ENGR421

HW4

Doğa Demirtürk
68859

In this homework, we implement three nonparametric regression algorithms in Python. To accomplish this, I first read the related topics from the book. Then I started implementing by reading data into the memory and divide it into two groups: first 150 data points for training set and last 122 data points for test set.

The first nonparametric regression algorithm I implemented is regressogram with the bin width parameter 0.37 and the origin parameter 1.5. I started by assigning the parameters and determining the maximum value. Then I created two variables to hold the borders of the lines: left borders and right borders. They are created such that every line has width bin width. Then I calculated the p hat values by using the formula given in the cheat sheet.

$$g(x) = \frac{\sum_{i=1}^{N} b(x, x_i) y_i}{\sum_{i=1}^{N} b(x, x_i)}$$

where

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

Figure 1: Regressogram formula from cheat sheat

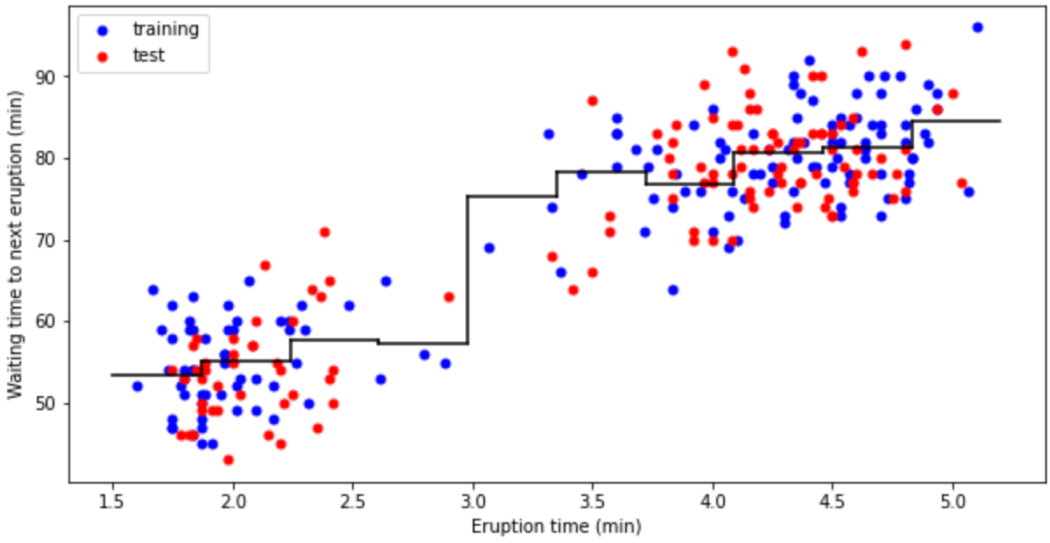

Figure 2: Plotted graph with Regressogram

After, I calculated RMSE value with use of an additional function calculate_RMSE. This function takes predicted y values as input and calculates RMSE for each algorithm.

```
Regressogram => RMSE is 5.962617204275405 when h is 0.37
```

Figure 3: RMSE output for Regressogram

The second algorithm is running mean smoother algorithm. To implement this algorithm, I created a data interval with 0.0001 width. I calculated the p hat values with the use of the cheat sheet. Again, I calculated the RMSE with the defined function.

$$g(x) = \frac{\sum_{i=1}^{N} w\left(\frac{x - x_i}{h}\right) y_i}{\sum_{i=1}^{N} w\left(\frac{x - x_i}{h}\right)}$$

where

$$w(u) = \begin{cases} 1 & \text{if } |u| \leq 1/2 \\ 0 & \text{otherwise} \end{cases}$$

