XGBoost

Working of XGBoost

Data set

Below is a sample dataset which we will be using for the illustration.

AGE	MASTER'S DEGREE?	SALARY
23	No	50
24	Yes	70
26	Yes	80
26	No	65
27	Yes	85

We have data for 5 persons about their ages, whether or not they have a master's degree, and their salary (in thousands). Our goal is to predict Salary using the XGBoost Algorithm.

Working of XGBoost

Step 1: Make an Initial Prediction and Calculate Residuals

This prediction can be anything. But let's assume our initial prediction is the average value of the variables we want to predict.

$$\frac{50+70+80+65+85}{5} = 70$$

Residuals are calculated using the following formula:

Residuals = Observed Values – Predicted Values

Here, our Observed Values are the values in the Salary column and all Predicted Values are equal to 70 because that is what we chose our initial prediction to be.

Working of XGBoost

Step 2: Build an XGBoost Tree

Each tree starts with a single leaf and all the residuals go into that leaf.

-20, 0, 10, -5, 15

Now we need to calculate something called a Similarity Score of this leaf.

$$\text{Similarity Score} = \frac{(Sum\ of\ Residuals)^2}{Number\ of\ Residuals + \lambda}$$

- λ (lambda) is a regularization parameter that reduces the prediction's sensitivity to individual observations
- **It prevents the overfitting of data (this is when a model fits exactly against the training dataset).**
- The default value of λ is 1 so we will let λ = 1 in this example.

Working of XGBoost

These are our original residuals which are assigned to the first leaf

-20, 0, 10, -5, 15

So the similarity score for this leaf will be

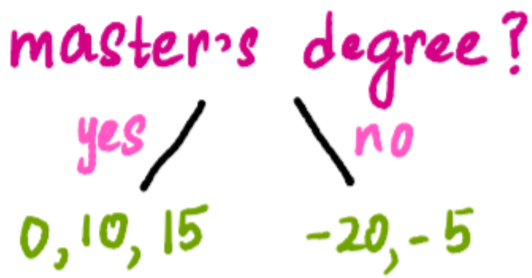
$$\frac{(-20 + 0 + 10 - 5 + 15)^2}{5 + 1} = 0$$

Working of XGBoost

Now we should see if we can do a better job clustering the residuals if we split them into two groups using thresholds based on our predictors — Age and Master's Degree?

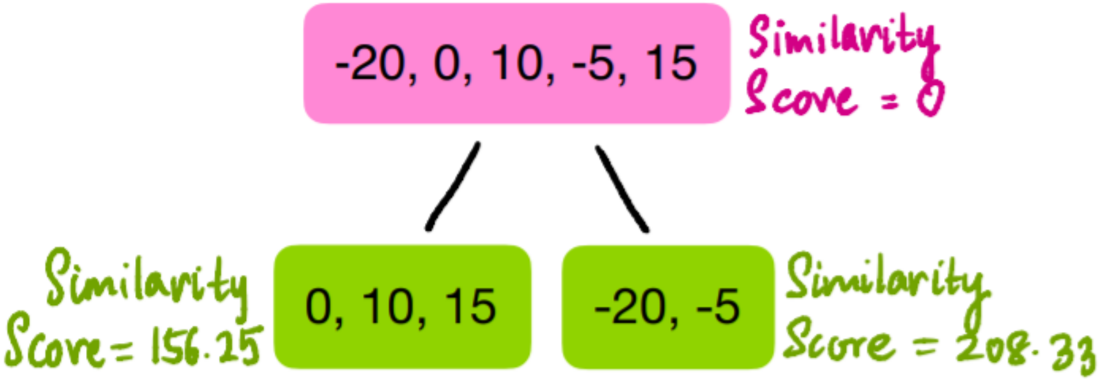
Splitting the Residuals basically means that we are adding branches to our tree.

First, let's try splitting the leaf using Master's Degree?



