DLI Accelerated Data Science Teaching Kit

Lecture 14.11 - Boosting

What is Boosting?

- Powerful machine learning algorithm
    - Achieves state-of-the-art accuracy on some tasks in regression, classification, ranking and more
- What does it do?
    - Combines a number of other weak models in order to generate a collectively strong model

Forms of Boosting

- Gradient Boosting
    - Supervised learning model using gradient descent to add weak models
- XG Boosting
    - Transforms the loss function into a more sophisticated objective function to inhibit overfitting

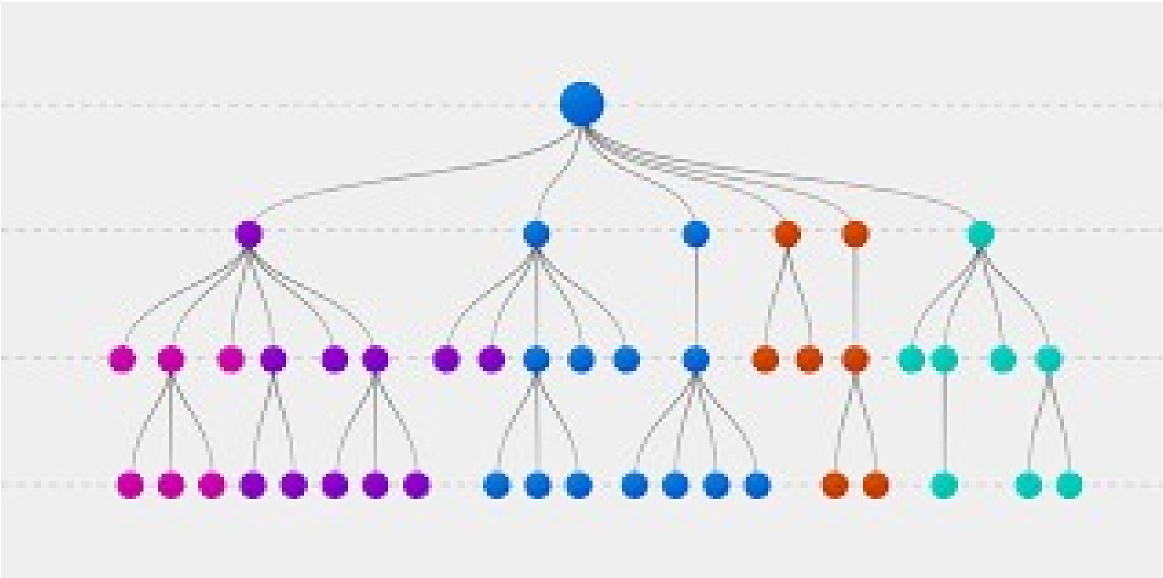


Image Reference: Explorium.ai

Example of Gradient Boosting

- Let us consider a scenario in which we are trying to predict the income of an individual
- To train a gradient boosting model, we need labeled training instances
    - Allows model to learn by example for later non-labeled examples

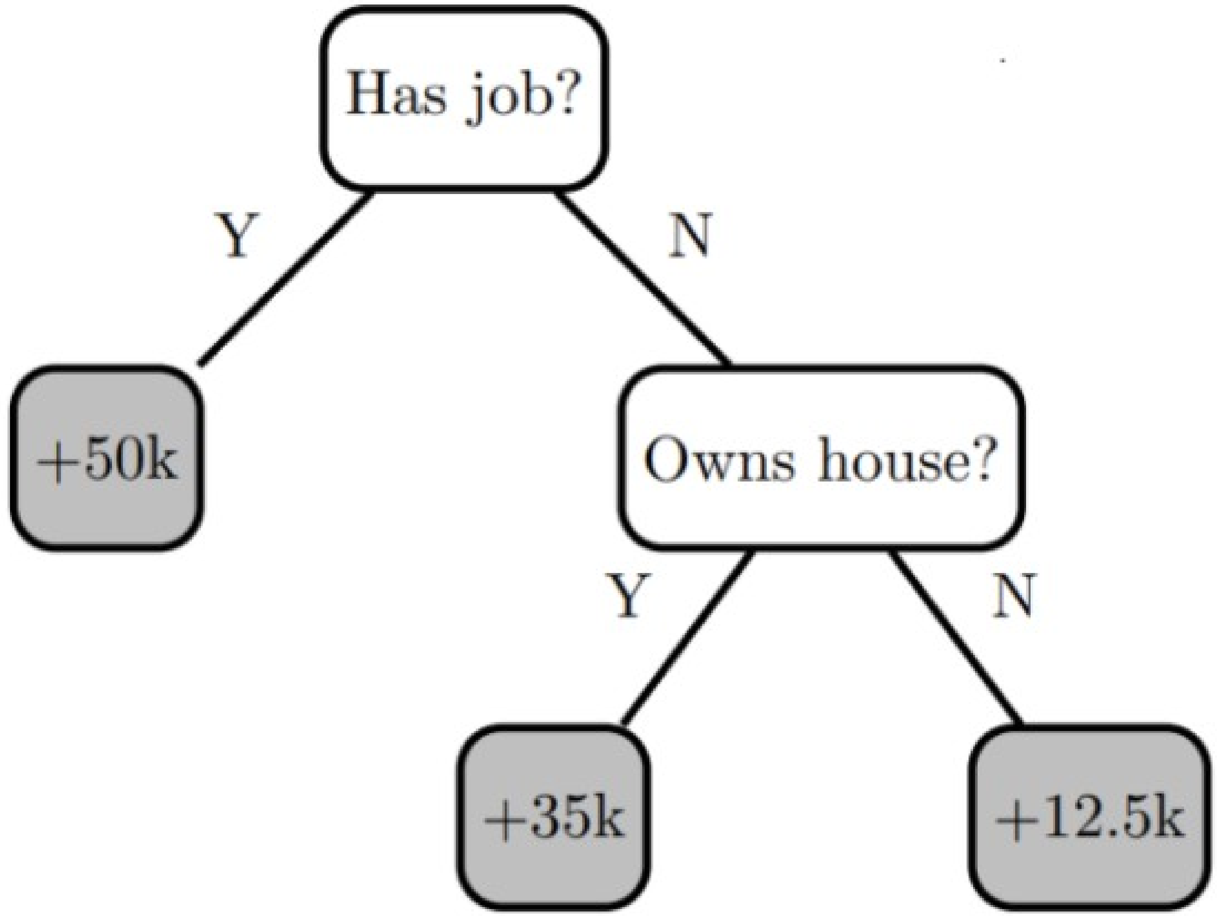
Reference: Nvidia RAPIDS
Image Credit: Entrepreneur.com

