CS 419 - Assignment 1 Report

Implementation of the tree:

The given dataset indicated that we need to use regression trees as we didn't have discrete values.

So the step one was to determine the split criteria of the tree. A binary split was done based on the loss value of the split for each value of attributes in both the datasets. The data was sorted as per the attribute on which we were calculating the loss function of the split which greatly optimised the code as for next iteration we needed to go through fewer examples.

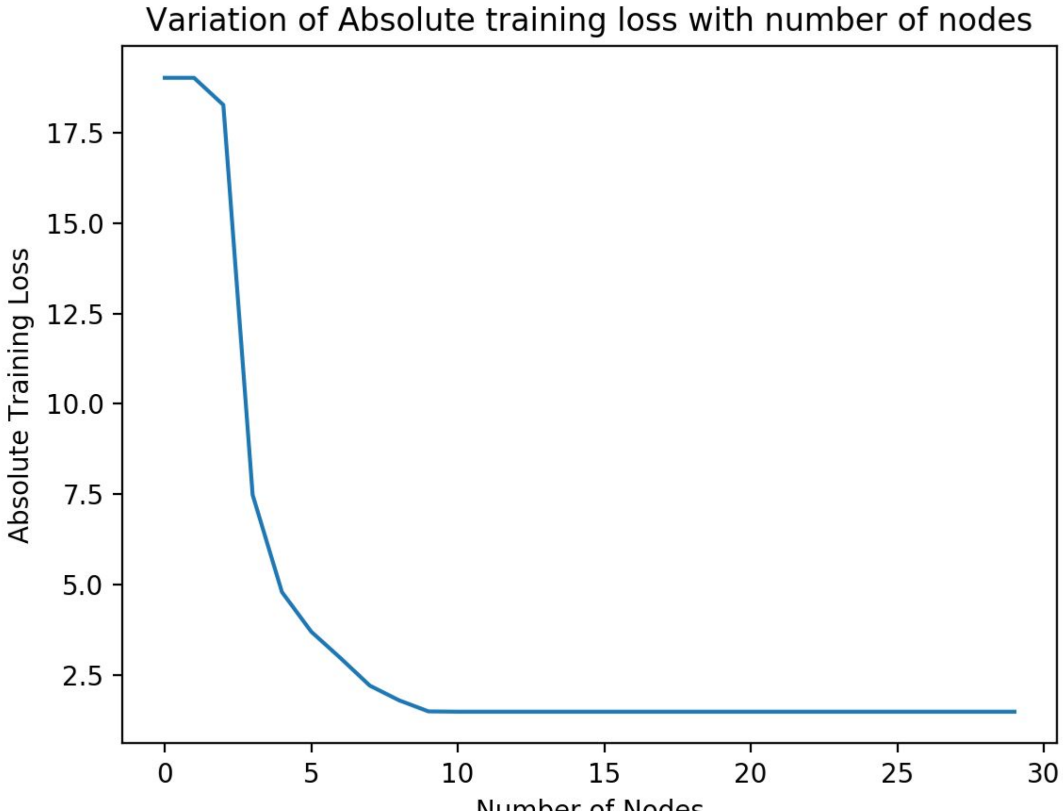
To prevent overfitting, we used early stopping based on the depth of the tree and minimum number of examples at the leaf node.

