parametric locally weighted regression on the given dataset, using scikit library

# STEP 2: ACQUISITION PROCEDURE:

**STEP-1:** Start the program.

**STEP-2:** import all the necessary libraries

1. Numpy – array manipulation
2. Pandas – dataframe manipulation
3. Matplotlib– for data visualization

**STEP-3:** Loading the dataset using read\_csv method in pandas module.

**STEP-4:** Read the Given data Sample to X and the curve (linear or non linear) to Y

**STEP-5:** Set the value for Smoothening parameter or Free parameter say τ

**STEP-6:** Set the bias /Point of interest set x0 which is a subset of X

**STEP-7:** Determine the weight matrix using :



**STEP-8:** Determine the value of model term parameter β using:



**STEP-9:** Prediction = x0\*β

**STEP-10:** Stop the program.

**PROGRAM:**

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

data = pd.read\_csv('10-dataset.csv')

bill = np.array(data.total\_bill)

tip = np.array(data.tip)

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))

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('Total bill')

plt.ylabel('Tip')

plt.show();

# Output:

# 