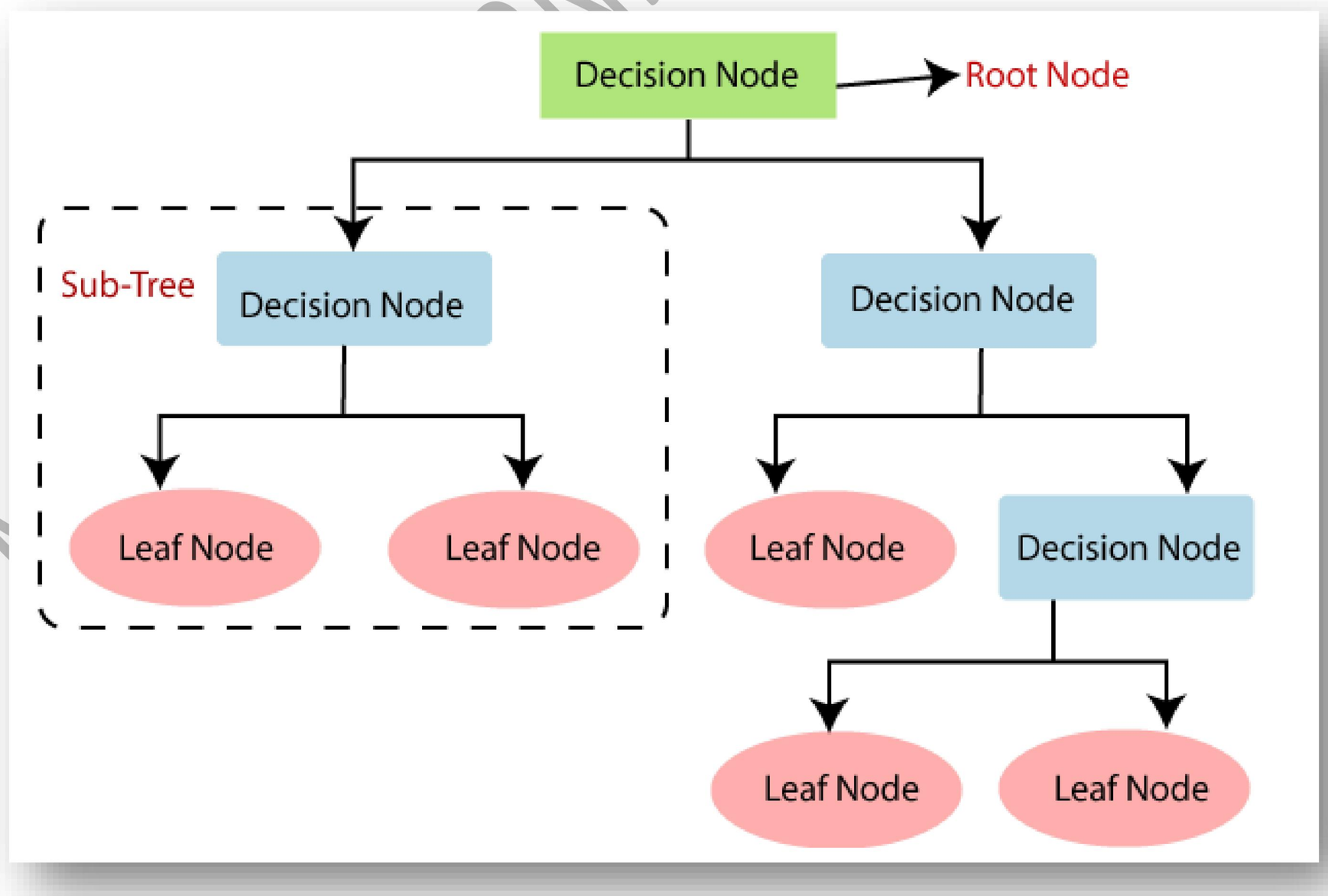
Chapter 9: Decision Tree:

- Decision Tree is a **Supervised learning technique** that can be used for solving Classification problems and Regression Problems.
- ***It is a graphical representation for getting all the possible solutions to a problem/decision based on given conditions.***
- It is called a decision tree because, similar to a tree, it starts with the root node, which expands on further branches and constructs a tree-like structure.

Types of Node:

- In a Decision tree, there are two nodes, which are the **Decision Node** and **Leaf Node.**
  - Decision nodes are used to make any decision and have multiple branches, whereas
  - Leaf nodes are the output of those decisions and do not contain any further branches.
- The decisions or the test are performed on the basis of features of the given dataset.



CART Algorithm for Classification:

- sklearn uses Classification & Regression Tree (CART) algorithm for making decision tree which uses Gini index to create decision points for the tree.
- Gini index is a measure of impurity.
- An attribute with the low Gini index should be preferred as compared to the high Gini index.
- The Gini Index is calculated for i = 1 to the n, number of attributes:

