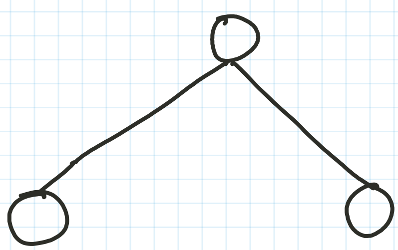
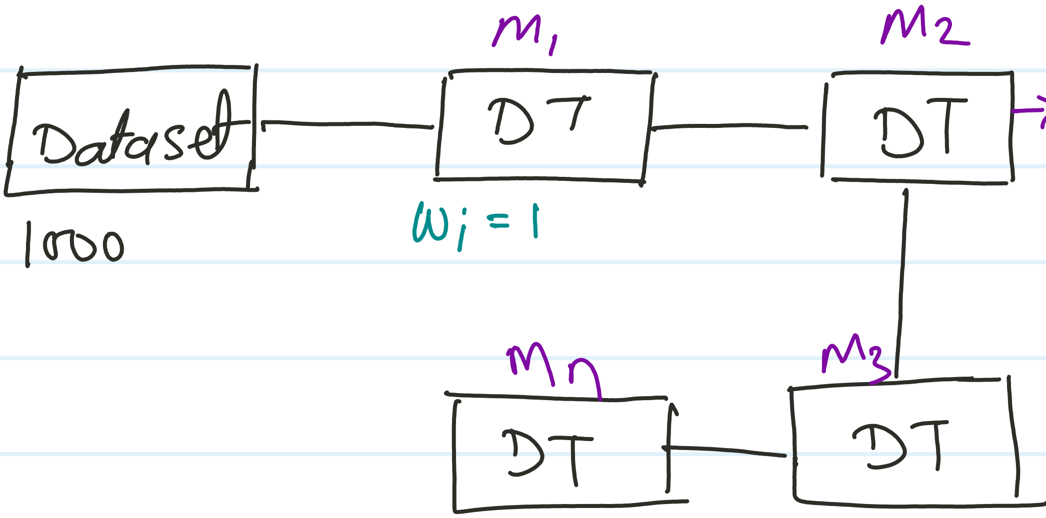
Ada Boost

Same as bagging, Boosting also has homogeneous and heterogeneous model but we use DT only in the ada boosting model.



single branch of tree called "Stump"

It is sequancial learning model in this model, every model is known as weaklearner, it pass o/p to the next weaklearner with some weight assigned to it.



Add more weight to misclassified point and reduce for correct class. point

Same process occurs for N numbers of weaklearner based of Combine prediction using a weight majority vote.

AdaBoost commonly used for "classification" problems and very less use for Regression problem.

$$\text{Model} = \alpha_1 (m_1) + \alpha_2 (m_2) + \alpha_3 (m_3) + \cdots + \alpha_n (m_n)$$

