Note: These results are generated when we run the file, when you run it there might be changes in the results as it depends on the sampling of the dataset.

**Regression Tree Generation.**

Steps:

1. Splitting the dataset in **70:30** ratio.
2. Grown the decision tree with the best split taking **max\_depth =5** and **min\_samples=1.**
3. Resultant Tree’s **Training squared error = 38750.0** and the **Testing squared error 61250.0.**
4. Resultant Tree :-

**{'Extra Spicy = 1': [{'Extra Mushroom = 1': [{' Size by Inch <= 12': [925.0,**

**900.0]},**

**{' Size by Inch <= 9': [700.0,**

**{'Extra Cheeze = 1': [725.0,**

**700.0]}]}]},**

**{' Size by Inch <= 8': [550.0,**

**{'Extra Cheeze = 1': [650.0,**

**575.0]}]}]}**

**Finding the Best Tree.**

Steps:-

1. Here firstly we defined set of parameters such as  **[max\_depth, r\_squared\_train, r\_squared\_test].**
2. Loop from **1 to 7** for **max\_depth** of the tree taking **min\_samples =1.**
3. Various trees are created with respect to the parameters and start updated the best tree on the basis of testing squared error.
4. Results:-

Resultant Best Tree:**-**

**{'Extra Spicy = 1': [{'Extra Mushroom = 1': [{' Size by Inch <= 12': [925.0,**

**900.0]},**

**{' Size by Inch <= 9': [700.0,**

**725.0]}]},**

**{' Size by Inch <= 8': [550.0,**

**{' Size by Inch <= 9': [650.0,in**

**700.0]}]}]}**

**Parameters for the Best tree**

**max\_depth min\_samples r\_squared\_train r\_squared\_test**

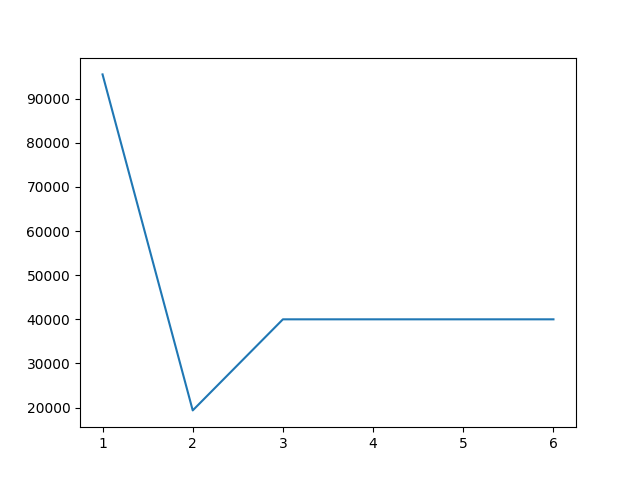
**3 1 27500.0 33125.0**

**Finding the tree depth where it overfits and variation of test accuracy with varying depths.**

Steps:

1. Split the data into training and testing as 70:30.
2. Loop the max\_depth from 1 to 7
3. Create the tree with respect to max\_depth and calculate the training squared error and testing squared error.
4. At every depth testing squared error is taken into consideration, and plot is plotted.
5. The Depth of the Tree which overfits is calculated by the minimum training squared error.

**Results:**

****

In the above figure x-axis is the number of depth and on y-axis it is total squared error. Here we can observe that the ***depth=2*** we get the minimum squared error.

And In results we found that the model gets overfits on ***depth=4.***

**Rule Post Pruning.**

Rule Post Pruning is applied to the best tree found in the Step 2.

Results that we get are these:-

Tree before pruning:-

**{'Extra Spicy = 1': [{'Extra Mushroom = 1': [{' Size by Inch <= 12': [925.0,**

**900.0]},**

**{' Size by Inch <= 9': [700.0,**

**725.0]}]},**

**{' Size by Inch <= 8': [550.0,**

**{' Size by Inch <= 9': [650.0,**

**700.0]}]}]}**

**Tree after pruning:-**

**{'Extra Spicy = 1': [850.0,**

**{' Size by Inch <= 8': [550.0,**

**{' Size by Inch <= 9': [650.0,**

**700.0]}]}]}**

***Squared Error of Tree: 1,875***

***Squared Error of Pruned Tree: 35,000***