Aykut Aykut
0034039
November 29th, 2018
Comp421 Homework05

In this homework, I was asked to implement a decision tree regression algorithm with pre-pruning parameter P. In order to do that , I followed these 11 steps:

1) I read the dataset into memory as *data*. The data has 133 rows and 2 columns which are 'x' and 'y'. Each row corresponds to a data point.

2) Train-test split: I splitted the data into 2 groups, train and test. First 100 rows are in *train_data* and last 33 rows are in *test_data.* Then, I splitted 'x' and 'y' columns of both *train_data* and *test_data* into *train_x, train_y, test_x* and *test_y.*

3) I calculated the length of *train_y* and *test_y* to find out how many train and test samples I have.

4) I defined *node_impurity* function with *data_y* input in order to calculate the impurity of given data in a node using the following formula:
$E_m(y) = \frac{\sum_{i=1}^{N} (y_i - g)^2}{N}$ *where* $g = \frac{\sum_{i=1}^{N} y_i}{N}$ *and N is the lenght of y vector*

5) I defined *split_impurity* function with *left_data_y* and *right_data_y* inputs in order to calculate the impurity of a split using the following formula:
$E'_m(y_{left}, y_{right}) = \frac{E_m(y_{left}) * N_{left} + E_m(y_{right}) * N_{right}}{N_{left} + N_{right}}$
where N_{left} *and* N_{right} *are the length of* y_{left} *and* y_{right} *vectors respectively*

6) I defined *decision_tree_fit_test* function with *P, plot_fit = FALSE* and *extract_rule = FALSE* inputs where *P* is the pre-pruning limit parameter, *plot_fit* controls whether to draw data points and decision tree fit or not, it is false by default, and *extract_rule* controls whether to extract and print the fitted decision tree rules or not, it is also false by default.
    a) First, I created the necessary data structures *node_indices* to store data point indices at each node in the decision tree, *is_terminal* to store the information whether a node is terminal or not, *need_split* to store the information which nodes need splitting further, *node_splits* to store the split position at each non-terminal node, *node_means* to store the means of y values of data points at each node.
    b) I initialized the decision tree by putting all *train_x* indices into *node_indices* since its first element is the root node. Since the first element is always the root node, it is not terminal and need split. Therefore, *is_terminal* is initialized as *c(FALSE)* and *need_split* is initialized as *c(TRUE)*.

c) Then, I implemented the decision tree algorithm. I first find out which nodes in tree needs split and check whether it is empty or not. If there is no node needs split, then it means all nodes at leafs are terminal nodes in the tree, and the tree is fitted to data. If there are nodes need split, then I one by one iterate over them to split them or make them a terminal node if either the node has equal or fewer data points than *P* or the data points cannot be splitted further. If the node is not a terminal node, then I find the split positions and iterate over them. At each split position, I simulate the splitting and calculate split impurity. After that, I split the node at the split position whose split impurity is the smallest, create new 2 nodes as left and right child of the current node and record the split position. During the splitting process, I assign data points whose 'x' is less than or equal to the split position to the left child and others to the right child.

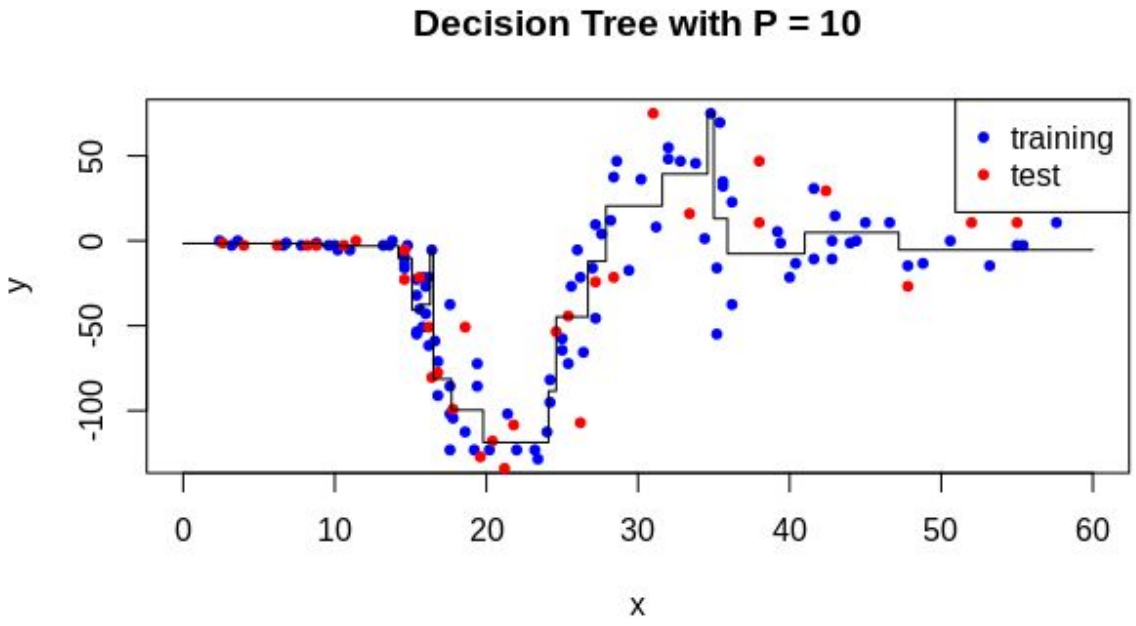
d) If *plot_fit* is TRUE, then I draw the *train_data, test_data* and fitted decision tree.

e) If *extract_rule* is TRUE, then I extract the rules used while splitting and print them on the console.

f) Finally, I test the decision tree with *test_data,* calculate RMSE of it and return the RMSE calculated.

7) I set *P* to 10.

