Gradient Boosting

Dataset:

Experience	Degree	Salary
2	BE	50k
3	Masters	70k
5	Masters	80k
7	PhD	100k

Independent features: Experience and Degree

Dependent feature: Salary

Step wise methods to do this process:

1) Make a base model: Which will give one output.
   Average (salary) = 75k
   Whenever we are about to deal with training dataset, our output is always being 75k.

Experience	Degree	Salary	Pred (y^)
2	BE	50k	75k
3	Masters	70k	75k
5	Masters	80k	75k
7	PhD	100k	75k

2) Computing Residual errors:
   Loss Function: Actual value – Predicted

Experience	Degree	Salary	Pred (y^)	Residuals
2	BE	50k	75k	-25k
3	Masters	70k	75k	-5k

5	Masters	80k	75k	5k
7	PhD	100k	75k	25k

3) Construction of Decision Tree:
Here the dependent feature will be Residual columns.

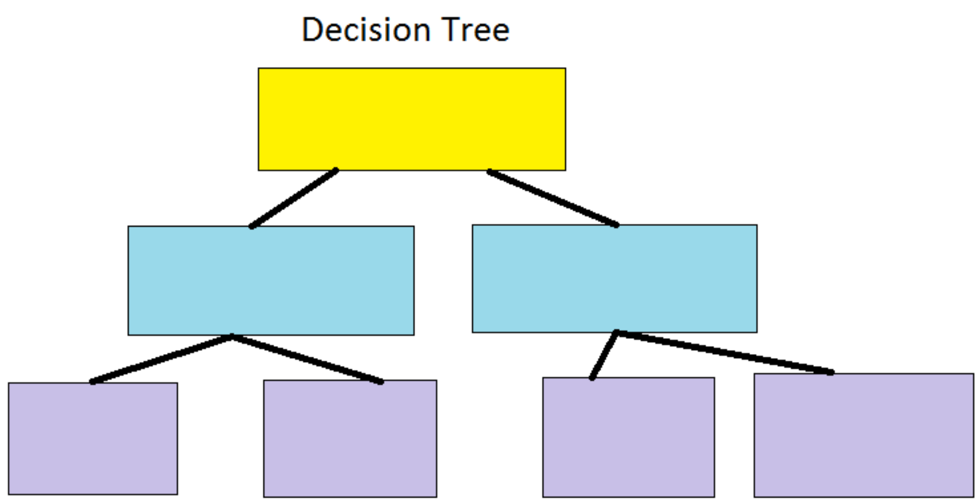

Figure 1 Decision Tree based of R1

Suppose after applying the values to the DT (Decision Tree), we got another Residual value column in our dataset:

Experience	Degree	Salary	Predicted (y^)	Residuals (R1)	Residuals (R2)
2	BE	50k	75k	-25k	-23k
3	Masters	70k	75k	-5k	-3k
5	Masters	80k	75k	5k	3k
7	PhD	100k	75k	25k	20k

Now what if we give our first entry to the DT, it'll predict 75k as predicted value. Our R2 is -23. Now addition of these values, we get **52k**.

Now take a look that our actual value (salary) is **50k**, but model shows **52k**. Is is better model showing good value? **No!** it is the issue of **Overfitting**.

What is the solution for such error? How to handle this issue?
We add a learning rate to their addition, and learning rate range is (0-1).
Now see the mathematics behind this:

= 75+ 0.1 x (-23)
= 72.7

We are still beyond our value 50K. Now, again what to do?

We now add a new Decision Tree again.

