ENGR 421 / DASC 521: Introduction to Machine Learning
Homework 03: Nonparametric Regression

İzel Yazıcı - 77549

I read the csv and split data into two dataset which first 150 rows of data as a train and last 122 rows of the data a test data. The data was a univariate regression dataset. It was about duration of the eruption and waiting time between eruption for a geyser.

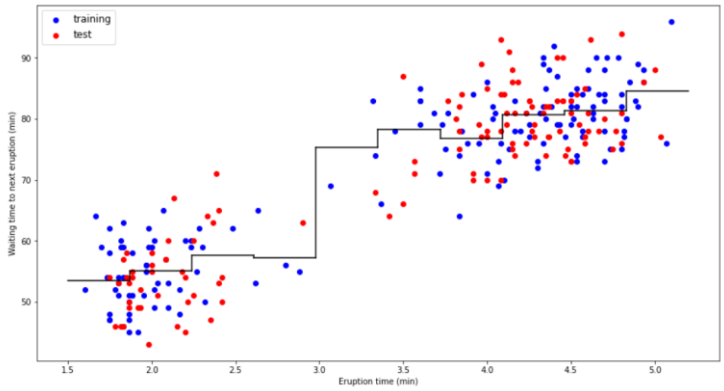
Regressogram

$$\hat{g}(x) = \frac{\sum_{t=1}^{N} b(x, x^t) r^t}{\sum_{t=1}^{N} b(x, x^t)}$$

where

$$b(x, x^t) = \begin{cases} 1 & \text{if } x^t \text{ is the same bin with } x \\ 0 & \text{otherwise} \end{cases}$$

I implemented regressogram formula by setting the bin width parameter to 0.37 and the origin parameter to 1.5 . The implementation graph is the following.



RMSE of Regressogram

$$\sqrt{\frac{\sum_{i=1}^{N_{test}} (y_i - \hat{y}_i)^2}{N_{test}}}.$$

5. Calculate the root mean squared error (RMSE) of your regressogram for test data points.

```
rmse = np.asarray([(np.sum(np.square(p_hat[b] - (y_test[(left_borders[b] < x_test) & (x_test <= right_borders[b])])))for b in
rmse= np.nan_to_num(rmse)
rmse= np.sqrt(np.sum(rmse) / len(x_test))
print(f'Running Regressogram => RMSE is {rmse} when h is {bin_width}')
```

```
Running Regressogram => RMSE is 5.962617204275406 when h is 0.37
```

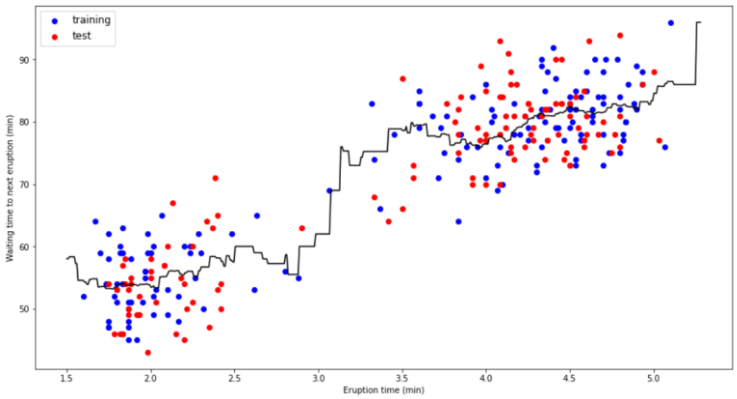
Mean Smoother

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

where

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

I implemented Mean Smoother formula by setting the bin width parameter to 0.37 Mean Smoother Graph is the following:



RMSE of Mean Smoother

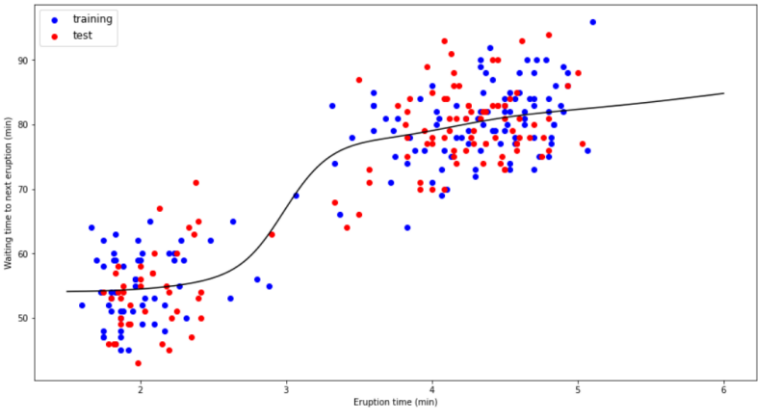
```
origin=1.5
error = [(y_test[i] - p_hat2[int((x_test[i]-origin)*x_test.shape[0])])**2 for i in range(len(x_test))]
error= np.nan_to_num(error)
rmse = np.sqrt(np.sum(error) / len(x_test))
print("Mean Smoother => RMSE is", rmse, " when h is", bin_width)
```

```
Mean Smoother => RMSE is 6.093749038108317  when h is 0.37
```

Kernel Smoother

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

I implemented Kernel Smoother formula by setting the bin width parameter to 0.37 Kernel Smoother Graph is the following:



RMSE of Kernel Smoother

```
origin = 1.5
error = [(y_test[i] - p_hat3[int((x_test[i]-origin)*x_test.shape[0])])**2 for i in range(len(x_test))]
error= np.nan_to_num(error)
rmse = np.sqrt(np.sum(error) / len(x_test))
print("Kernel Smoother => RMSE is", rmse, " when h is", bin_width)
```

```
Kernel Smoother => RMSE is 5.856645912298686  when h is 0.37
```