Now we will calculate the similarity Scores for the left and right leaves of the above splits

Working of XGBoost



- Now we need to **quantify** how much better the **leaves cluster similar Residuals than the root does**.
- We can do this by **calculating the Gain of splitting the Residuals** into two groups.
- **If the Gain is positive, then it's a good idea to split, otherwise, it is not.**

Working of XGBoost

$$\text{Gain} = \text{Left}_{\text{Similarity}} + \text{Right}_{\text{Similarity}} - \text{Root}_{\text{Similarity}}$$

$$= 156.25 + 208.33 - 0 = 364.5833$$

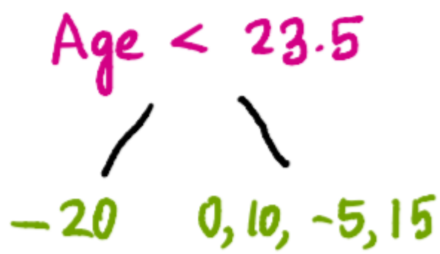
- Then we compare this Gain to those of the splits in Age.
- Since Age is a continuous variable, the process to find the different splits is a little more involved.
- First, we arrange the rows of our dataset according to the ascending order of Age.
- Then we calculate the average values of the adjacent values in Age.

AGE	
23	
	23.5
24	
	25
26	
	26
26	
	26.5
27	

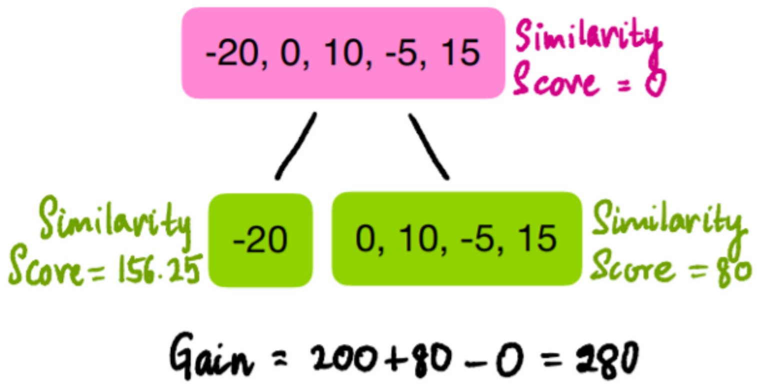
Working of XGBoost

- Now we split the Residuals using the four averages as thresholds and calculate Gain for each of the splits. The first split uses

  Age < 23.5:

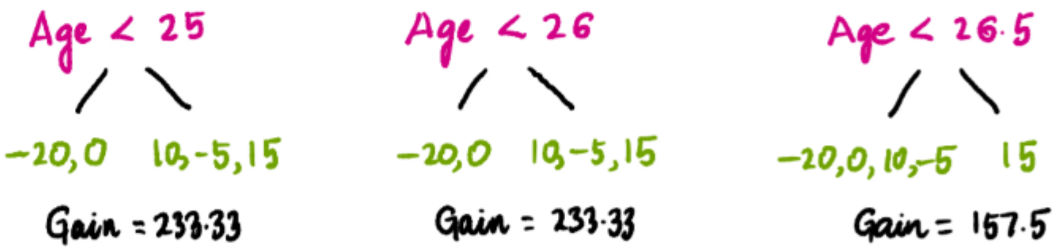


- For this split, we find the Similarity Score and Gain the same way we did for Master's Degree?



Working of XGBoost

- Do the same thing for the rest of the Age splits:



- Out of the one Master's Degree? split and four Age splits, the Master's Degree split has the greatest Gain value.
- so we'll use that as our initial split. Now we can add more branches to the tree by splitting our Master's Degree? leaves again using the same process described above.
- We use the initial Master's Degree? leaves as our root nodes and try splitting them by getting the greatest Gain value that is greater than 0.

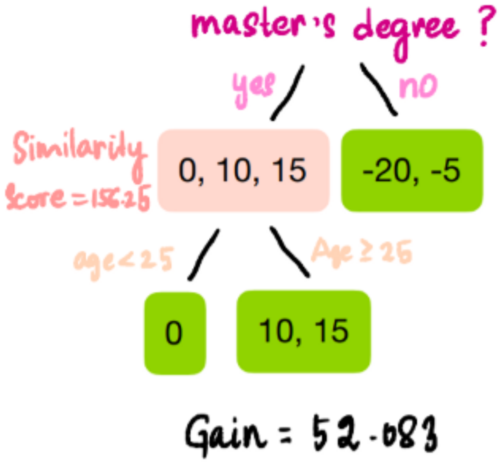
Working of XGBoost

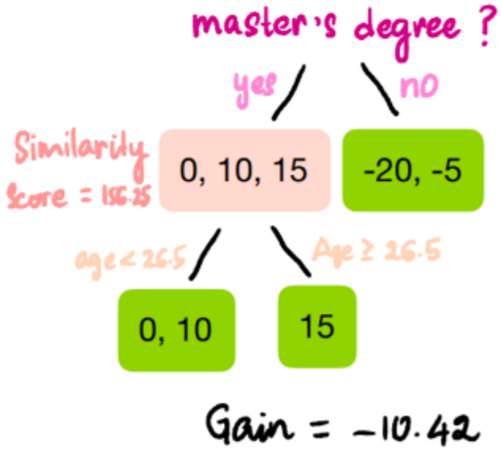
- Let’s start with the left node. For this node, we only consider the observations that have the value ‘Yes’ in Master’s Degree? because only those observations land in the left node.