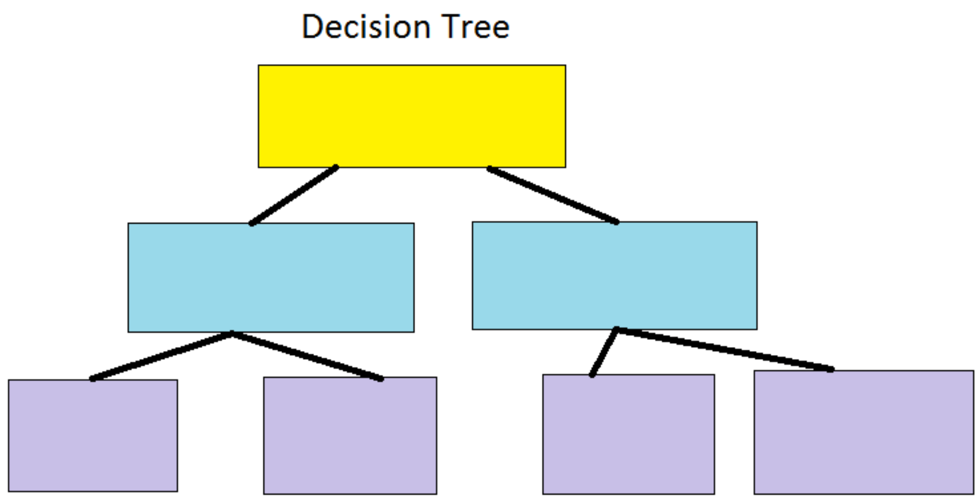

Figure 2 Decision Tree based on R2

$$\boldsymbol{f(x) = h_0(x) + L_1 h_2(x) + \cdots + L_n h_n(x)}$$

h_0 = Base model
h_2 = First DT, h_n = nth DT
L_1 or λ is Learning rate, which is being decided by hyper parameter tuning.

So our head formula for this is:

$$\boldsymbol{f(x) = \sum_{i=1}^{n} L_i h_i(x)}$$

This complete method is used just to reduce to Residuals. This is how we get multiple R2, R3, R4 etc decreased value to compute our main value (salary).

Mathematics behind Gradient Boosting:

$$F_0(x) = \arg min\left(\sum L(y, \gamma)\right)$$

$\text{Loss(L)} = \sum \frac{1}{2}(y - \tilde{y})^2$, γ = Predicted value.

Put all the values of y into the equation from table 1.

$L = \frac{1}{2}(50 - \tilde{y})^2 + \frac{1}{2}(70 - \tilde{y})^2 + \frac{1}{2}(80 - \tilde{y})^2 + \frac{1}{2}(100 - \tilde{y})^2$

Now compute the values with first order derivative:

$\frac{\partial L}{\partial y} = \frac{2}{2}(50 - \tilde{y})(-1) + \frac{2}{2}(70 - \tilde{y})(-1) + \frac{2}{2}(80 - \tilde{y})(-1) + \frac{2}{2}(100 - \tilde{y})(-1)$

$= -300 + 4\tilde{y}$

$4\tilde{y} = 300$

$\tilde{y} = 75$

This is how we got our Residual value. Hence this is the value of base model.

Experience	Degree	Salary	Pred (y^)
2	BE	50k	75k
3	Masters	70k	75k
5	Masters	80k	75k
7	PhD	100k	75k

See the table we got as we recommended…

Computing Residual

$$-\frac{\partial L}{\partial y} = \left[\frac{\partial L\left(y_i, f(x_i)\right)}{\partial f(x_i)}\right]$$

$$\frac{\partial L}{\partial y} = \text{rim}$$

$$\frac{\partial L}{\partial y} \text{ : Residuals}$$

Like above table we can our residuals as well by the difference of actual to the predicted, like above table.

Experience	Degree	Salary	Pred (y^)	Residuals
2	BE	50k	75k	-25k
3	Masters	70k	75k	-5k
5	Masters	80k	75k	5k
7	PhD	100k	75k	25k

Here R2 is now the dependent variable.

$$F_0(x) = \arg min\left(\sum L(y, \gamma 1)\right)$$

$$\gamma 1 = F_{m-1}(x_i) + \gamma$$

Here $\gamma 1$ reduces the values of Residuals. $F_{m-1}(x_i)$, Value of previous model.

L1 = $\sum \frac{1}{2}(y - 75 + \tilde{y})^2$

This is how previous model will take the value to compute further residual value.

Just like this we work to reduce residual values and finally we get a proper value (salary) predicted columns with low error. The simplest way to understand this is: If we get our base model value, we give it to a decision tree. Then again another value is given to another DT, meanwhile this iteration goes until we reach to minimum residual value. After then, what we get our value is , that is our main predicted value.

Predicted_value = (Predicted – Residuals)

This is how we go in iteration for each DT and Residual values.

END