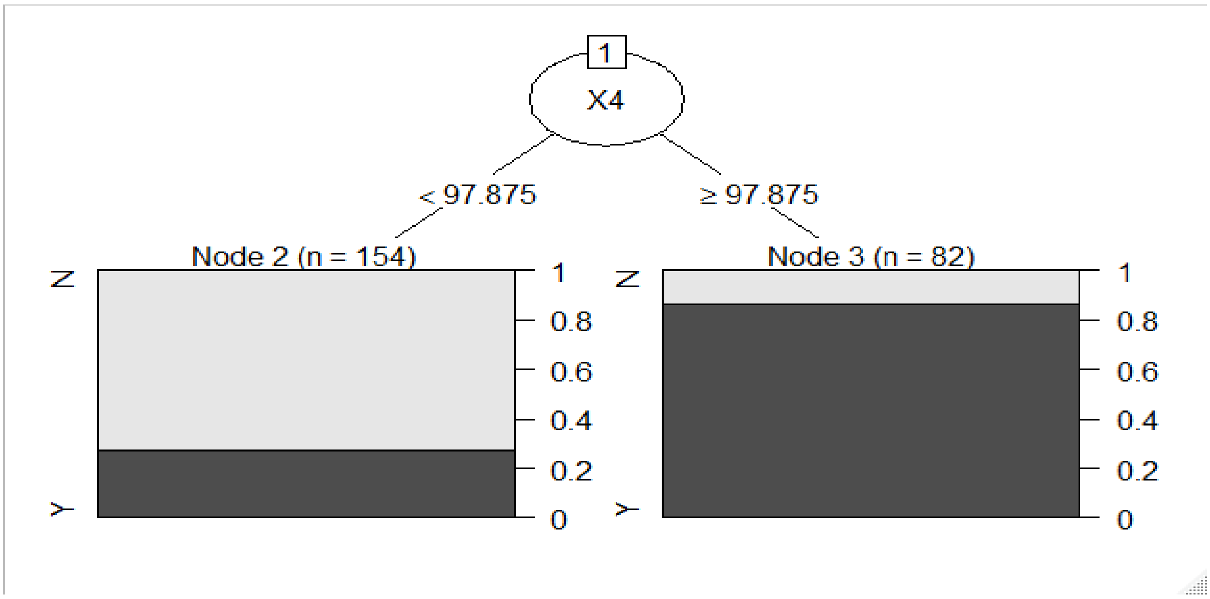
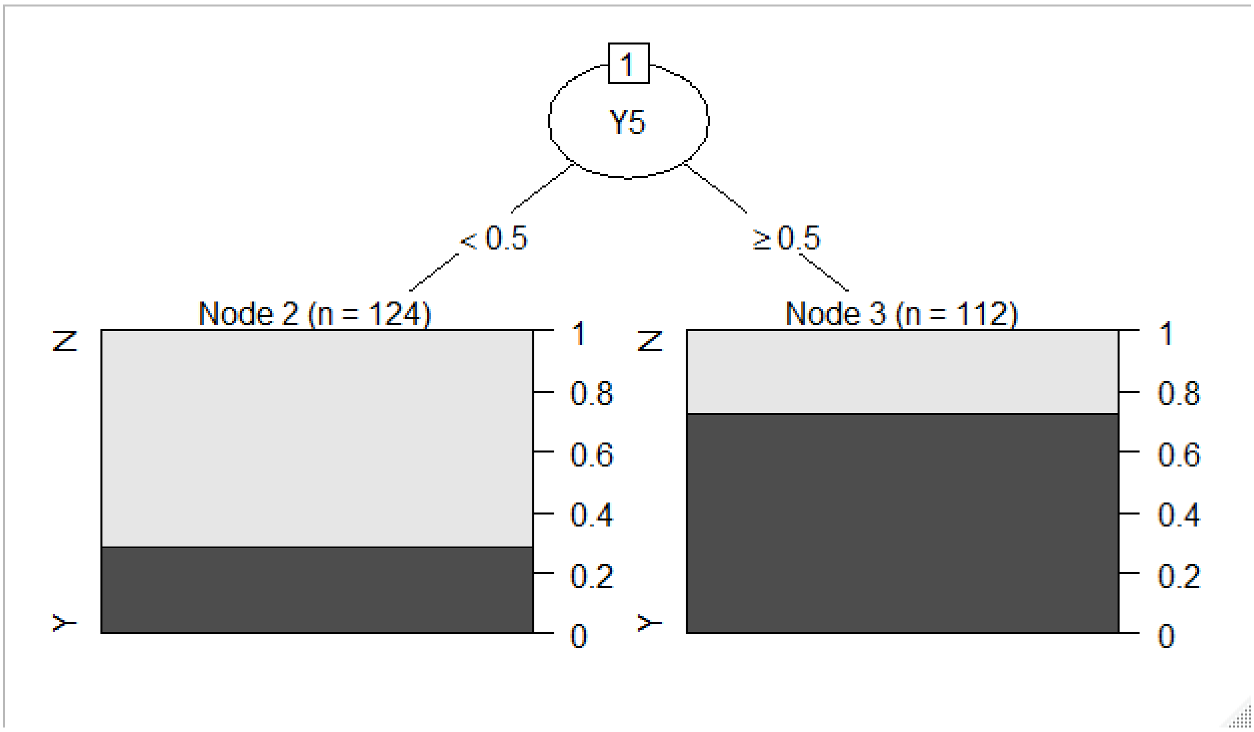
X vs Response results