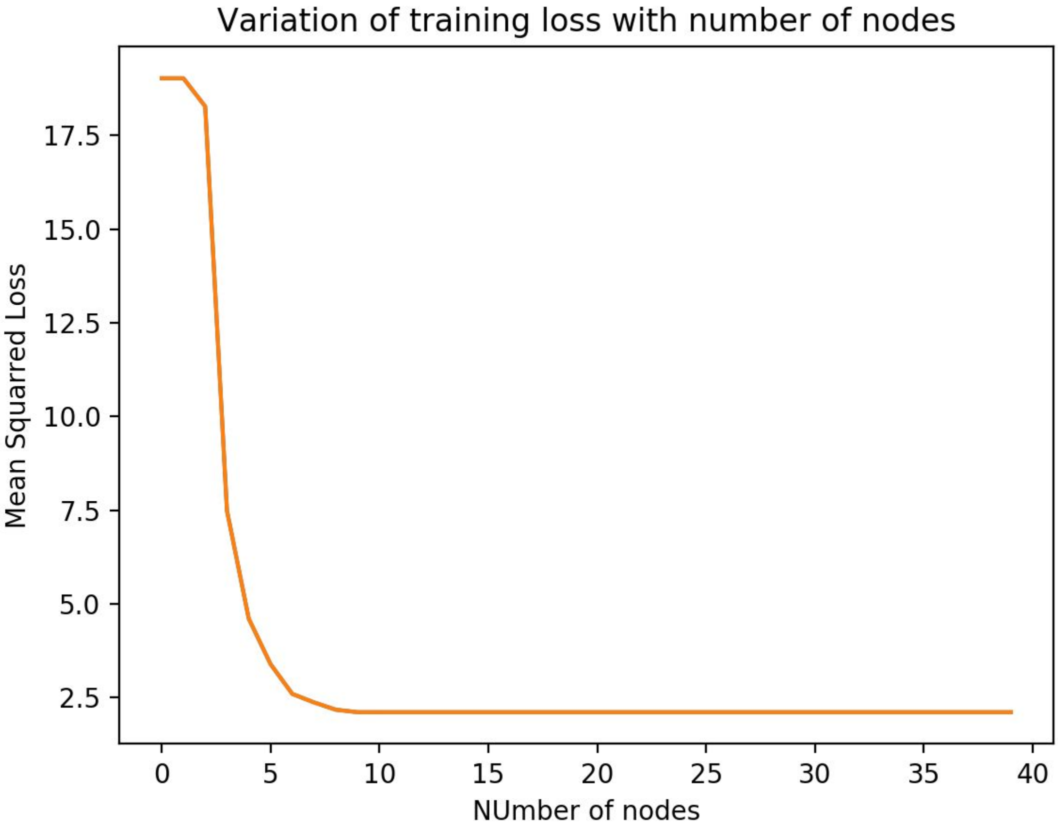
In Competition 2:

For getting the best Kaggle score, train.csv was shuffled randomly and 90% was used as training data and rest 10 % was used as testing data. Best Kaggle score was achieved with a tree of depth 25 and min_leaf_examples 3

Best training loss error : 1.4862
Time to train: 1.632 sec
Infer Time: 0.000746 sec

Below are the plots showing variation of training loss with number of nodes
(For absolute Loss: min_leaf sample = 3, For squared Loss: min_leaf sample = 3)

