ENGR421

HW5

Doğa Demirtürk
68859

In this homework, we implement a decision tree regression algorithm. In our tree, we use pre-prunning rule in which we call a node terminal node if the node has P or less data points.

I started by reading the dataset into memory and dividing it to two parts: first 150 data points for training set and remaining 122 data points for test set.

After that, I implemented a learn_tree method that takes inputs x, y and P, prepares the decision tree and returns the node_splits, node_means and is_terminal values which are necessary for predictions. I implemented this method with use of the tree algorithm we used in the lab 7. I changed some parts of it such as I removed node_frequencies since in our problem y values are not discrete and node_features since we have one feature x for each data point. I added node_means since the y values are continuous. For best split selection, I calculated the split scores using entropy for each possible split and chose the smallest one among them as best split.

Then I implemented predict method which uses the returned values of the learn_tree method to find which node given data point belongs to in order to predict its value. Also, I implemented calc_RMSE method which calculates RMSE values for given y predicted and y truth values. I used the calc_RMSE method I previously implemented in the homework 4.

I draw the desired figure in step 4 by using the learn_tree and predict methods using P value as 25. It is same as the figure given in the homework description file.

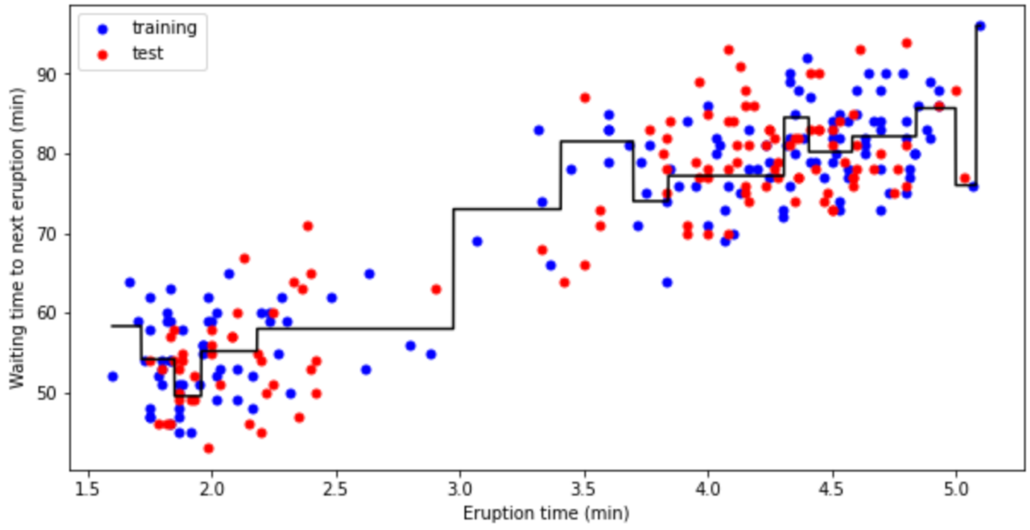
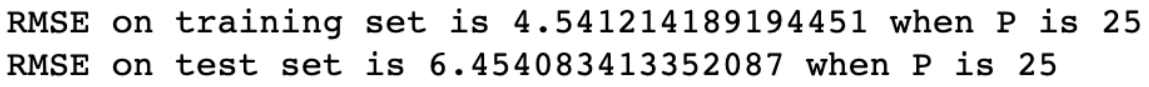


Figure 1: Training data points, test data points and fit using decision tree with P = 25

Then, I calculated RMSE values for training and test datasets. They are also same as the expected outputs given in the homework description file.



```
RMSE on training set is 4.541214189194451 when P is 25
RMSE on test set is 6.454083413352087 when P is 25
```

Figure 2: RMSE values on training and test sets with P = 25

Finally, I calculated RMSE values for different P values (5, 10, 15, … up to 50) and draw them separately for training and test sets. To accomplish this, I used all the methods I mentioned above. The drawn figure is again the same as in the description file.