$$\textbf{Gini} = 1 - \sum(p_i)^2$$

Step 1: Find $Gini_k(D)$ for each attribute k

$$Gini_k(D) = \sum|D_i|/|D| * Gini(D_i)$$

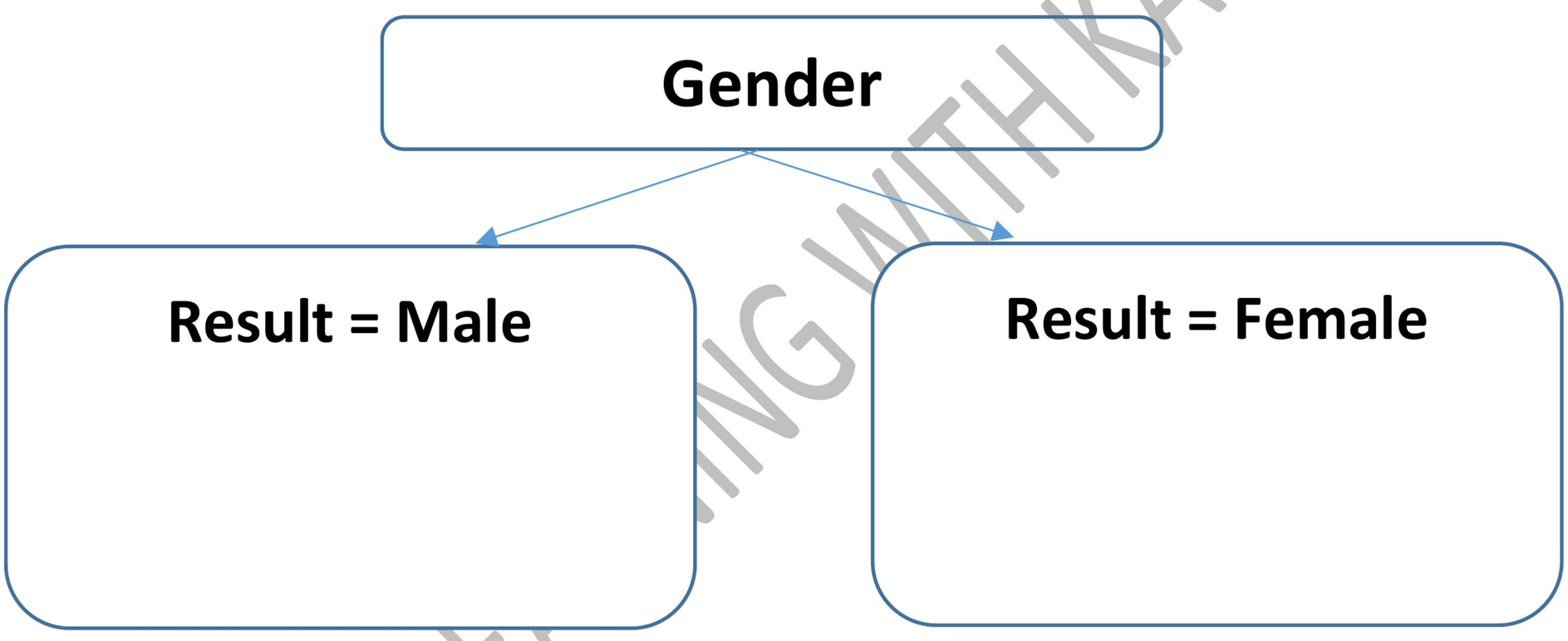
For i = 1 to n,

where D_i is a partition of the dataset D

Step 2: Partition at Minimum Gini Index Values

LOAN DEFAULTER CLASSIFIER

GENDER	OCCUPATION	DEFAULT
MALE	SALARY	yes
MALE	SALARY	yes
MALE	SALARY	no
MALE	SALARY	yes
MALE	SALARY	yes
MALE	BUSINESS	no
FEMALE	BUSINESS	no
FEMALE	BUSINESS	yes
FEMALE	BUSINESS	yes
FEMALE	BUSINESS	no

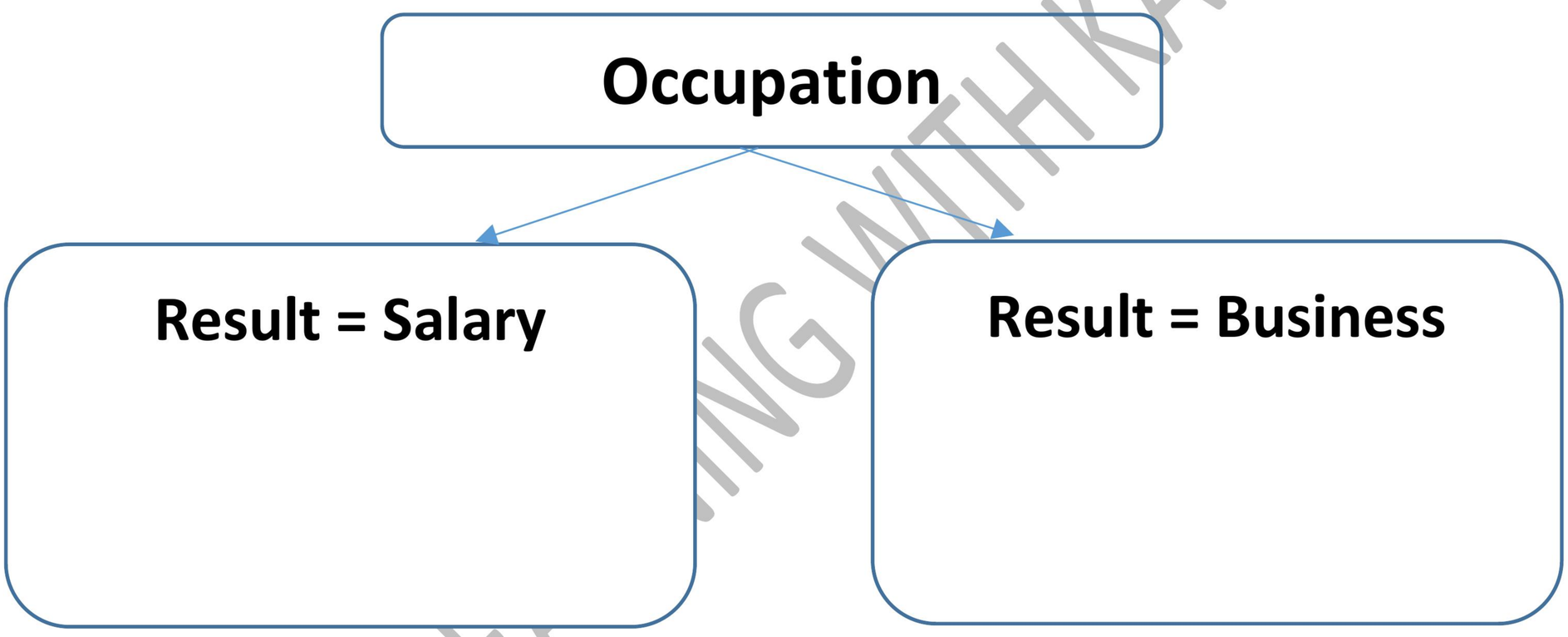


$1 – (\text{prob. of yes})^2 – (\text{prob. Of no})^2$
$1 – (4/(4+2))2 – (2/(4+2))2$
$1 – 0.44 – 0.11$
Gini Impurity = 0.45

$1 – (\text{prob. of yes})^2 – (\text{prob. Of no})^2$
$1 – (2/(2+2))^2 – (2/(2+2))^2$
$1 – 0.25 – 0.25$
Gini Impurity = 0.5

$6/(6+4) * 0.45 + 4/(6+4) * 0.5$
$0.27 + 0.20$
Weighted Gini Impurity= 0.47

GENDER	OCCUPATION	DEFAULT
MALE	SALARY	yes
MALE	SALARY	yes
MALE	SALARY	no
MALE	SALARY	yes
MALE	SALARY	yes
MALE	BUSINESS	no
FEMALE	BUSINESS	no
FEMALE	BUSINESS	yes
FEMALE	BUSINESS	yes
FEMALE	BUSINESS	no



$1 – (\text{prob. of yes})^2 – (\text{prob. Of no})^2$
$1 – (4/(4+1))^2 – (1/(4+1))^2$
$1 – 0.64 – 0.04$
Gini Impurity = 0.32

$1 – (\text{prob. of yes})^2 – (\text{prob. Of no})^2$
$1 – (2/(2+3))^2 – (3/(2+3))^2$
$1 – 0.16 – 0.36$
Gini Impurity = 0.48

