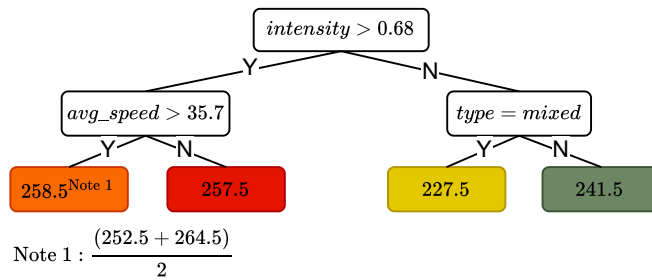


Step 1 : The implementation of the XGBoost algorithm sets  $\hat{y}_i^1 = 0.5$  for all  $i$ .

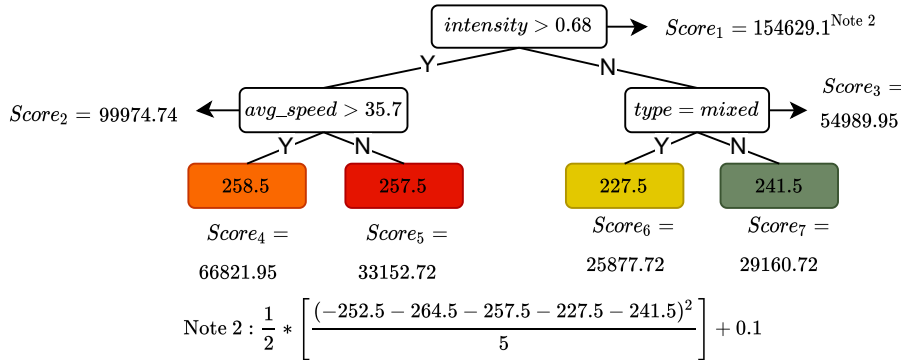
$y(\text{avg\_power\_comb})$	$\text{intensity}$	$\text{avg\_speed}$	$\text{type}$	$\hat{y}_i^1$	$y_i - \hat{y}_i^1$
253	0.7	42	mixed	0.5	252.5
258	0.83	35.7	sprinter	0.5	257.5
265	0.84	44.8	sprinter	0.5	264.5
228	0.62	26.1	climber	0.5	227.5
242	0.68	33	mixed	0.5	241.5

Step 2 : Generate a tree with the prediction error of the initial prediction or the weights of the last tree. Calculate the weights (Note 1) and the new prediction.

$\hat{y}_i^2$	$w_{1,i}^*$
-5.17	258.17
-0.17	258.17
6.83	258.17
0.5	227.5
0.5	241.5



Step 3 : Calculate the similarity scores  $Score_1$  for each node  $j$



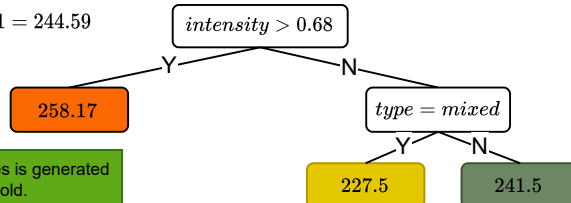
Step 4 : Calculate the Gain for each split

$$\text{Gain}_{\text{intensity} > 0.68} = 99974.74 + 54898.95 - 154629.1 = 244.59$$

$$\text{Gain}_{\text{avg\_speed} > 35.7} = -0.07$$

$$\text{Gain}_{\text{type} = \text{mixed}} = 48.49$$

Step 5 : Prune the tree for all splits with a negative gain and calculate the new weights and predictions



Step 6 : Repeat step 2-5 until a specified amount of trees is generated or the  $e$  weights fall below a specified threshold.