Pruned model trained with 80% of data



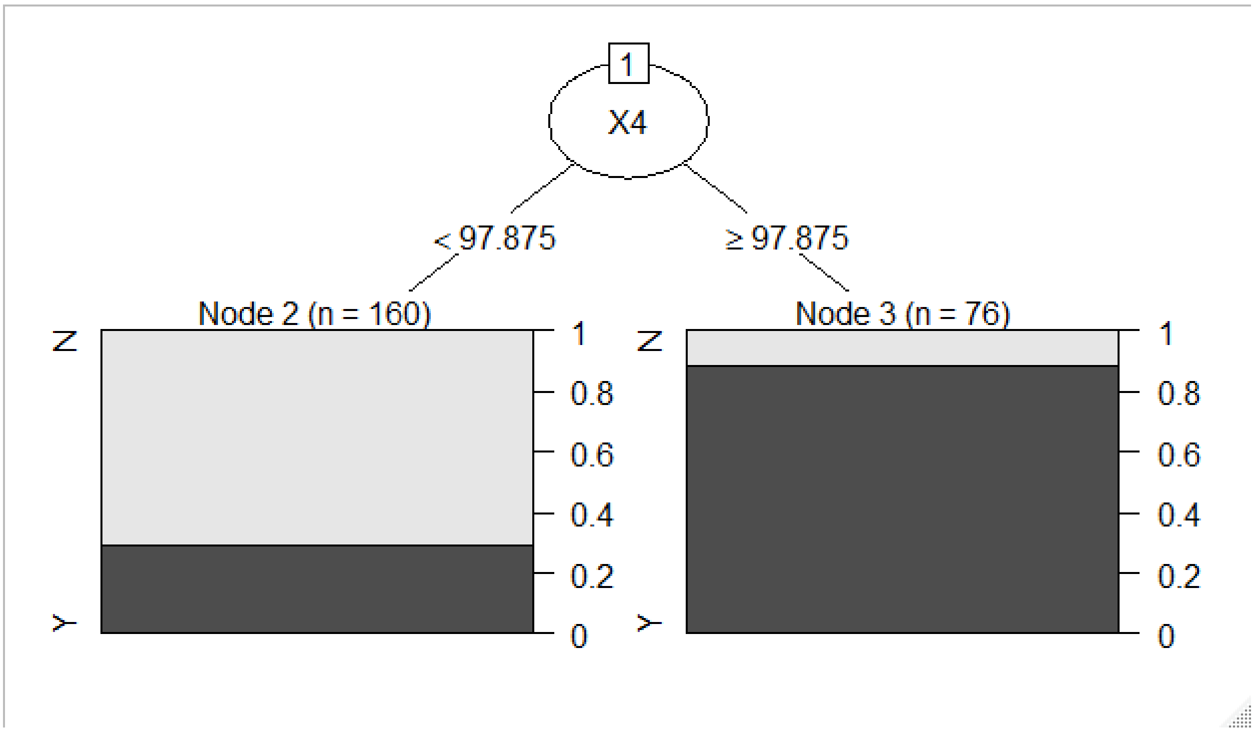
Y vs Response results

Pruned model trained with 80% of data



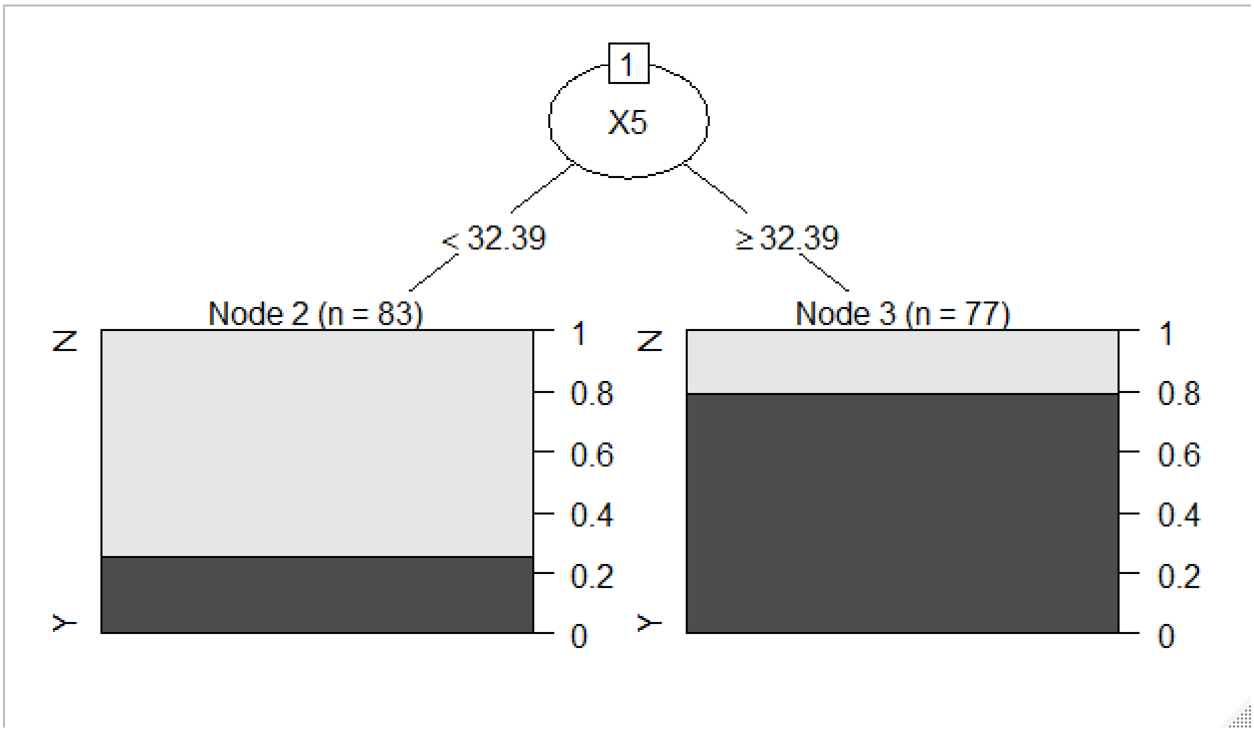
XY vs Response results

Pruned model trained with 80% of data



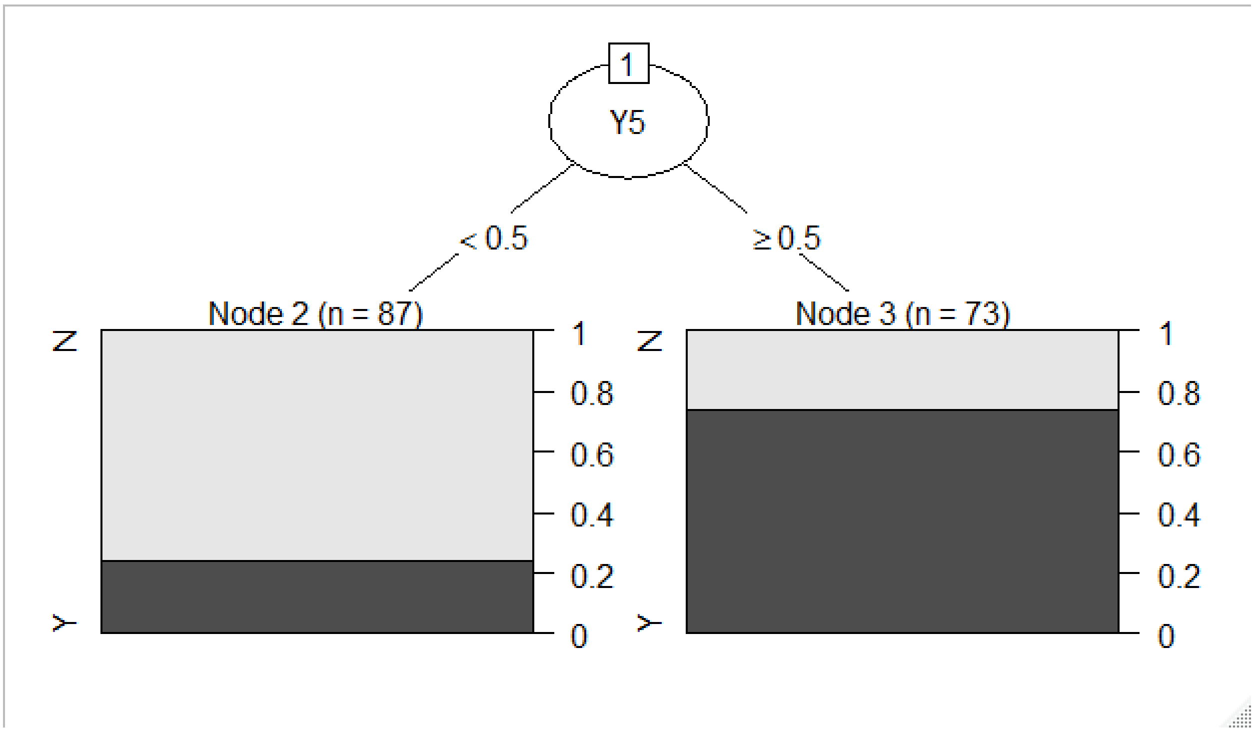
X-1 vs Response results