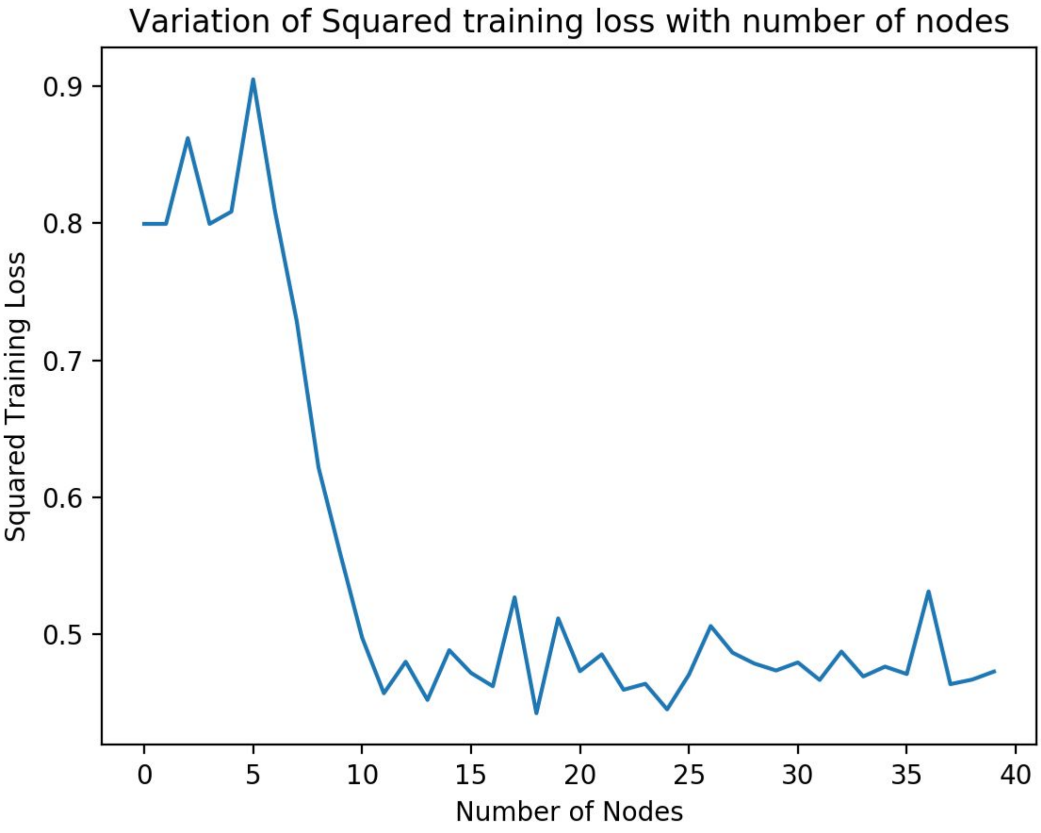


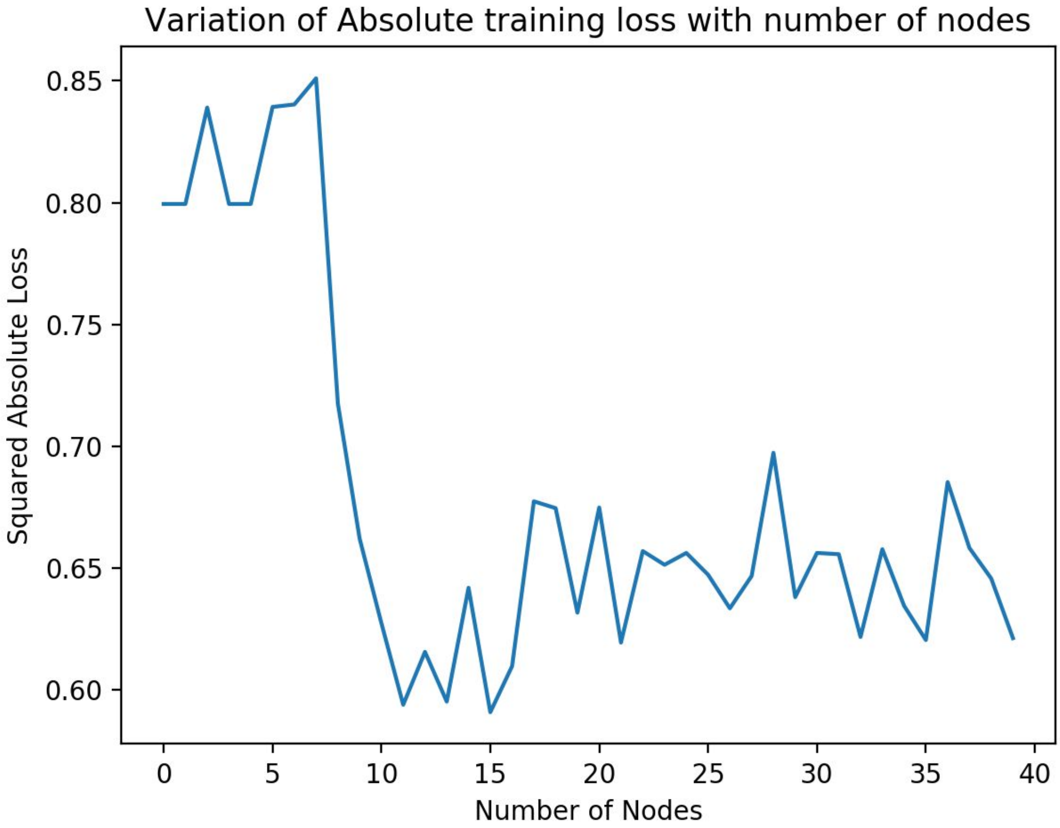

Variation of training loss with number of nodes
Mean Squarred Loss
17.5
15.0
12.5
10.0
7.5
5.0
2.5
0
5
10
15
20
25
30
35
40
NUmber of nodes

In Competition 1:

Similar approach was followed of sorting the dataset based on attribute values and building trees on randomly shuffled datasets for better kaggle score. Best Kaggle score was achieved with a tree of depth 7 and min_leaf_examples 15 and squared loss function.

Best training loss error : 0.4427
Time taken to train : 21.026 sec
Inference Time: 0.00381 sec




Variation of Absolute training loss with number of nodes
Squared Absolute Loss
0.85
0.80
0.75
0.70
0.65
0.60
0
5
10
15
20
25
30
35
40
Number of Nodes