Dataset for Gradient Boosting

Instance	Age	Has Job	Owns House	Income
0	12	N	N	0
1	32	Y	Y	90
2	25	Y	Y	50
3	48	N	N	25
4	67	N	Y	35
5	18	Y	N	10

Our First Decision Tree

- To start our model, we develop a simple decision tree
  - Perform well on some examples
  - Not so well on others
- To improve the decision tree, we must add more nodes that account for more possibilities



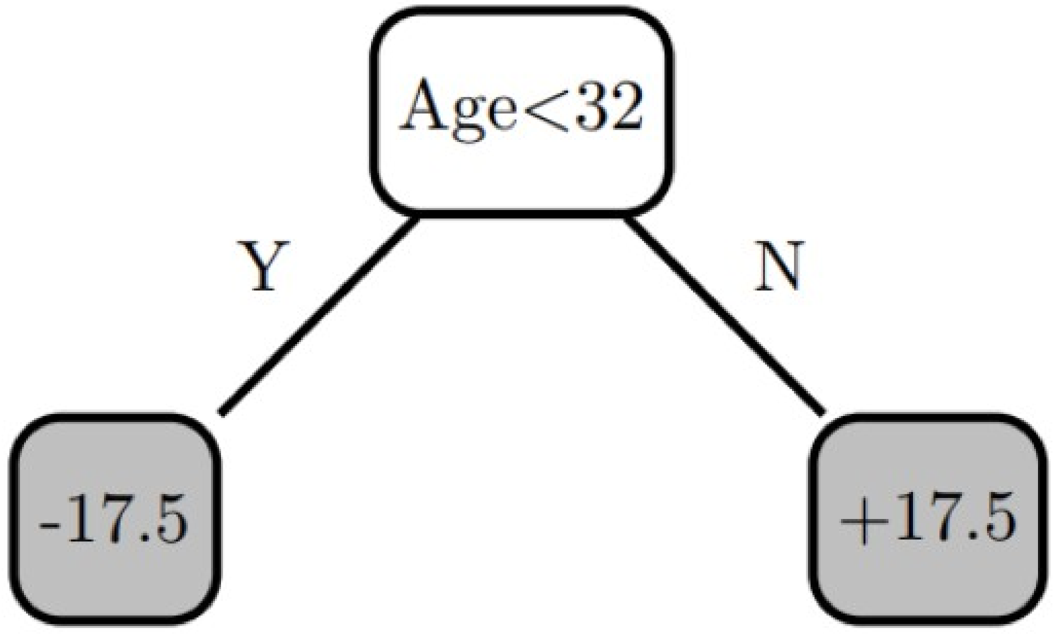
Reference: Nvidia RAPIDS

Residuals from First Decision Tree

Instance	Age	Has Job	Owns House	Income	Residuals
0	12	N	N	0	-12.5
1	32	Y	Y	90	40
2	25	Y	Y	50	0
3	48	N	N	25	12.5
4	67	N	Y	35	0
5	18	Y	N	10	-40

Our Second Decision Tree

- To improve model accuracy, we develop a second decision tree on top of the former one
  - Train on the residuals instead of the original labels
- We then recalculate the residuals after the new tree is added



Reference: Nvidia RAPIDS

Residuals from Second Decision Tree

Instance	Age	Has Job	Owns House	Income	Tree 0 Residuals	Tree 1 Residuals
0	12	N	N	0	-12.5	5
1	32	Y	Y	90	40	22.5
2	25	Y	Y	50	0	17.5
3	48	N	N	25	12.5	-5
4	67	N	Y	35	0	-17.5
5	18	Y	N	10	-40	-22.5

Comparison of Sum of Squared Errors

- When we compare the sum of squared errors, we see a drop after adding the second decision tree
  - As more trees are added, the SSE will drop steadily
  - Must be wary of overfitting
- This process creates gradient descent algorithm on the squared error loss function

Model	SSE
No Model (predict 0)	6275
Tree 0	1756
Tree 0 + Tree 1	837

$$SSE(y, \hat{y}) = \frac{1}{2}\Sigma_i(y_i - \hat{y}_i)^2$$

$$\frac{dSSE(y_i, \hat{y_i})}{d\hat{y}} = -(y_i - \hat{y_i})$$

Reference: Nvidia RAPIDS

DLI Accelerated Data Science Teaching Kit

Thank You