ENGR 421: Introduction to Machine Learning
Fall 2020 – Homework 4
Ege Yelken – 61742

The aim of this assignment is to implement 3 nonparametric regression algorithms, namely Regressogram, Running Mean Smoother and Kernel Smoother. I have followed the steps below:

1. Divided the data set into two by assigning the first 100 data points to the training set and the rest to the test set for x and y:

```
data = np.array(pd.read_csv("hw04_data_set.csv", header=None))[1:]
x_data = data[:, 0]
y_data = data[:, 1]

x_train = x_data[0:100]
y_train = y_data[0:100]

x_test = x_data[100:]
y_test = y_data[100:]
```

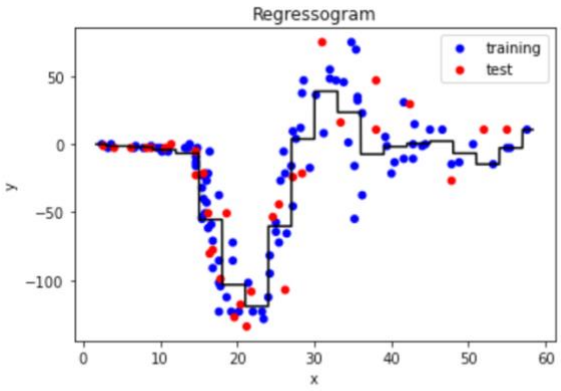
2. Implemented the Regressogram and calculated the RMSE for it:

$$\hat{g}(x) = \frac{\sum_{t=1}^{N} w\left(\frac{x-x^t}{h}\right) r^t}{\sum_{t=1}^{N} w\left(\frac{x-x^t}{h}\right)}$$

where

$$w(u) = \begin{cases} 1 & \text{if } |u| < 1 \\ 0 & \text{otherwise} \end{cases}$$

```
def regressogram(x):
    bin_width = 3
    origin = 0
    index = int((x - origin) / bin_width)
    left = origin + index * bin_width
    right = origin + (index + 1) * bin_width
    values = y_train[np.where((x_train >= left) & (x_train < right))]
    sum_val = np.sum(values)
    return sum_val / len(values)
```

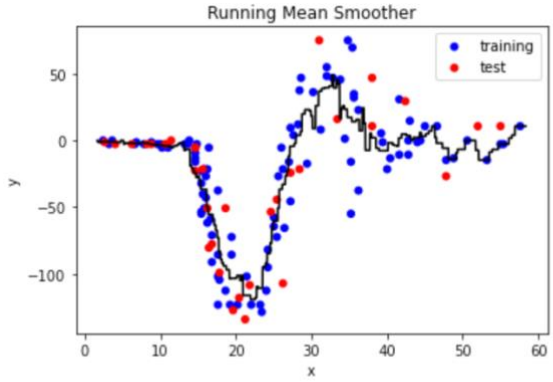



```
Regressogram => RMSE is  24.259855270587714 when h is 3 .
```

3. Implemented the Running Mean Smoother and calculated the RMSE for it:

$$\hat{g}(x) = \frac{\sum_t K\left(\frac{x-x^t}{h}\right) r^t}{\sum_t K\left(\frac{x-x^t}{h}\right)}$$

```
def running_mean_smoother(x):
    bin_width = 3
    left = (x - bin_width / 2)
    right = (x + bin_width / 2)
    values = y_train[np.where((x_train >= left) & (x_train <= right))]
    sum_val = np.sum(values)
    return sum_val / len(values)
```

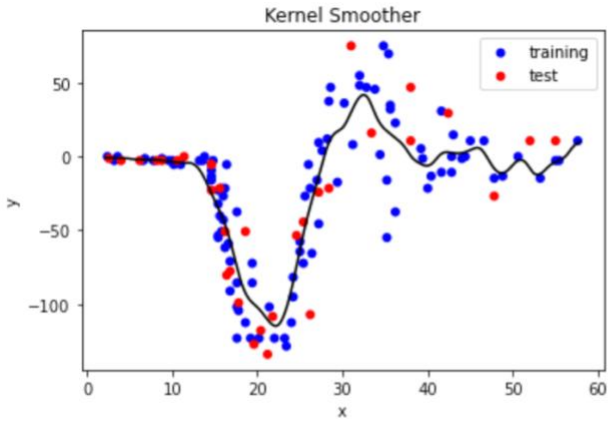



```
Running Mean Smoother => RMSE is  23.840322905172517 when h is 3 .
```

4. Implemented the Kernel Smoother and calculated the RMSE for it:

```
Gaussian_Kernel = 1/np.sqrt(2* math.pi) * np.exp(-1* u ** 2 / 2)
```

```
def kernel_smoother(x):
    bin_width = 1
    k = (x_train - x) / bin_width
    gauss_k = (1 / np.sqrt(2 * np.pi)) * np.exp(-(k * k) / 2)
    return np.dot(gauss_k, y_train) / np.sum(gauss_k)
```




```
Kernel Smoother => RMSE is  24.167246266469398 when h is 1 .
```