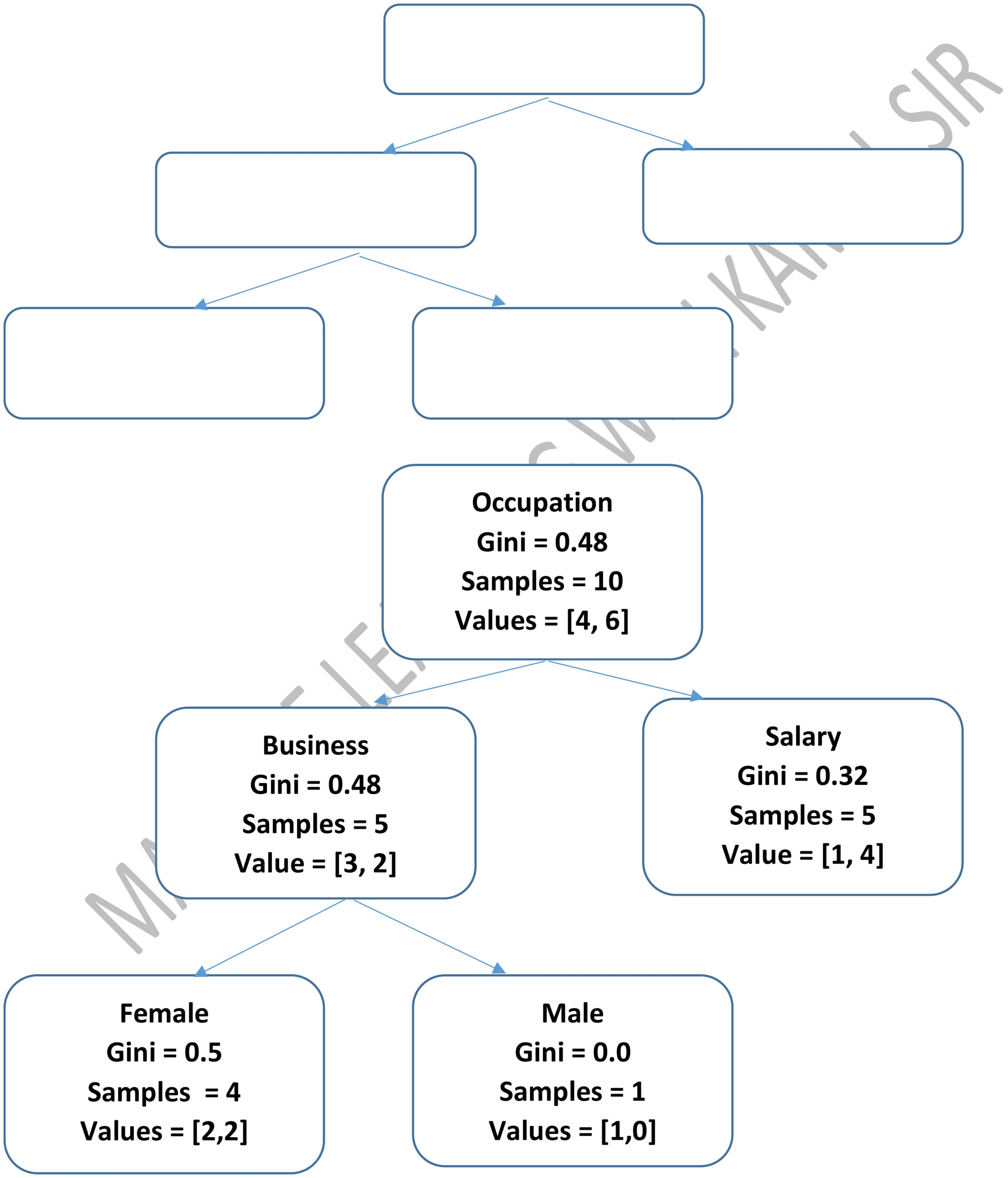
$5/(5+5) * 0.32 + 5/(5+5) * 0.48$
$0.16 + 0.24$
Weighted Gini Impurity= 0.40

Gender

Gini Impurity =

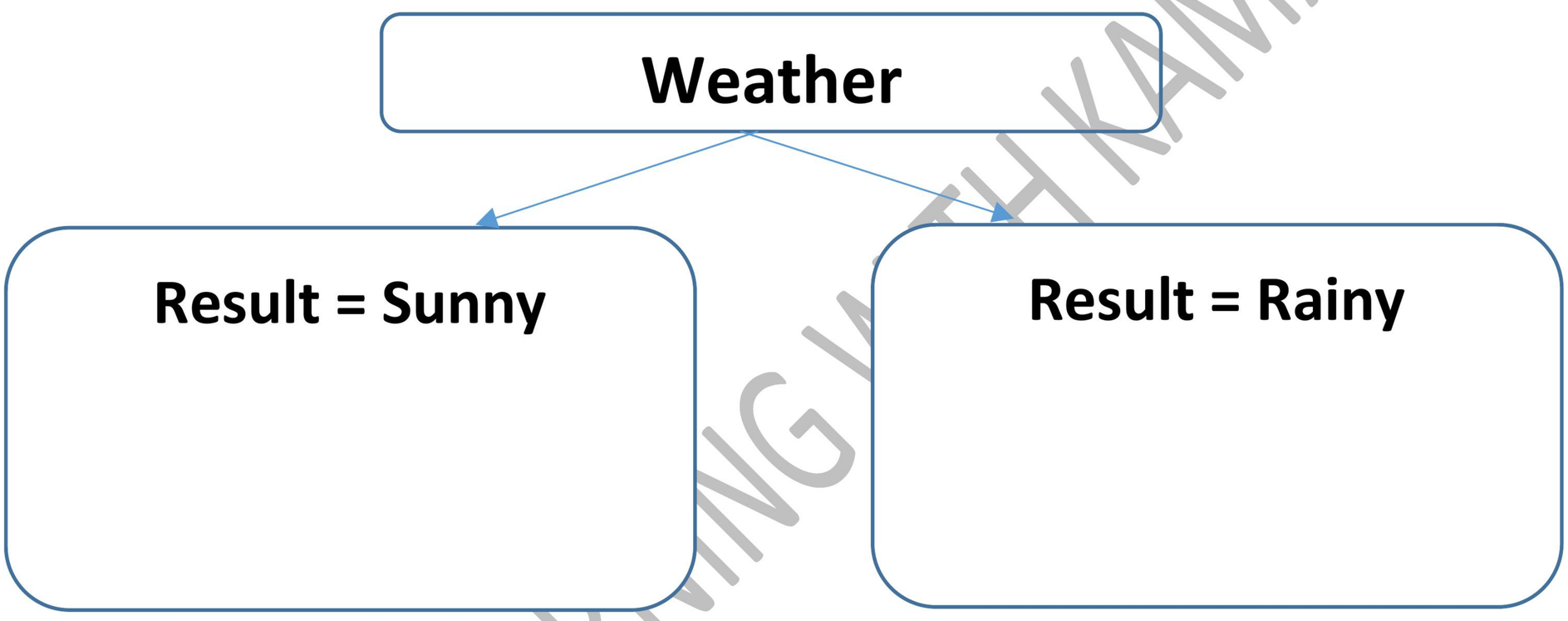
Occupation

Gini Impurity =



GO FOR RUNNING CLASSIFIER

Day	Weather	Just Ate	Will I Go Running?
1	Sunny	yes	yes
2	Rainy	yes	no
3	Sunny	no	yes
4	Rainy	no	no
5	Rainy	no	yes
6	Sunny	yes	yes
7	Rainy	no	no