AGE	MASTER'S DEGREE?	SALARY	Residuals
23	No	50	-20
24	Yes	70	0
26	Yes	80	10
26	No	65	-5
27	Yes	85	15

- So we calculate the Gain of the Age splits using the same process as before, but this time using the Residuals in the highlighted rows only.

Working of XGBoost





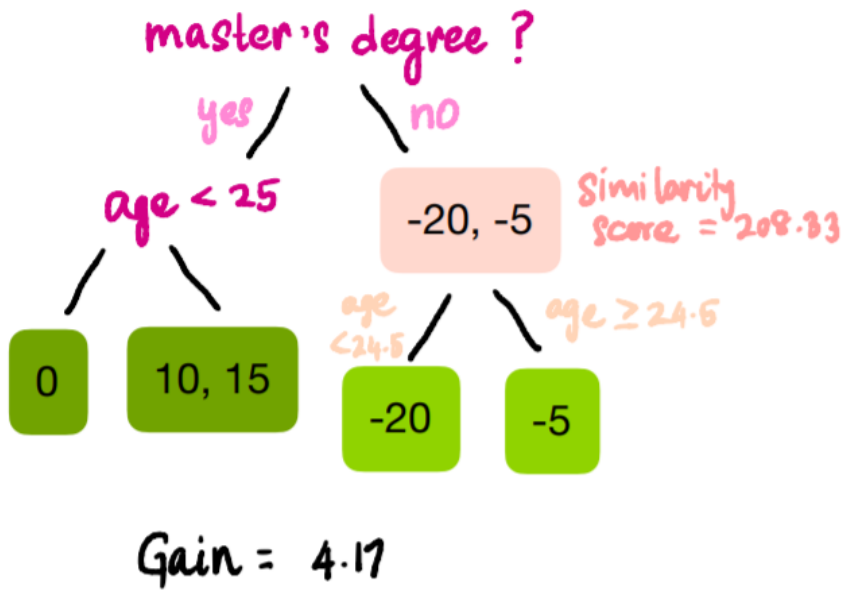
- Since only Age < 25 gives us a positive Gain, we split the left node using this threshold. Moving onto our right node, we only look at values with 'No' values in Master's Degree?

Working of XGBoost

AGE	MASTER'S DEGREE?	SALARY	Residuals
23	No	50	-20
24	Yes	70	0
26	Yes	80	10
26	No	65	-5
27	Yes	85	15

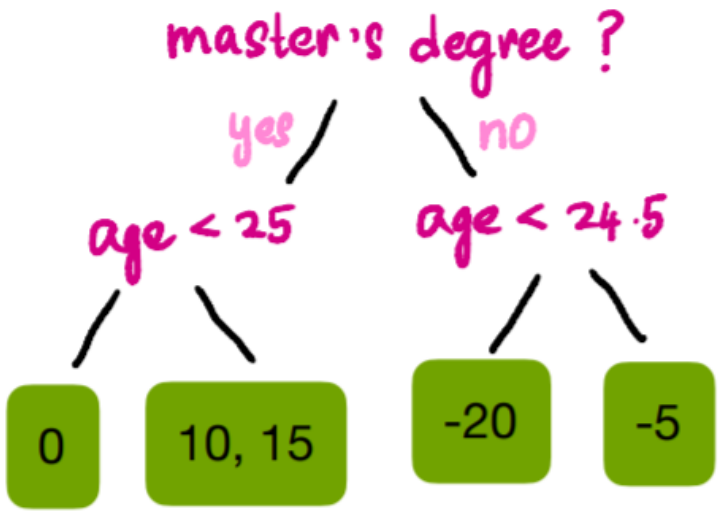
- We only have two observations in our right node, so the only split possible is Age < 24.5 because that is the average of the two Age values in the highlighted rows.