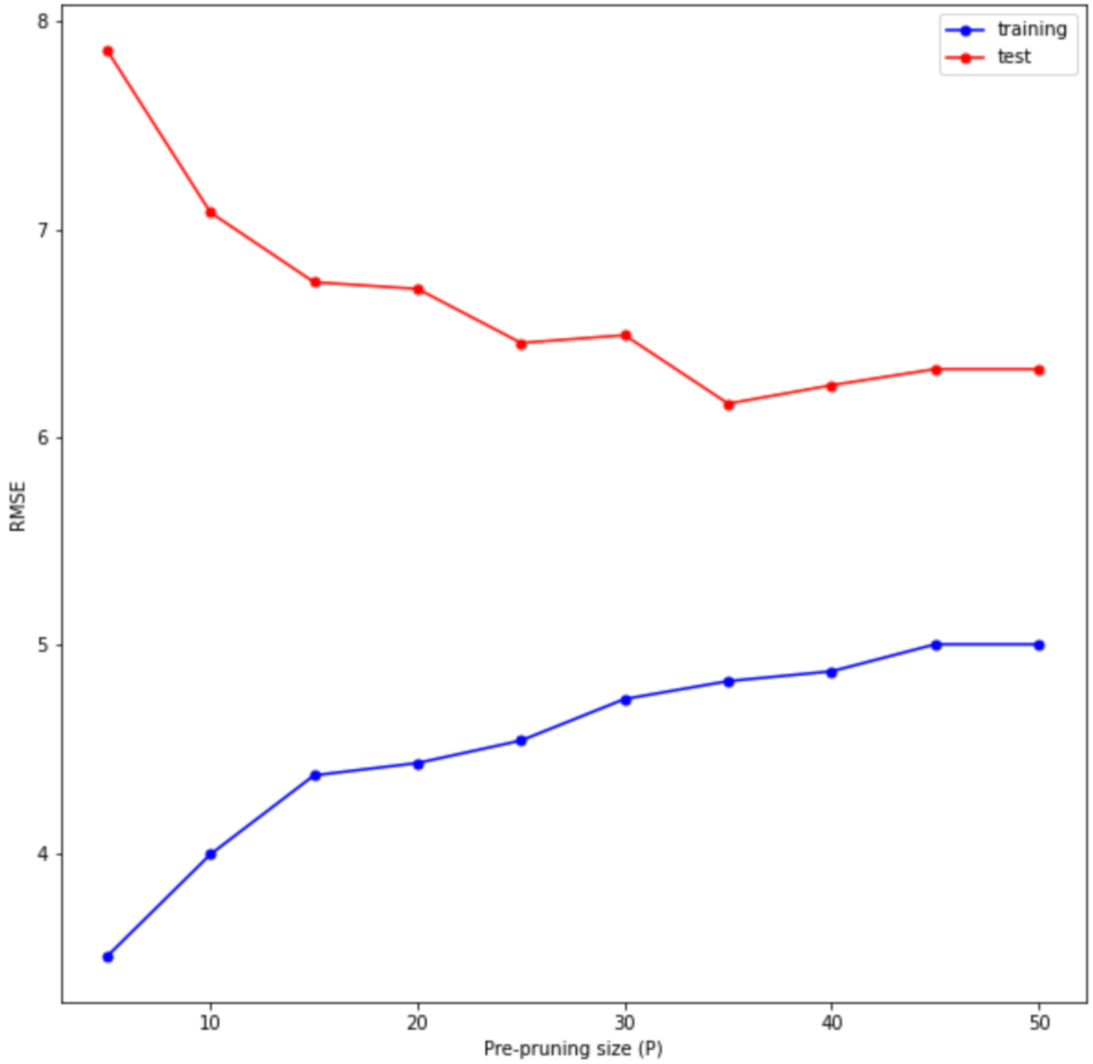


Figure 3: RMSE for training and test data points as a function of P