INTELLIGENT SYSTEMS

LAB-6

NAME: UMANG TIWARI

SEC: A

REG NO.: 201800591

ROLL: 53

PROBLEM STATEMENT

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

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

# Lab Assignment:1. Repeat the same experiment for different point (.5 to .05) and tau (10 to 100)

# Lab Assignment: 2 Perform LOESS or LOWESS using the dataset given below

1. Submit pdf with code and graph.

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

3. Analyze with different test cases.

PROBLEM SOLUTION

SOURCE CODE

import numpy as np

import matplotlib.pyplot as plt

import plotly.express as px

import plotly.graph\_objects as go

from sklearn.linear\_model import LinearRegression

from statsmodels.nonparametric.smoothers\_lowess import lowess

*# ASSIGNMENT 1*

def wm(point, X, tau):

    m = X.shape[0]

    w = np.mat(np.eye(m))

    for i in range(m):

        xi = X[i]

        d = (-2 \* tau \* tau)

        w[i, i] = np.exp(np.dot((xi-point), (xi-point).T)/d)

    return w

def predict(X, y, point, tau):

    m = X.shape[0]

    X\_ = np.append(X, np.ones(m).reshape(m,1), axis=1)

    point\_ = np.array([point, 1])

    w = wm(point\_, X\_, tau)

    theta = np.linalg.pinv(X\_.T\*(w \* X\_))\*(X\_.T\*(w \* y))

    pred = np.dot(point\_, theta)

    return theta, pred

def plot\_predictions(X, y, tau, nval):

    X\_test = np.linspace(-3, 3, nval)

    preds = []

    for point in X\_test:

        theta, pred = predict(X, y, point, tau)

        preds.append(pred)

    X\_test = np.array(X\_test).reshape(nval,1)

    preds = np.array(preds).reshape(nval,1)

    plt.plot(X, y, 'b.')

    plt.plot(X\_test, preds, 'r.')

    plt.show()

X = np.random.randn(1000,1)

y = 2 \* (X \*\* 3) + 10 + 4.6 \* np.random.randn(1000, 1)

plt.plot(X, y, 'b.')

plt.show()

for i in [0.5, 0.1, 0.05]:

    for j in [10, 50, 100]:

        print("\n\nTau: {}\nPoints: {}".format(i, j))

        plot\_predictions(X, y, i, j)

*# ASSIGNMENT 2*

X = np.linspace(-3, 3, 1000)

print(X.shape)

X += np.random.normal(scale = 0.05, size = 1000)

Y = np.log(np.abs((X\*\*2) - 1) + 0.5)

print(Y.shape)

plt.scatter(X, Y, alpha = 0.32)

plt.show()

X\_Reshape = X.reshape(-1, 1)

model = LinearRegression()

LR = model.fit(X\_Reshape, Y)

x\_range = np.linspace(X\_Reshape.min(), X\_Reshape.max(), 20)

y\_range = model.predict(x\_range.reshape(-1, 1))

y\_hat = lowess(Y, X)

fig = px.scatter(x = X, y = Y, opacity = 0.8, color\_discrete\_sequence=['black'])

fig.add\_traces(go.Scatter(x = x\_range, y=y\_range, name='Linear Regression', line=dict(color='limegreen')))

fig.add\_traces(go.Scatter(x = y\_hat[:,0], y=y\_hat[:,1], name = 'LOWESS Smoothening', line=dict(color='red')))

fig.update\_layout(dict(plot\_bgcolor = 'white'))

fig.update\_xaxes(showgrid=True, gridwidth=1, gridcolor='lightgrey', zeroline=True, zerolinewidth=1, zerolinecolor='lightgrey', showline=True, linewidth=1, linecolor='black')

fig.update\_yaxes(showgrid=True, gridwidth=1, gridcolor='lightgrey', zeroline=True, zerolinewidth=1, zerolinecolor='lightgrey', showline=True, linewidth=1, linecolor='black')

fig.update\_layout(title=dict(text="LOWESS on DataSet", font=dict(color='black')))

fig.update\_traces(marker=dict(size=3))

fig.show()

OUTPUT:

Performing necessary imports

Graphical user interface, text

Description automatically generated

Experiment for different point (.5 to .05) and tau (10 to 100)

Graphical user interface, text

Description automatically generated

Text

Description automatically generated

Plotting

Text

Description automatically generated

Graphical user interface, chart, histogram

Description automatically generated

Chart, histogram

Description automatically generated

Chart, histogram

Description automatically generated

Chart, histogram

Description automatically generated

Chart, histogram

Description automatically generated

Performing LOWESS

Graphical user interface, text

Description automatically generated with medium confidence

Graphical user interface, text, application

Description automatically generated

Plotting

A screenshot of a computer

Description automatically generated with medium confidence

Chart, line chart

Description automatically generated