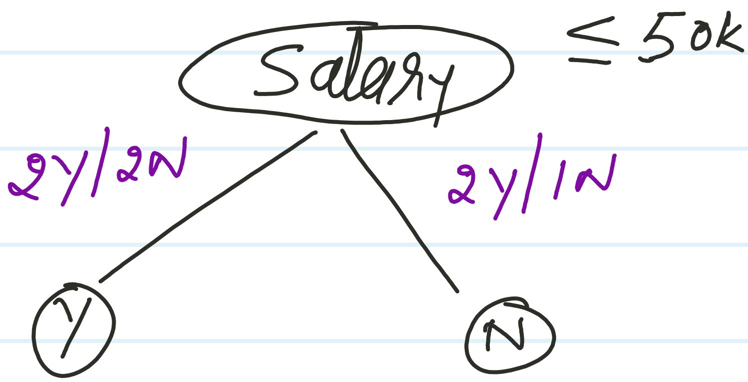
α = weight

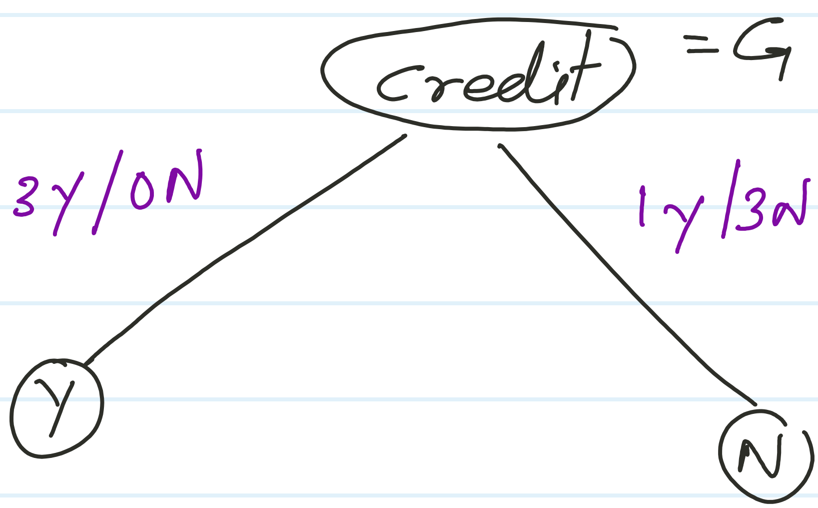
m_1, m_2, m_3, m_n = weaklearner.

Dataset

	Salary	credit	Approach		weight Ass
1	≤ 50K	Bad	No	$w_i = 1$	1/7
2	≤ 50K	Good	Yes		1/7
3	≤ 50K	Good	Yes		1/7
4	> 50K	Bad	N		1/7
5	> 50K	Good	Yes		1/7
6	> 50K	Neutral	Yes		1/7
7	<= 50K	Neutral	NO		1/7

step-1 we create decision tree stump





We use either Entrop or Gini and Find Information Gain

Step - 2 After first weaklearner training, how many wrong predicted point identify.

Add more weight to the wrong predicted point (misclassified point)

Reduce weight to the correct classified point.

Step - 3 one wrongly predicted point

performance of stump $= \frac{1}{2} \ln \left[\frac{1 - T.E}{T.E} \right]$

$$\alpha_1 = \frac{1}{2} \ln \left[\frac{1 - 1/7}{1/7} \right]$$

$$\simeq 0.896$$

$\alpha_1 = 0.896$

step-4 update weight

For correctly classified point

$= \text{weight} \times e^{-\text{performance of stump}}$

$= \frac{1}{7} \times e^{-(0.896)}$

$= 0.058$

For incorrect classified point

$= \text{weight} \times e^{\text{performance of stump}}$

$= 1/7 \times e^{(0.896)}$

$= 0.349$

For corret data point weight deerease
For incorreet datapoint weight Incrase.

Salary	Credit	Approch (y)	predicted point ($\hat{y}$)	updated weight
–	–	Y	Y	0.058
–	–	N	N	0.058
–	–	Y	-N	0.349
–	–	N	N	0.058
–	–	Y	Y	0.058
–	–	Y	-N	0.349
–	–	Y	Y	0.058

Step - 5

Normalization weight computation and assign Bins

Sal	credi.	App.	weight	update weight	Normalization
–	–	–	–	0.058 ÷ 0.988	0.0587
–	–	–	–	0.058 ÷ 0.988	0 0587
–	–	–	–	0.349 ÷ 0 988	0. 353
–	–	–	–	0.058 ÷ 0 988	0.0587
–	–	–	–	0.058 ÷ 0 988	0 . 0587
–	–	–	–	0.349 ÷ 0.988	0. 353
				0 .058 ÷ 0.988	0 . 0587
				0.988	1

Bin Assignment

0 – 0.0587	1
0.0587 – 0.1174	2
0.1174 – 0.4704	3 ✓
0.4704 – 0.5291	4
0.5291 – 0.5878	5
0.5878 – 0.9408	6 ✓
0.9408 – 1	7

Bin size will be big for incorrect data point, so from this method all the incorrect datapoint selected automatically.

And same process will occure for all the weak learnes and find weaklearnes delives us correct output.

pros -

1. Robust for overfitting condition

2. No feature scaling needed for base tree. (learner)

3. No feature selection needed for base tree (Learner)

Cons -

1. Affected by Noisy data

2. Complutation required more time than parallel training