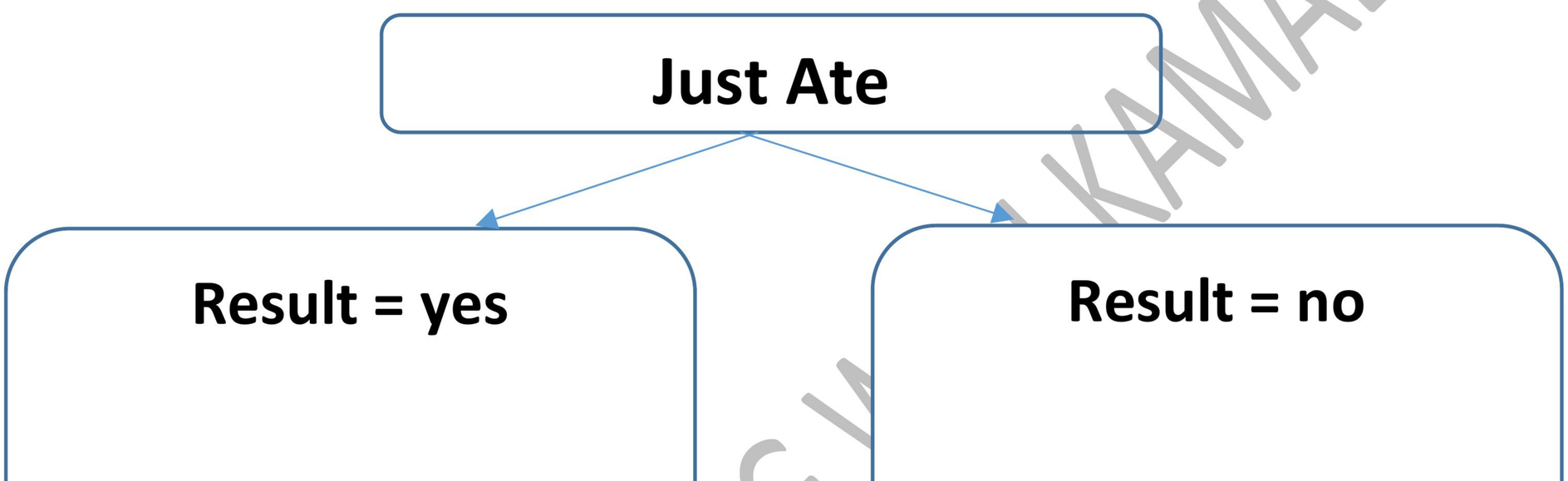


$1 - (\text{prob. of yes})^2 - (\text{prob. Of no})^2$
$1 - (3/(3+0))^2 - (0/(3+0))^2$
$1 - 1 - 0$
Gini Impurity = 0

$1 - (\text{prob. of yes})^2 - (\text{prob. Of no})^2$
$1 - (1/(1+3))^2 - (3/(1+3))^2$
$1 - 0.06 - 0.56$
Gini Impurity = 0.38

$3/(3+4) * 0 + 4/(3+4) * 0.38$
$0 + 0.22$
Weighted Gini Impurity= 0.22

Day	Weather	Just Ate	Will I Go Running?
1	Sunny	yes	yes
2	Rainy	yes	no
3	Sunny	no	yes
4	Rainy	no	no
5	Rainy	no	yes
6	Sunny	yes	yes
7	Rainy	no	no



$1 - (\text{prob. of yes})^2 - (\text{prob. Of no})^2$
$1 - (2/(2+1))^2 - (1/(2+1))^2$
$1 - 0.44 - 0.11$
Gini Impurity = 0.45

$1 - (\text{prob. of yes})^2 - (\text{prob. Of no})^2$
$1 - (2/(2+2))^2 - (2/(2+2))^2$
$1 - 0.25 - 0.25$
Gini Impurity = 0.5

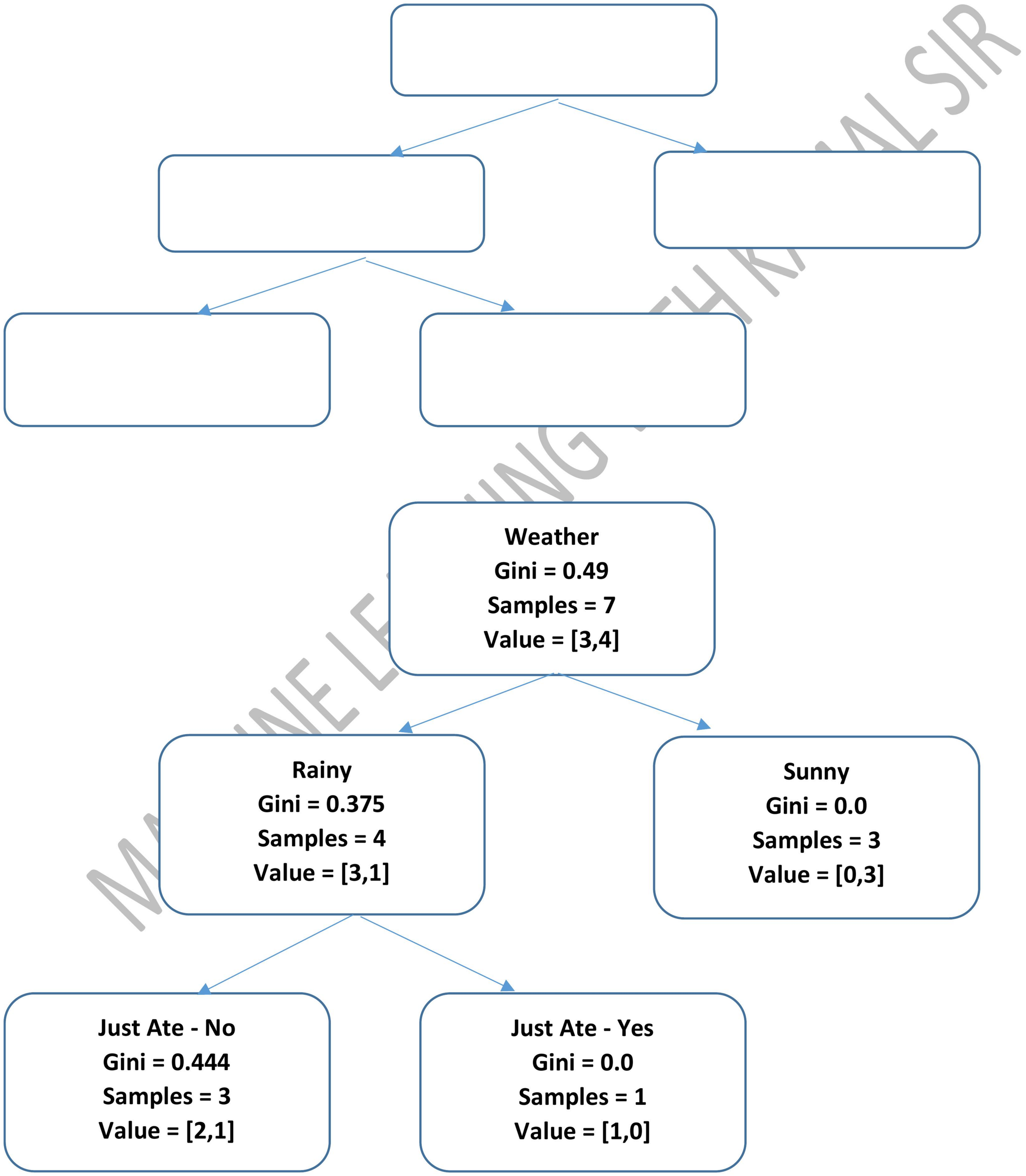
$3/(3+4) * 0.45 + 4/(3+4) * 0.5$
$0.19 + 0.29$
Weighted Gini Impurity= 0.48

