In this homework, I implemented a decision tree regression algorithm for univariate data.

Part 1)

Read Data from hw05_data_set.csv

Part 2)

Data is divided into train and test set.

150 data points to the training set and the remaining 122 data points to the test set.

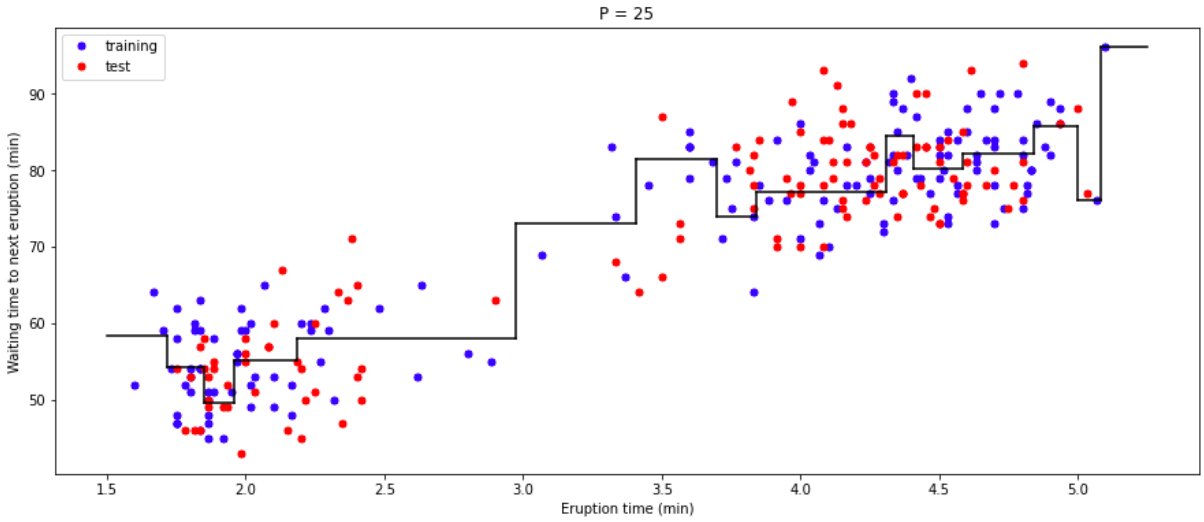
Part 3)

I implemented a decision tree regression algorithm using pre-pruning rule which is: If a node has P or fewer data points, convert this node into a terminal node and do not split further. I used lab07 code and made some changes on it. In lab code we had iteration for features. Now we have 1.

In this part, tried to estimate y values using the x values of the given data points using the decision tree regression method. For each node, determine possible split points and calculate their scores values which is error. Select best split which has min score. Split from split point and do this again until it is pure or number of data in a node is less than 25. After terminal node no more splitting will be needed.

Part 4)

In this part give pre pruning value as 25 and run defined decision tree algorithm accordingly. Then Draw training data points, test data points, and fit in the same figure.



Part 5)

I defined RMSE function with given formula to use in this step later and later steps. In this step I calculated RMSE values to understand how well our estimation.

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

```
def calculateRMSE(N,x,y,p_hat):
    sum_regress = 0
    for i in range(N):
        for a in range(len(left_borders)):
            if (left_borders[a] < x[i]) and (x[i] <= right_borders[a]):
                sum_regress += (y[i] - p_hat[a])**2
    rmse_regress = math.sqrt(sum_regress/N)
    return(rmse_regress)
```

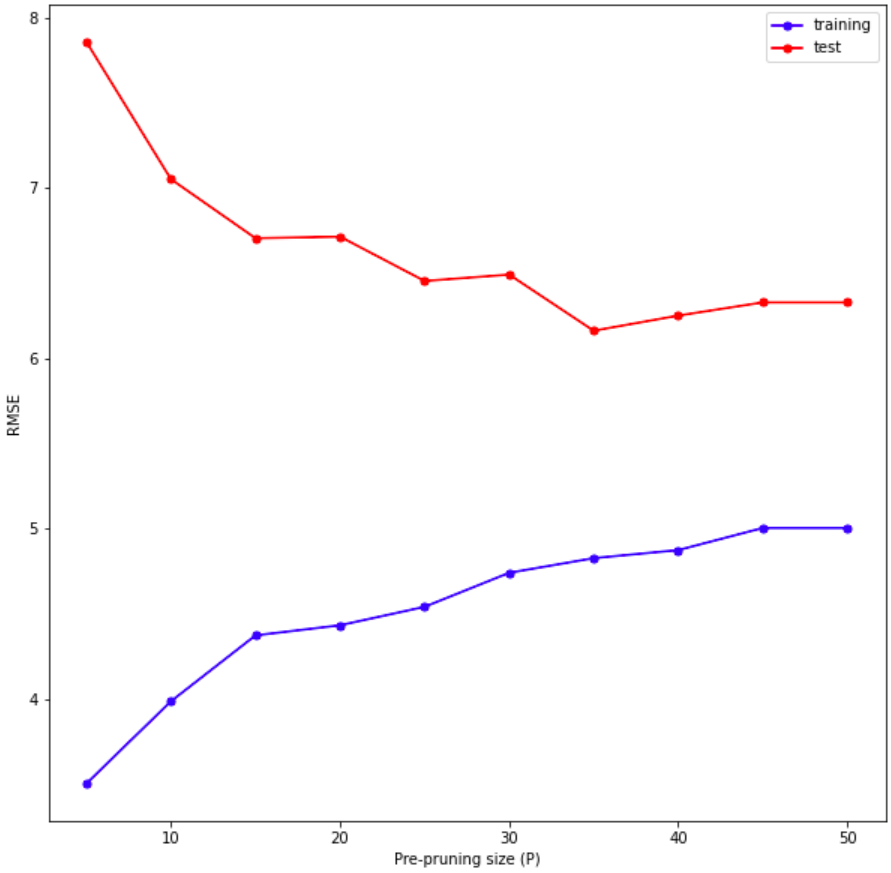
```
print("RMSE on training set is", calculateRMSE(N_train,x_train,y_train,p_hat), "when P is 25"
print("RMSE on test set is", calculateRMSE(N_test,x_test,y_test,p_hat), "when P is 25")
```

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

Part 6)

Learn decision trees by setting the pre-pruning parameter P to 5, 10, 15, ..., 50. I used function which is defined in step 3. Draw RMSE for training and test data points as a function of P.

In this step we compared RMSE values to understand how model changes with different pre-pruning values. For small p values we are overfitting our data and force most nodes to be pure. Hence increasing p values increases RMSE of the training data. However, in test data until p value is 35, we are increasing our model. After p>35 we are underfitting data. That's why best value for P is 35.


training
test
RMSE
Pre-pruning size (P)
8
7
6
5
4
10
20
30
40
50