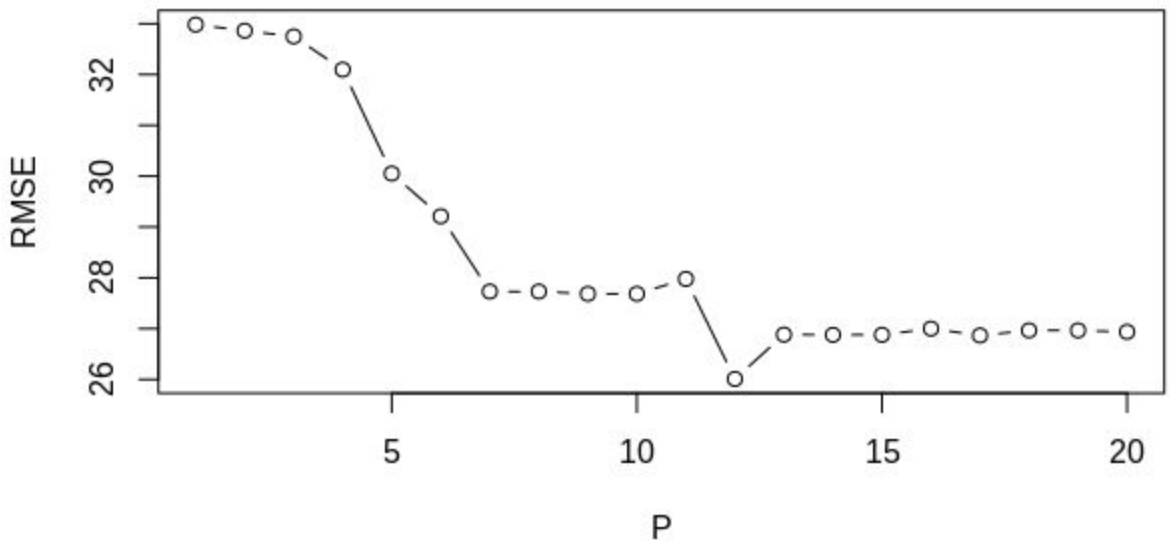
8) I called the *decision_tree_fit_test* function with *P, plot_fit = TRUE* and *extract_rule = TRUE* inputs. *decision_tree_fit_test* function first fits a decision tree, then plots the figure of data points and fit, and extracts splitting rules. I printed RMSE of *test_data* in the fitted decision tree. The figure of decision tree with *P* = 10 and RMSE of *test_data* is as follows:




```
> cat('RMSE is', test_data_rmse, 'when P is', P)
RMSE is 27.6841 when P is 10
```

9) I also extracted the splitting rules and they are as follows:
[1] "{x < 26.7 AND x >= 16.5 AND x >= 24.6} => [-44.8]"
[1] "{x >= 26.7 AND x < 35.9 AND x < 27.9} => [-12.025]"
[1] "{x >= 26.7 AND x >= 35.9 AND x < 41} => [-7.56666666666667]"
[1] "{x < 26.7 AND x < 16.5 AND x < 15.1 AND x >= 14.2} => [-10.325]"
[1] "{x < 26.7 AND x < 16.5 AND x >= 15.1 AND x >= 16.3} => [-5.4]"
[1] "{x < 26.7 AND x >= 16.5 AND x < 24.6 AND x < 17.7} => [-81.3142857142857]"
[1] "{x >= 26.7 AND x < 35.9 AND x >= 27.9 AND x >= 35} => [13.12]"
[1] "{x >= 26.7 AND x >= 35.9 AND x >= 41 AND x < 47.2} => [4.91111111111111]"
[1] "{x >= 26.7 AND x >= 35.9 AND x >= 41 AND x >= 47.2} =>
[-5.34285714285714]"
[1] "{x < 26.7 AND x < 16.5 AND x < 15.1 AND x < 14.2 AND x < 9.2} =>
[-1.52857142857143]"
[1] "{x < 26.7 AND x < 16.5 AND x < 15.1 AND x < 14.2 AND x >= 9.2} =>
[-3.08571428571429]"
[1] "{x < 26.7 AND x < 16.5 AND x >= 15.1 AND x < 16.3 AND x < 15.7} => [-40.7]"
[1] "{x < 26.7 AND x < 16.5 AND x >= 15.1 AND x < 16.3 AND x >= 15.7} =>
[-37.5333333333333]"
[1] "{x < 26.7 AND x >= 16.5 AND x < 24.6 AND x >= 17.7 AND x >= 24.1} =>
[-88.45]"
[1] "{x >= 26.7 AND x < 35.9 AND x >= 27.9 AND x < 35 AND x >= 34.6} => [75]"
[1] "{x < 26.7 AND x >= 16.5 AND x < 24.6 AND x >= 17.7 AND x < 24.1 AND x <
19.8} => [-99.58]"
[1] "{x < 26.7 AND x >= 16.5 AND x < 24.6 AND x >= 17.7 AND x < 24.1 AND x >=
19.8} => [-118.7]"
[1] "{x >= 26.7 AND x < 35.9 AND x >= 27.9 AND x < 35 AND x < 34.6 AND x <
31.6} => [20.55]"
[1] "{x >= 26.7 AND x < 35.9 AND x >= 27.9 AND x < 35 AND x < 34.6 AND x >=
31.6} => [39.38]"

10) I created a *P_list* list whose values are from 1 to 20, iterated over *P_list* and fitted a decision tree for each *P* values. Then I calculated RMSE of *test_data* for each decision tree and plotted P vs. RMSE figure. The figure is as follows:



11) I obtained the same results, decision tree fit with P = 10 figure, root mean squared error of *test_data* for the decision tree with P = 10 and P vs. RMSE figure with the results given in the homework description.