Weather

Gini Impurity =

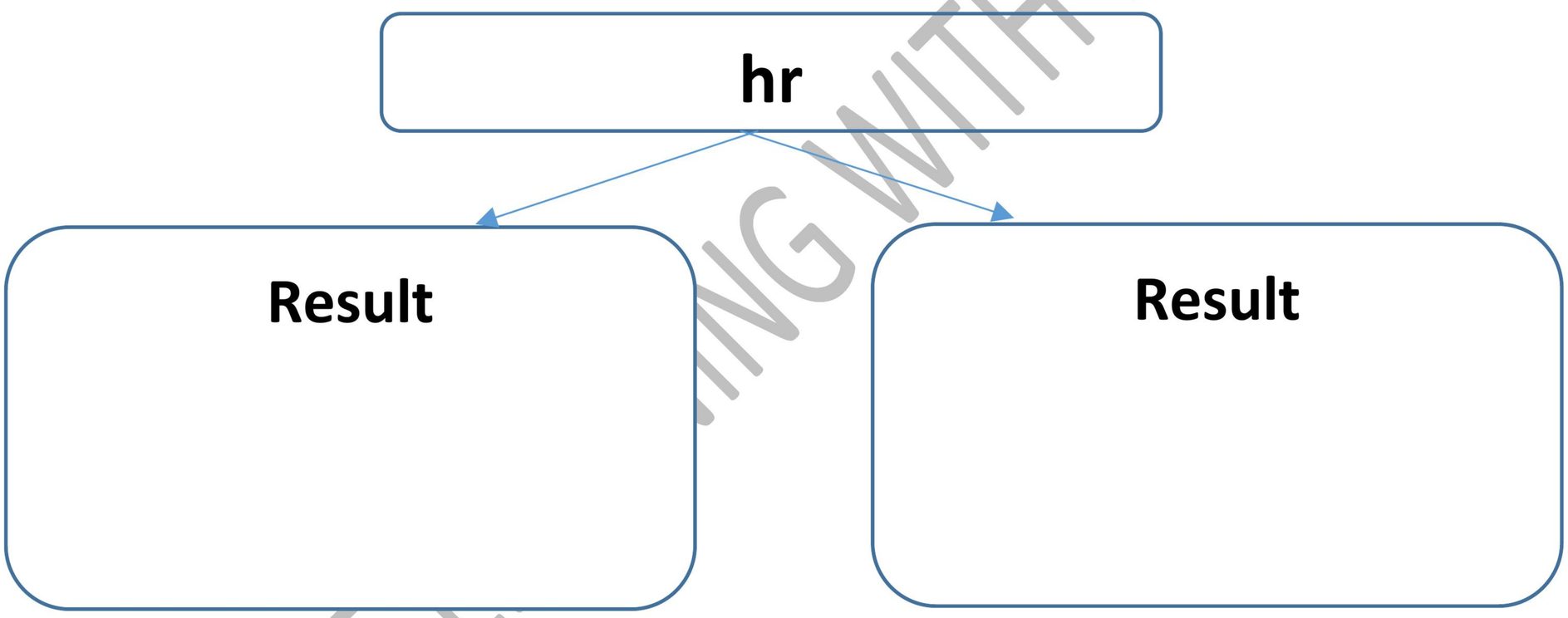
Just Ate

Gini Impurity =



RESULT CLASSIFIER

hr	result
4	Fail
5	Fail
7	Pass
10	Pass
13	Pass

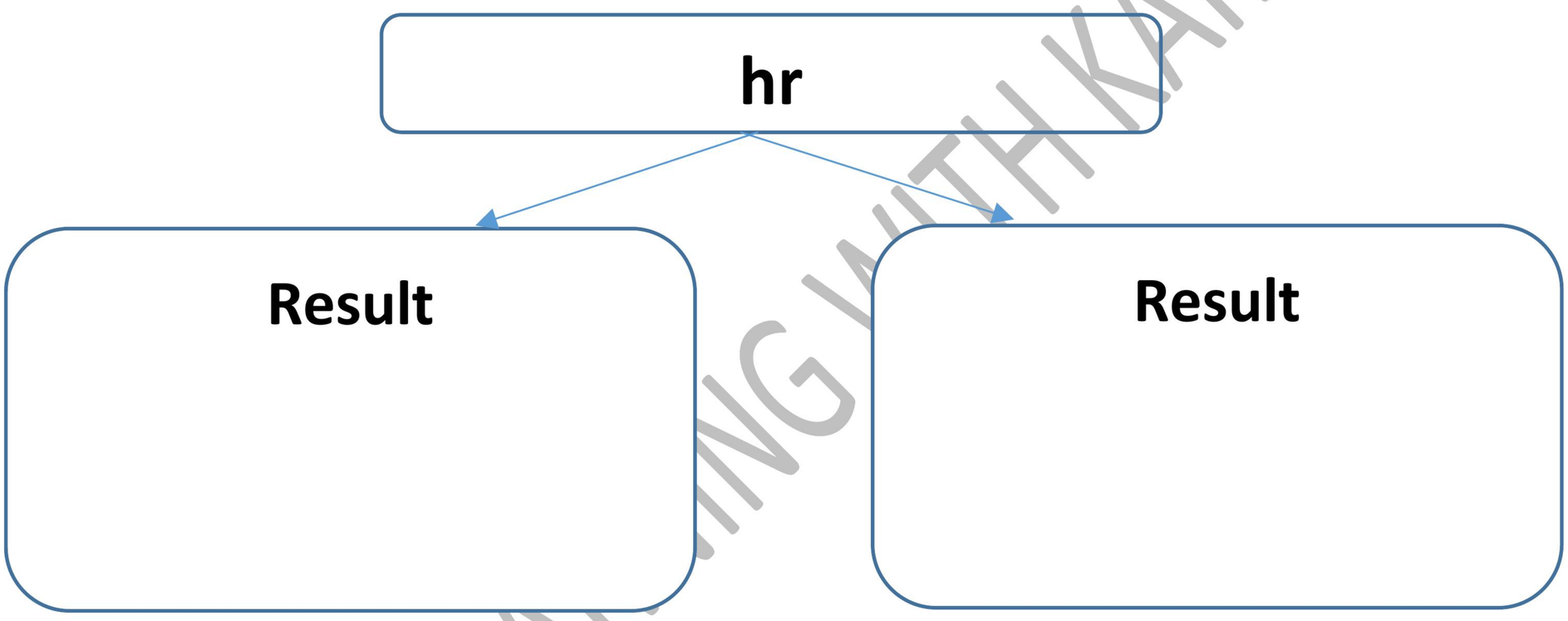


$1 – (\text{prob. of yes})^2 – (\text{prob. Of no})^2$
$1 – (0 /(0+1))^2 – (1 / (0 + 1))^2$
$1 – 0 – 1$
Gini Impurity = 0

$1 – (\text{prob. of yes})^2 – (\text{prob. Of no})^2$
$1 – (3 / (3+1))^2 – (1 / (3 + 1))^2$
$1 – 0.56 – 0.0625$