Working of XGBoost



- We can observe that the Gain of this split is positive.

Working of XGBoost



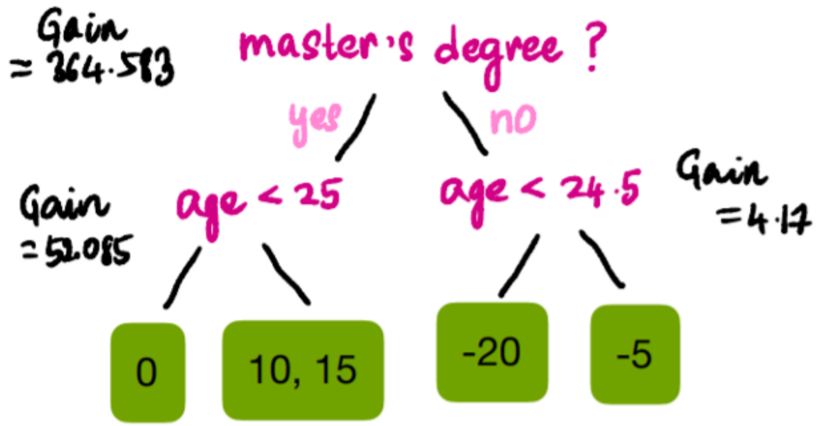
Working of XGBoost

Step 3: Prune the Tree

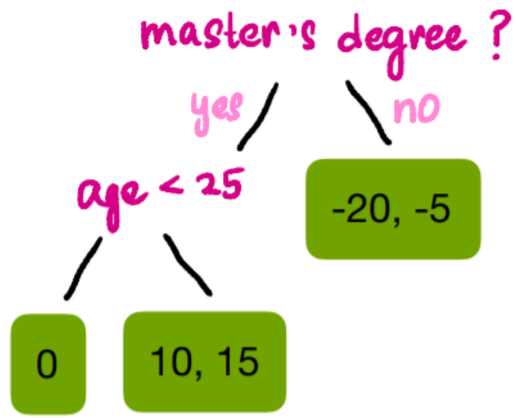
- Pruning is another way we can avoid overfitting the data.
- To do this we start from the bottom of our tree and work our way up to see if a split is valid or not.
- To establish validity, we use γ (gamma).
- If Gain — γ is positive then we keep the split, otherwise, we remove it.
- The default value of γ is 0.
- For illustrative purposes let’s set our γ to 50.

Working of XGBoost

Step 3: Prune the Tree



Since Gain — γ is positive for all splits except that of Age < 24.5, we can remove that branch.



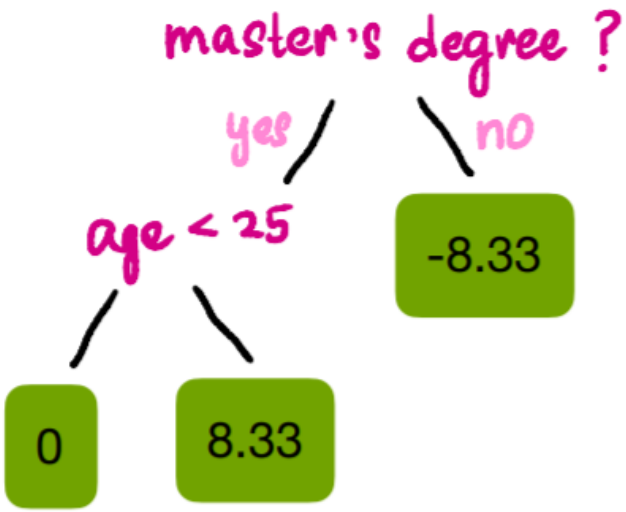
Working of XGBoost

Step 4: Calculate the Output Values of Leaves

Now we will calculate a single value in our leaf nodes because we can not have a leaf node giving us multiple outputs.