Figure 4: Running Mean Smoother formula from cheat sheet

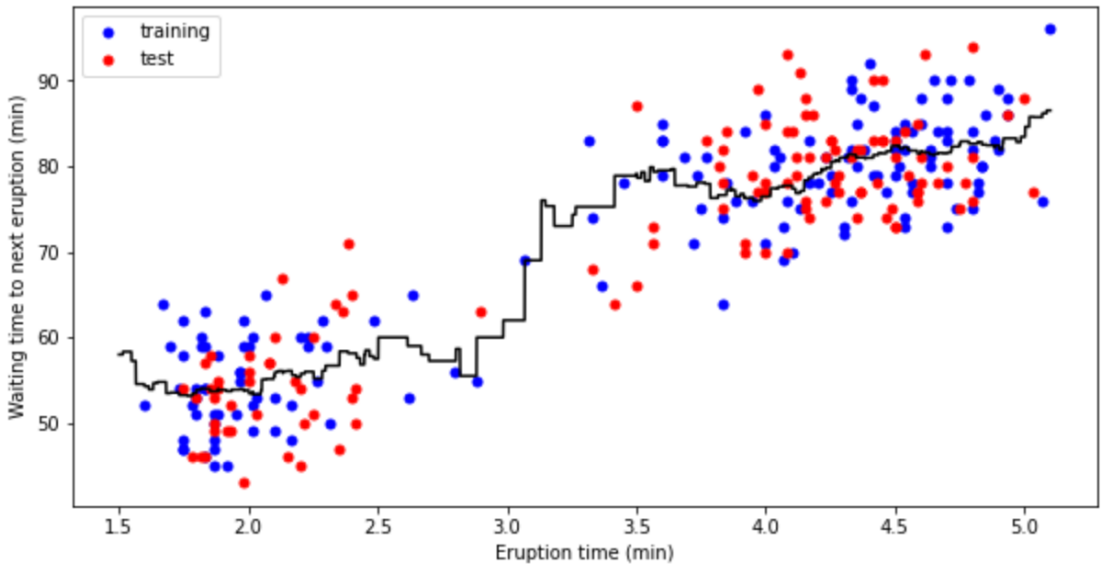

Figure 5: Plotted graph with Running Mean Smoother

```
Running Mean Smoother => RMSE is 6.089003211720321 when h is 0.37
```

Figure 6: RMSE output for Running Mean Smoother

Finally, the third algorithm was kernel smoother. Again, I used the same data interval I used in the mean smoother algorithm. I added a function called K to calculate the K value in the formula given in the cheat sheet. I used the calc_RMSE function to find RMSE of the algorithm.

$$g(x) = \frac{\sum_{i=1}^{N} K\left(\frac{x - x_i}{h}\right) y_i}{\sum_{i=1}^{N} K\left(\frac{x - x_i}{h}\right)}$$

where

$$K(u) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{u^2}{2}\right)$$

Figure 7: Kernel Smoother formula from cheat sheet

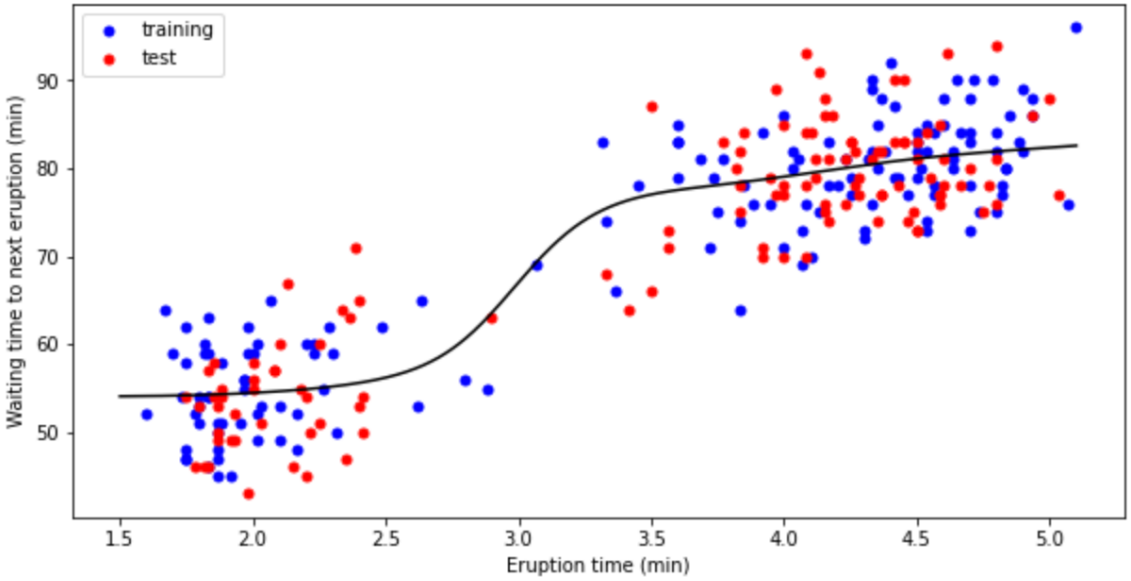

Figure 8: Plotted graph with Kernel Smoother

```
Kernel Smoother => RMSE is 5.874362846844901 when h is 0.37
```

Figure 9: RMSE output for Kernel Smoother