Gini Impurity = 0.38

$1 / (1+4) * 0 + 4 / (1 + 4) * 0.38$
$0 + 0.30$
Weighted Gini Impurity= 0.30

hr	result
4	Fail
5	Fail
7	Pass
10	Pass
13	Pass

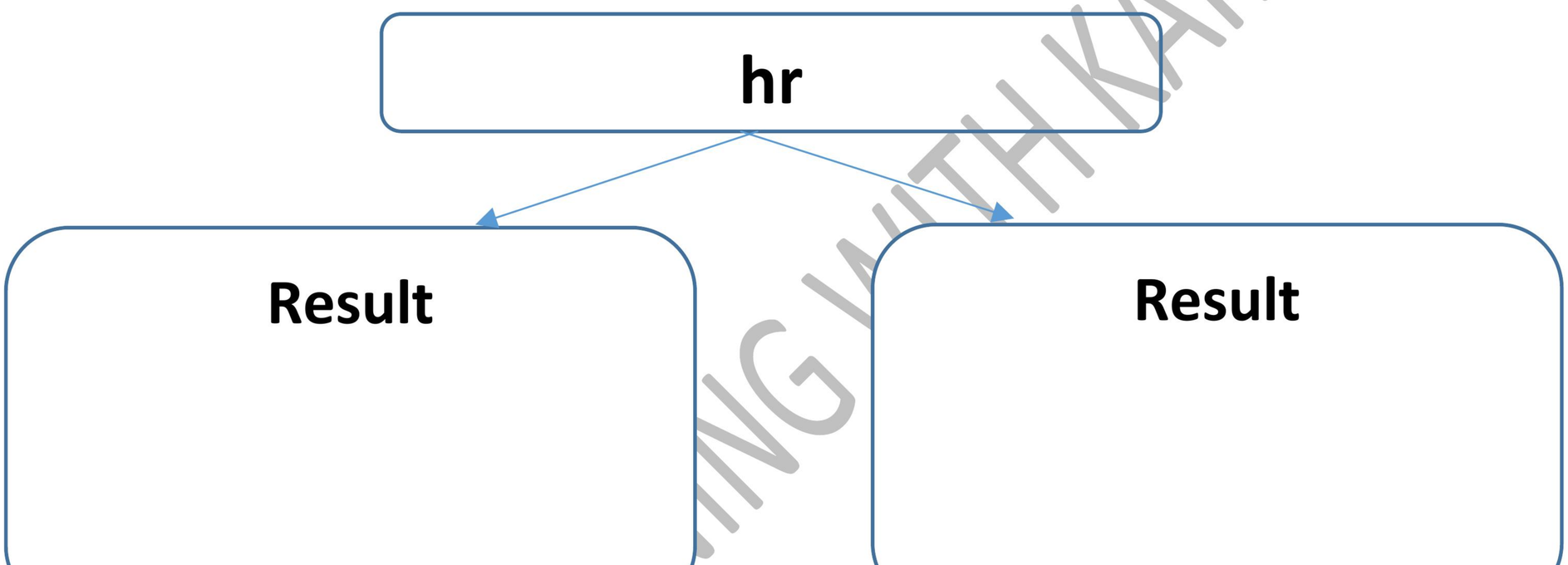


$1 – (\text{prob. of yes})^2 – (\text{prob. Of no})^2$
$1 – (0 / (0+2))^2 – (2/ (0 + 2))^2$
$1 – 0 – 1$
Gini Impurity = 0

$1 – (\text{prob. of yes})^2 – (\text{prob. Of no})^2$
$1 – (3 / (3+0))^2 – (0 / (3 + 0))^2$
$1 – 1 – 0$
Gini Impurity = 0