Pruned model trained with 80% of data



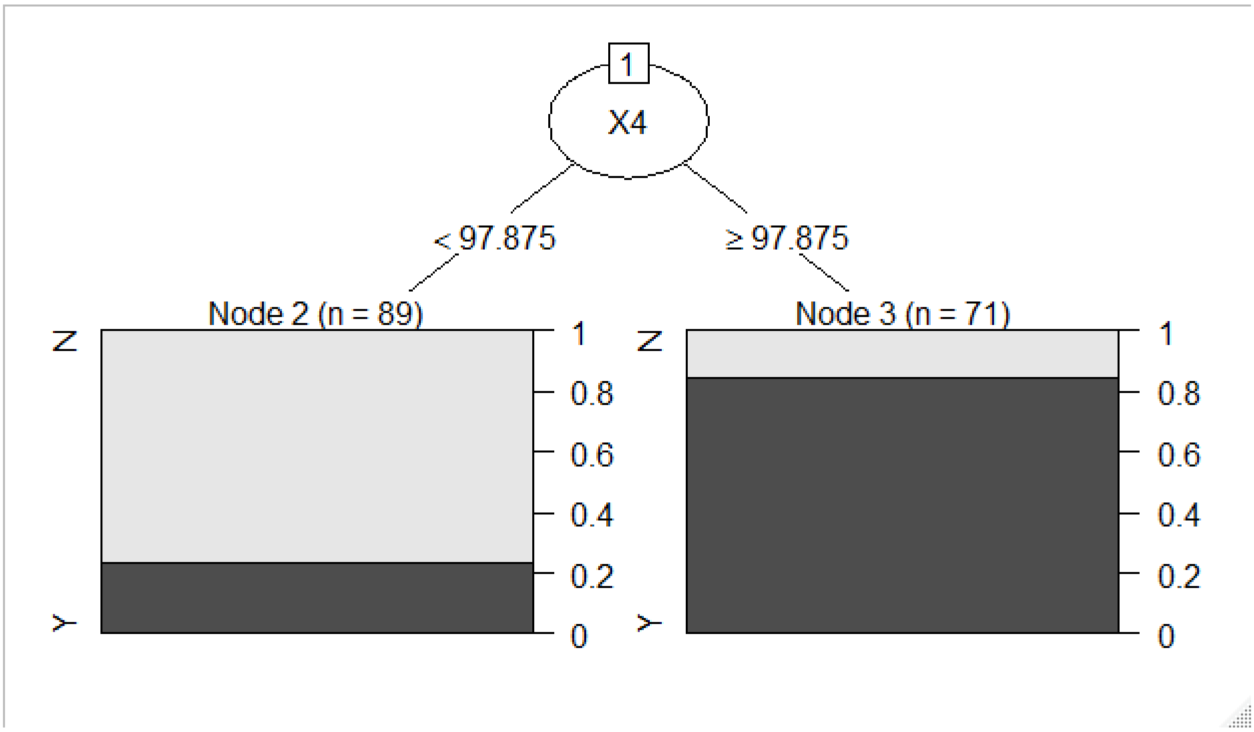
Y-1 vs Response results

Pruned model trained with 80% of data



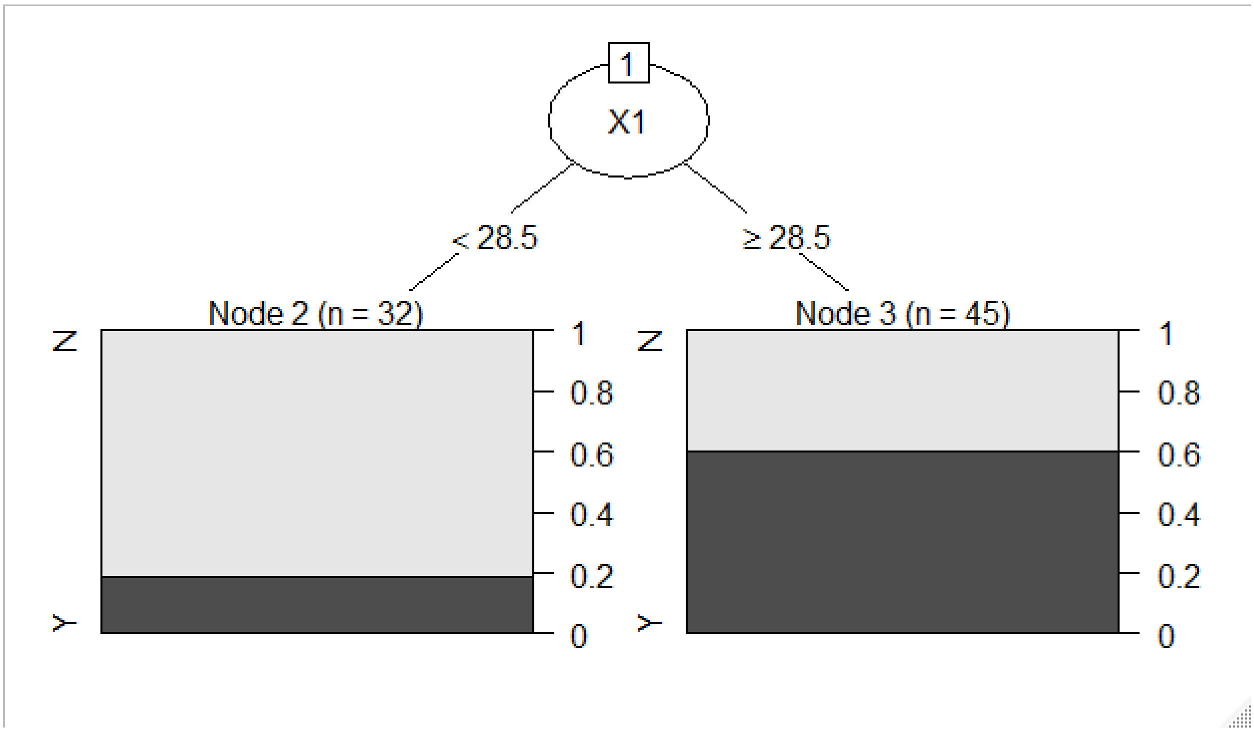
XY-1 vs Response results

Pruned model trained with 80% of data



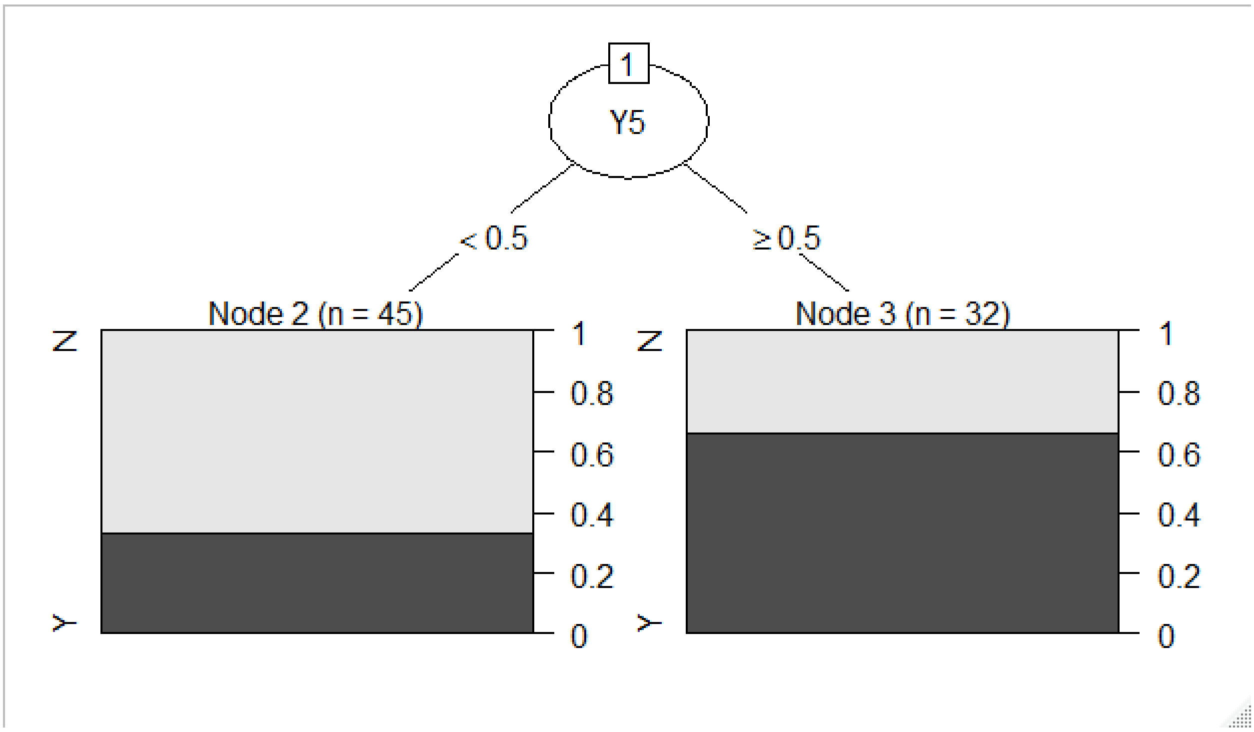
X-0 vs Response results