$$Output\ Value = \frac{Sum\ of\ Residuals}{Number\ of\ Residuals + \lambda}$$

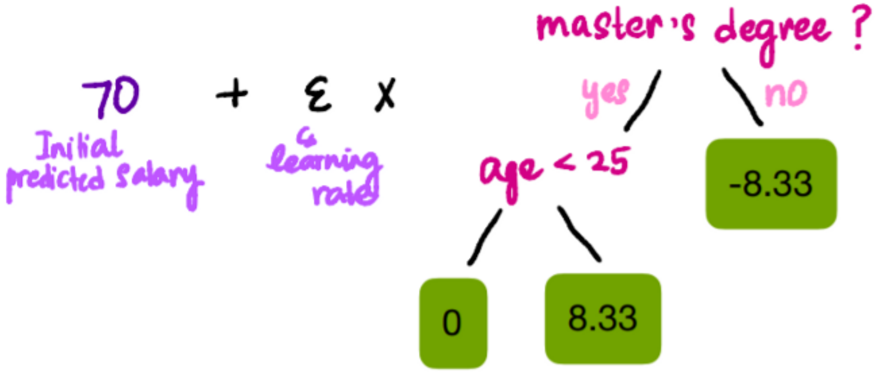
This is similar to the formula to calculate Similarity Score except we are not squaring the Residuals. Using the formula and λ = 1 our final tree is:



Working of XGBoost

Step 5: Make New Predictions

We can make predictions using this formula:



The XGBoost Learning Rate is ε (eta) and the default value is 0.3. So the predicted value of our first observation will be:

$$70 + 0.3 \times -8.33 = 67.501$$

Working of XGBoost

Step 5: Make New Predictions

Similarly, we can calculate the rest of the predicted values:

AGE	MASTER'S DEGREE?	SALARY	Predicted Values
23	No	50	67.501
24	Yes	70	70
26	Yes	80	72.499
26	No	65	67.501
27	Yes	85	72.499

Working of XGBoost

Step 6: Calculate Residuals Using the New Predictions

AGE	MASTER'S DEGREE?	SALARY	Residuals
23	No	50	-17.501
24	Yes	70	0
26	Yes	80	7.501
26	No	65	-2.501
27	Yes	85	12.501

- We see that the new Residuals are smaller than the ones before, this indicates that we've taken a small step in the right direction.
- As we repeat this process, our Residuals will get smaller and smaller indicating that our predicted values are getting closer to the observed values.

Working of XGBoost

Step 7: Repeat Steps 2–6

- Now we just repeat the same process over and over again, building a new tree, making predictions, and calculating Residuals at each iteration.
- We do this until the Residuals are super small or we reached the maximum number of iterations we set for our algorithm.
- If the tree we built at each iteration is indicated by T_i, where i is the current iteration, then the formula to calculate predictions is:

$$70 + \varepsilon \times T_1 + \varepsilon \times T_2 + \varepsilon \times T_3 + \dots + \varepsilon \times T_i$$

Working of XGBoost Classification

Working of XGBoost classification is almost same as XGBoost Regression with the following differences

1) In Classification we get predictions of log(odds) of probability

$$\log(\text{odds}) = \text{logit}(P) = \ln\left(\frac{P}{1-P}\right)$$

2) The formula for similarity score in classification is

$$\text{Similarity score} = \frac{(\Sigma \text{ Residuals})^2}{\Sigma\,[\text{Prob}*(1-\text{Prob})] + \lambda}$$

3) Prediction at a leaf node is

$$\text{Log-odds of a leaf} = \frac{\Sigma \text{Residuals}}{\Sigma[\text{Previous prob} * (1-\text{Previous prob})] + \lambda}$$

4) At the end we will convert these Log-odds of probability to probabilities using below formulae

New Log-odd =
Previous Log-odd +
Eta * Output value (Tree1) +
Eta * Output value (Tree2) +
Eta * Output value (Tree3) +
...

$$\text{Probability} = \frac{e^{\log(\text{odd})}}{1 + e^{\log(\text{odd})}}$$

Important Parameters in Sklearn

eta : This is the **learning rate** of the algorithm ,Default = 0.3

gamma : Default = 0 ,A node is split only when the resulting split gives a positive reduction in the loss function.

Gamma specifies the minimum loss reduction required to make a split.

max_depth : Default = 6 , It is the maximum depth of a tree

min_child_weight : Default = 0, It defines the minimum sum of weights of all observations required in a child.

It is used to control over-fitting. Higher values prevent a model from learning relations which might be highly specific to the particular sample selected for a tree.

Subsample : Default = 1, It denotes the fraction of observations to be randomly sampled for each tree.

colsample_bytree : Default = 1, the subsample ratio of columns when constructing each tree. Subsampling occurs once for every tree constructed.

lambda : Default = 1, This is used to handle the regularization part of XGBoost.

scale_pos_weight : Default = 0, It controls the balance of positive and negative weights. The ratio of number of negative class to the positive class.

n_estimators : Number of estimators (base learners)