$2/ (2+3) * 0 + 3 / (2+3) * 0$
$0 + 0$
Weighted Gini Impurity= 0

hr	result
4	Fail
5	Fail
7	Pass
10	Pass
13	Pass

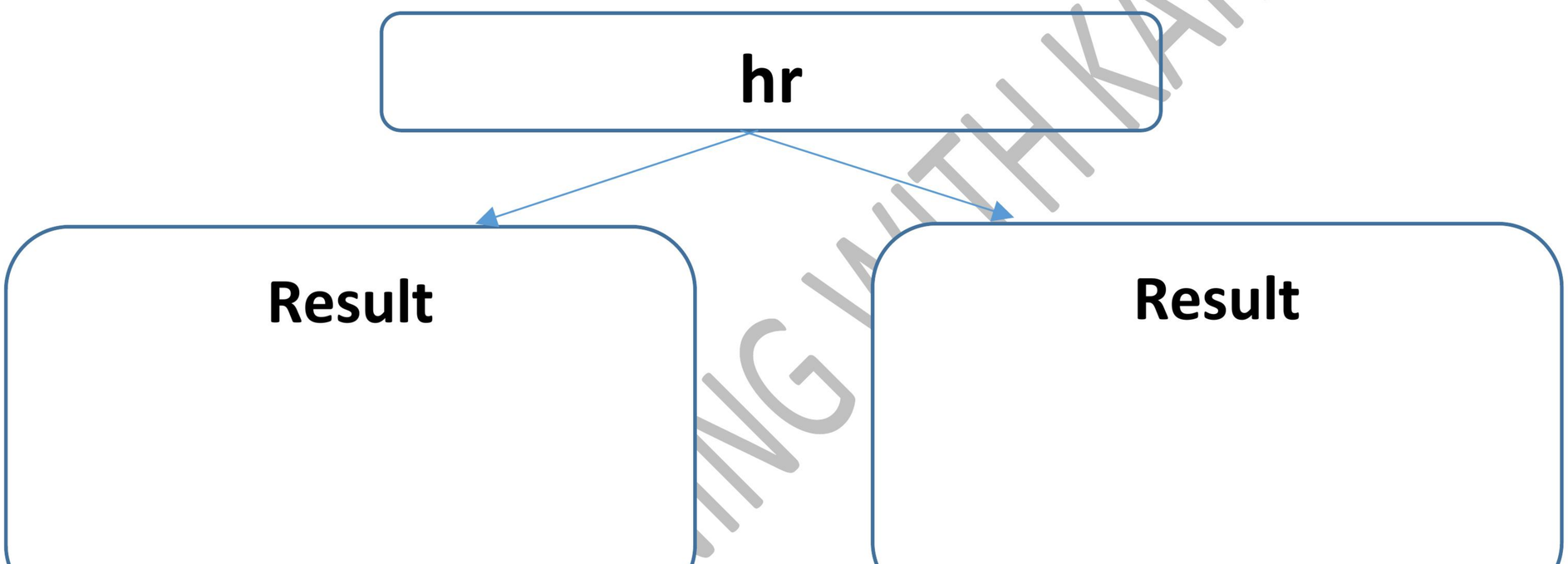


$1 – (\text{prob. of yes})^2 – (\text{prob. Of no})^2$
$1 – (1 / (1+2))^2 – (2/ (1 + 2))^2$
$1 – 0.11 – 0.44$
Gini Impurity = 0.44

$1 – (\text{prob. of yes})^2 – (\text{prob. Of no})^2$
$1 – (2 / (2+0))^2 – (0 / (2 + 0))^2$
$1 – 1 – 0$
Gini Impurity = 0

$3/ (3+2) * 0.44 + 2 /(3+2) * 0$
$0.26 + 0$
Weighted Gini Impurity= 0.26

hr	result
4	Fail
5	Fail
7	Pass
10	Pass
13	Pass



$1 – (\text{prob. of yes})^2 – (\text{prob. Of no})^2$
$1 – (2 /(2+2))^2 – (2/(2 + 2))^2$
$1 – 0.25 – 0.25$
Gini Impurity = 0.5

$1 – (\text{prob. of yes})^2 – (\text{prob. Of no})^2$
$1 – (1 / (1+0))^2 – (0 / (1 + 0))^2$
$1 – 1 – 0$
Gini Impurity = 0

$4/ (4+1) * 0.5 + 1 /(4+1) * 0$
$0.4 + 0$
Weighted Gini Impurity= 0.4

4.5

Gini Impurity =

6

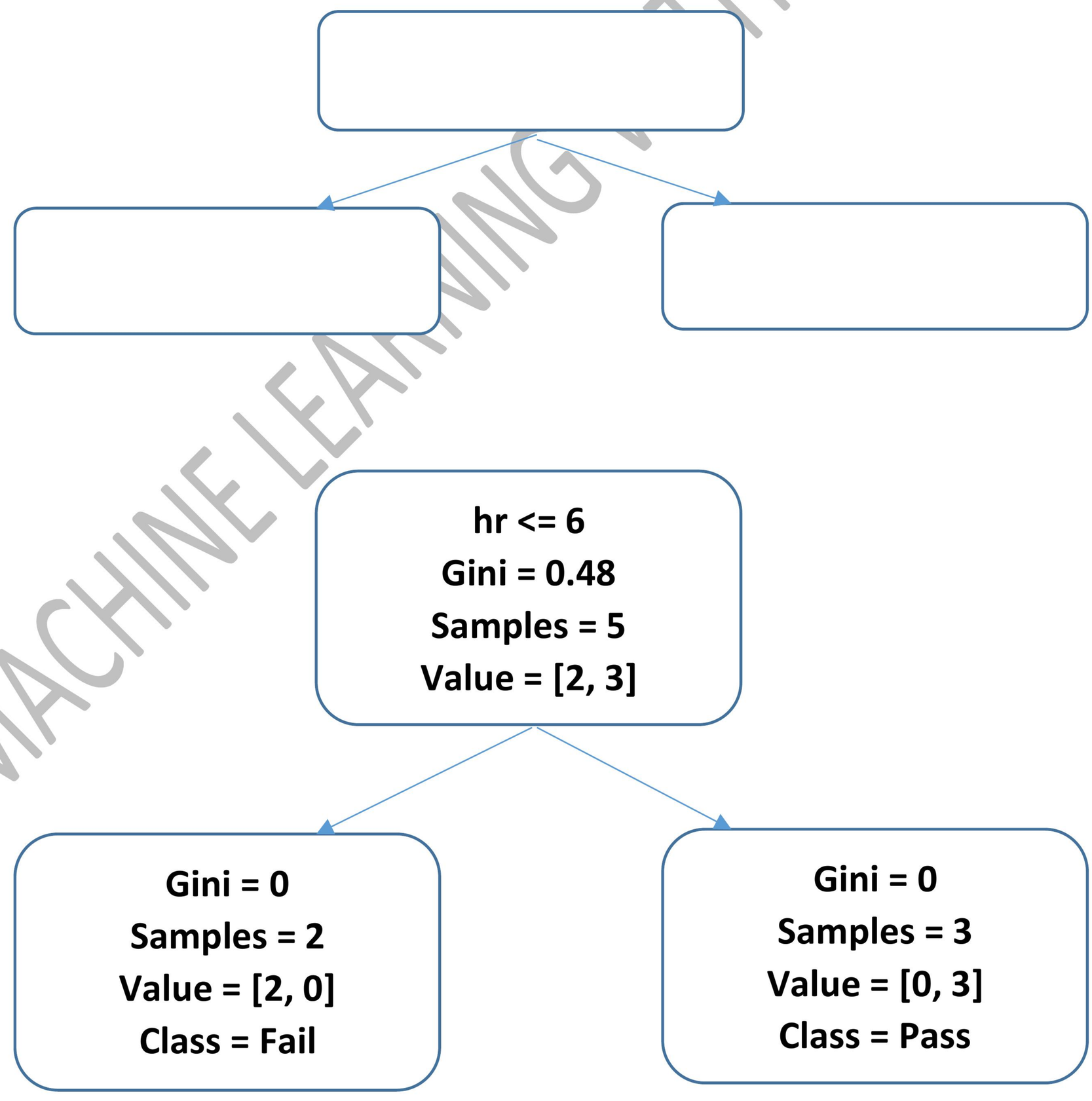
Gini Impurity =

8.5

Gini Impurity =

11.5

Gini Impurity =



Decision Tree Regressor:

- Decision trees algorithm can be more suitable for regression problems than other common and popular algorithms.
- Below are the cases where you would likely prefer a decision tree over other regression algorithms:
  - There are non-linear or complex relationships between features and labels
  - You need a model that is easy to explain
  - Regression trees are used when the dependent variable is continuous.