Pruned model trained with 80% of data



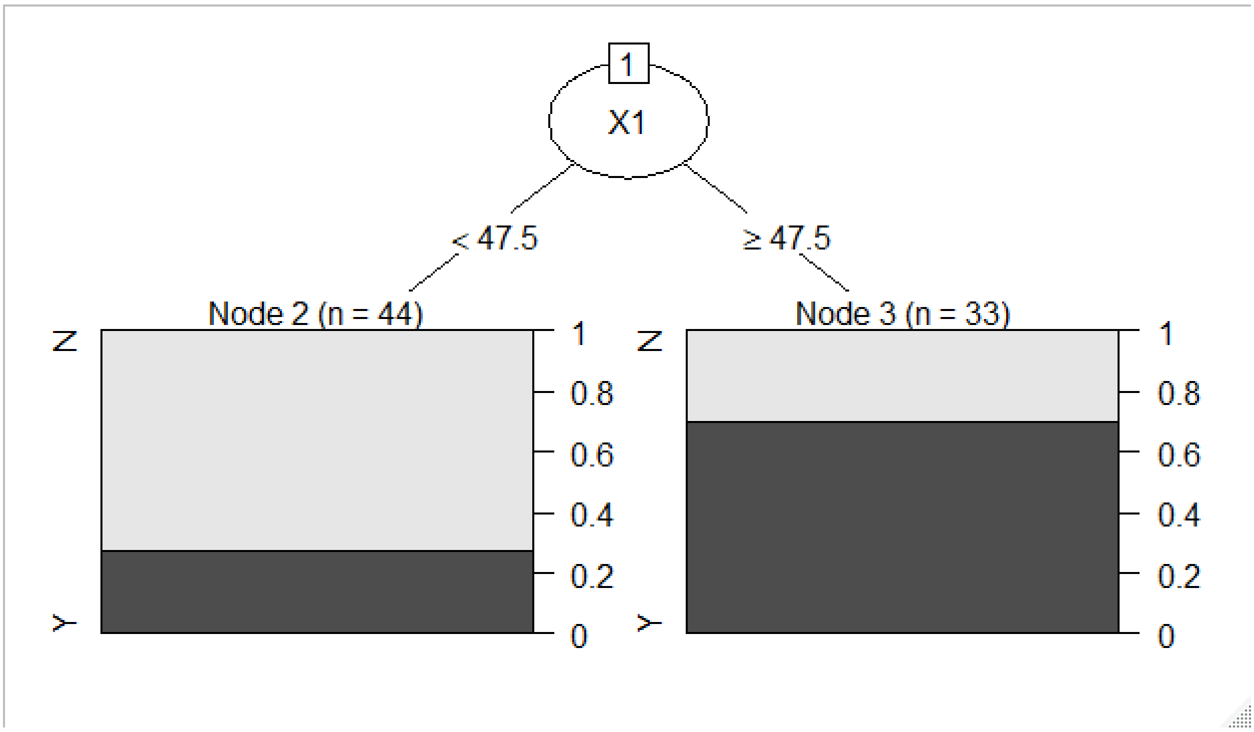
Y-0 vs Response results

Pruned model trained with 80% of data



XY-0 vs Response results

Pruned model trained with 80% of data



Conclusion:

The Response is in the form of 1 and 0 in the original dataset. This is converted to Y and N. Then dataset is **split into 80% and 20% for training and validation**. Pruned model is developed and prediction is run on test dataset and the results are compared with original expected responses in the test data. The comparison is represented in the form of confusion matrix .

The best model is then predicted on the basis of %Accuracy which is calculated by **{true positives+ true negatives }/{[true positives+true negatives+false positives+false negatives]}**

I have found the **model X-1 vs Response working best with average %Accuracy around 80%.** Also the

The **condition for best model (rel error + xstd < xerror) fits the best in this model** as the difference between the LHS and RHS of the above expression is the maximum in this model.

```
> print(DT_Model2$cptable)
         CP nsplit rel error    xerror       xstd
1 0.5408163      0 1.0000000 1.1428571 0.07163236
2 0.0100000      1 0.4591837 0.5408163 0.06368769
```