CART Algorithm for Regression:

- sklearn uses Classification & Regression Tree (CART) algorithm for making decision tree which uses Mean Square Errors (MSE) to create decision points for the tree.
- MSE is calculated as:

$$MSE = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

Step 1: Find MSE by dividing data into two region

Step 2: Partition at Minimum MSE

SALARY PREDICTOR

Position	Level	Salary
Business Analyst	1	3.5
Junior Consultant	2	5
Senior Consultant	3	6
Manager	4	8

Level

Business Analyst	1	3.5

Junior Consultant	2	5
Senior Consultant	3	6
Manager	4	8

Find Mean:	**3.5**		**Find Mean:**	**6.33**
$m1 = (3.5 - 3.5)^2$			**$m2 = (5 - 6.33)^2 + (6 - 6.33)^2 + (8 - 6.33)^2$**	
m1 = 0			**m2 = 4.67**	

R1 = m1 + m2 = 4.67

Position	Level	Salary
Business Analyst	1	3.5
Junior Consultant	2	5
Senior Consultant	3	6
Manager	4	8

Level

Business Analyst	1	3.5
Junior Consultant	2	5

Senior Consultant	3	6
Manager	4	8

Find Mean:	4.25		Find Mean:	7
$m1 = (3.5 – 4.25)^2 + (5 – 4.25)$			$m2 = (6 – 7)^2 + (8 - 7)^2$	
m1 = 1.13			m2 = 2	

R2 = m1 + m2 = 3.13

Position	Level	Salary
Business Analyst	1	3.5
Junior Consultant	2	5
Senior Consultant	3	6
Manager	4	8

Level

Business Analyst	1	3.5
Junior Consultant	2	5
Senior Consultant	3	6

Manager	4	8

Find Mean:	4.83		Find Mean:	8
$m1 = (3.5 - 4.83)^2 + (5 - 4.83)^2 + (6 - 4.83)^2$			$m2 = (8 - 8)^2$	
m1 = 3.17			m2 = 0	

R3 = m1 + m2 = 3.17

Level <= 1.5

R1 = m1 + m2 =

Level <= 2.5

R1 = m1 + m2 =

Level <= 3.5

R1 = m1 + m2 =

Position	Level	Salary
Business Analyst	1	3.5
Junior Consultant	2	5
Senior Consultant	3	6
Manager	4	8

Level <= 2.5

Business Analyst	1	3.5
Junior Consultant	2	5

Senior Consultant	3	6
Manager	4	8

Level <= 1.5

Business Analyst	1	3.5

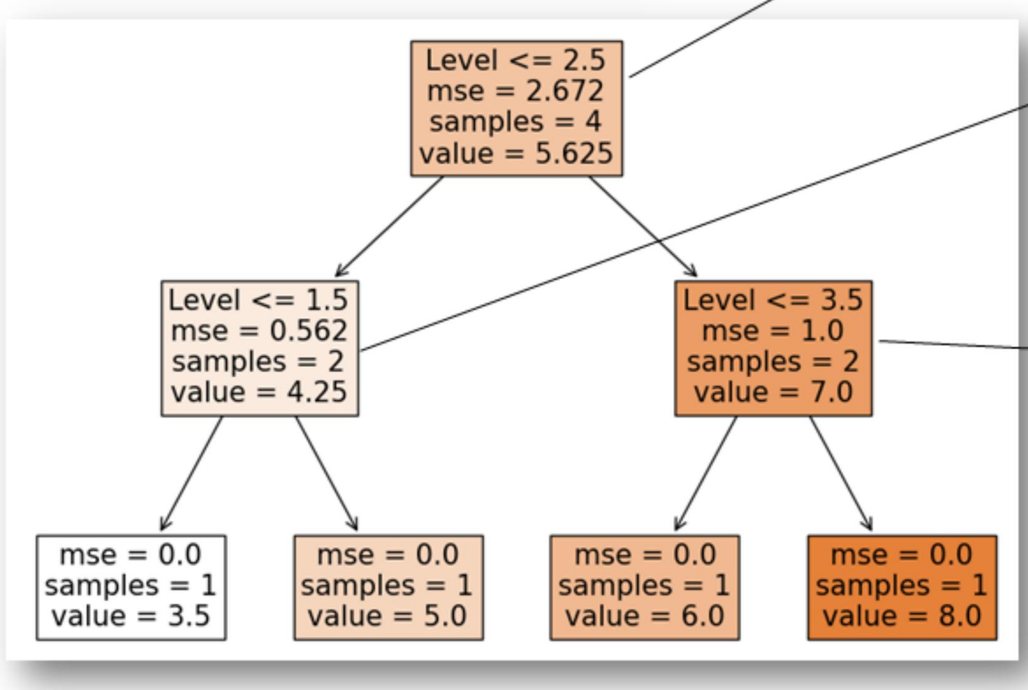
Junior Consultant	2	5

Level <= 3.5

Senior Consultant	3	6

Manager	4	8

Position	Level	Salary
Business Analyst	1	3.5
Junior Consultant	2	5
Senior Consultant	3	6
Manager	4	8



Level <= 2.5

Mse = $(3.5-5.63)^2 + (5-5.63)^2 + (6-5.63)^2 + (8-5.63)^2 / 4 = 2.67$

Samples = 4

Value = (3.5 + 5 + 6 + 8) / 4 = 5.625

Level <= 1.5

Mse = $(3.5-4.25)^2 + (5-4.25)^2 / 2 = 2.67$

Samples = 2

Value = (3.5 + 5) / 2 = 4.25

Level <= 3.5

Mse = $(7-6)^2 + (7-8)^2 / 2 = 2$

Samples = 2

Value = (6 + 8) / 2 = 7

Advantages:

1. Decision trees can be used to predict both continuous and discrete values i.e. they work well for both regression and classification tasks.
2. They require relatively less effort for training the algorithm.
3. They can be used to classify non-linearly separable data.
4. They're very fast and efficient compared to KNN and other classification algorithms.

Disadvantages:

It may have an Overfitting issue, which can be resolved using the **Random Forest algorithm.**