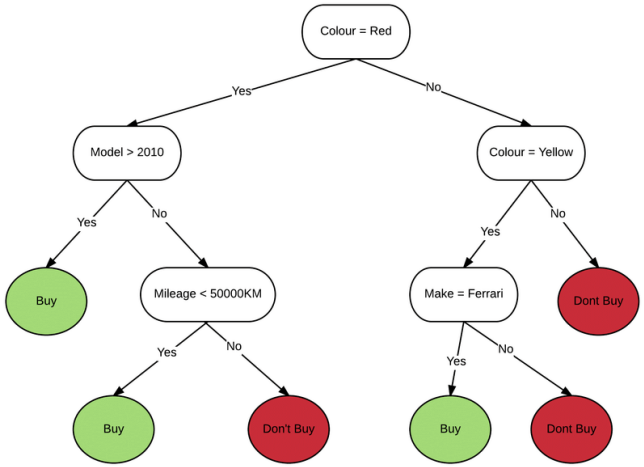
Decision Tree

Decision Tree

- Decision Tree is a Supervised learning technique that can be used for both classification and Regression problems, but mostly it is preferred for solving Classification problems.
- 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.
- 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.
- It is a graphical representation for getting all the possible solutions to a problem/decision based on given conditions.
- A decision tree simply asks a question, and based on the answer (Yes/No), it further split the tree into subtrees.



Decision Tree

Decision Tree Terminologies

Root Node: Root node is from where the decision tree starts. It represents the entire dataset, which further gets divided into two or more homogeneous sets.

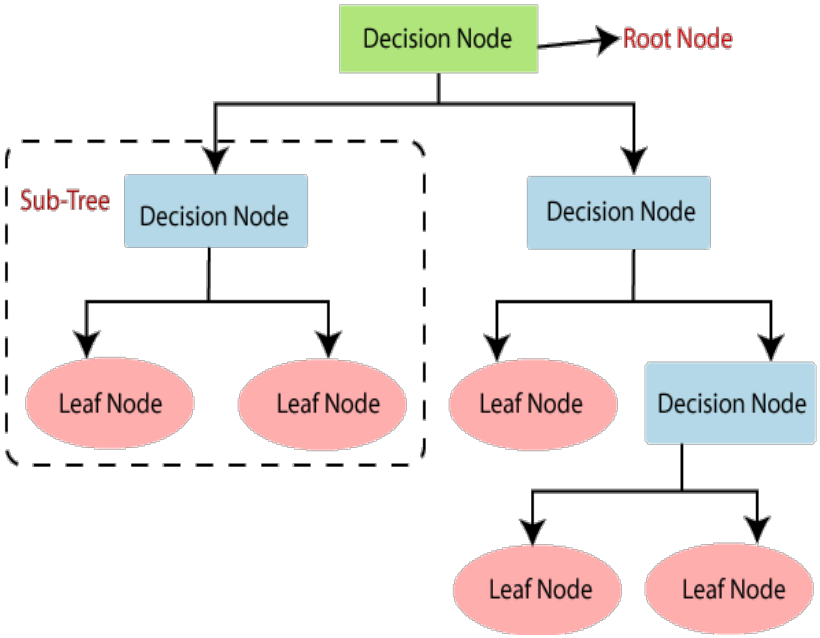
Leaf Node: Leaf nodes are the final output node, and the tree cannot be segregated further after getting a leaf node.

Splitting: Splitting is the process of dividing the decision node/root node into sub-nodes according to the given conditions.

Branch/Sub Tree: A tree formed by splitting the tree.

Pruning: Pruning is the process of removing the unwanted branches from the tree.

Parent/Child node: The root node of the tree is called the parent node, and other nodes are called the child nodes.



How decision tree algorithm works?

In a decision tree, for predicting the class of the given dataset, the algorithm **starts from the root node** of the tree. This algorithm **compares** the **values of root attribute** with the **record (real dataset) attribute** and, based on the comparison, follows the branch and jumps to the next node.

For the next node, the algorithm again compares the attribute value with the other sub-nodes and move further. It continues the process until it reaches the leaf node of the tree.

Steps of Decision Tree

- **Step-1:** Begin the tree with the root node, says S, which contains the complete dataset.
- **Step-2:** Find the best attribute in the dataset using **Attribute Selection Measure (ASM).**
- **Step-3:** Divide the S into subsets that contains possible values for the best attributes.
- **Step-4:** Generate the decision tree node, which contains the best attribute.
- **Step-5:** Recursively make new decision trees using the subsets of the dataset created in **step -3.** Continue this process until a stage is reached where you cannot further classify the nodes and called the final node as a leaf node.

Attribute Selection Measure (ASM)

- While implementing a Decision tree, the main issue arises that how to select the best attribute for the root node and for sub-nodes. So, to solve such problems there is a technique which is called as Attribute selection measure or ASM.
- By this measurement, we can easily select the best attribute for the nodes of the tree.

There are two popular techniques for ASM, which are:

1) **Information Gain**
2) **Gini Index**

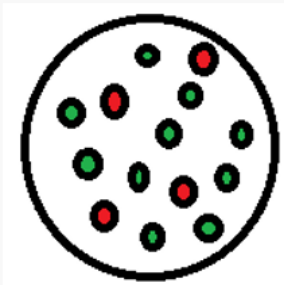
Entropy

Entropy: Entropy is a metric to measure the **impurity** in a given attribute. It specifies randomness in data.
Entropy can be calculated as:

$$H(S) = -\sum_{i=1}^{N} p_i \log_2 p_i$$

where:

- *S – set of all instances in the dataset*
- *N – number of distinct class values*
- *pi – event probability*

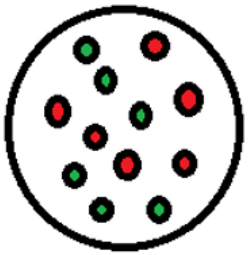


In this group, we have 14 circles, out of which 10 are green (10/14) and 4 are red (4/14). Let's find the entropy of this group.

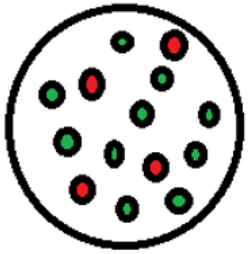
Entropy(Green) = $\log_2(10/14) = -0.4854268272$
Entropy(Red) = $\log_2(4/14) = -1.807354922$

Entropy (Set) = $-(10/14)(-0.4854268272) - (4/14)(-1.807354922) =$ **0.86**

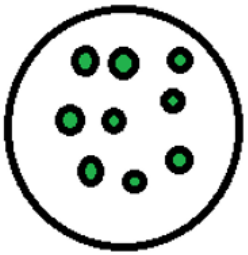
Very Impure



Less Impure



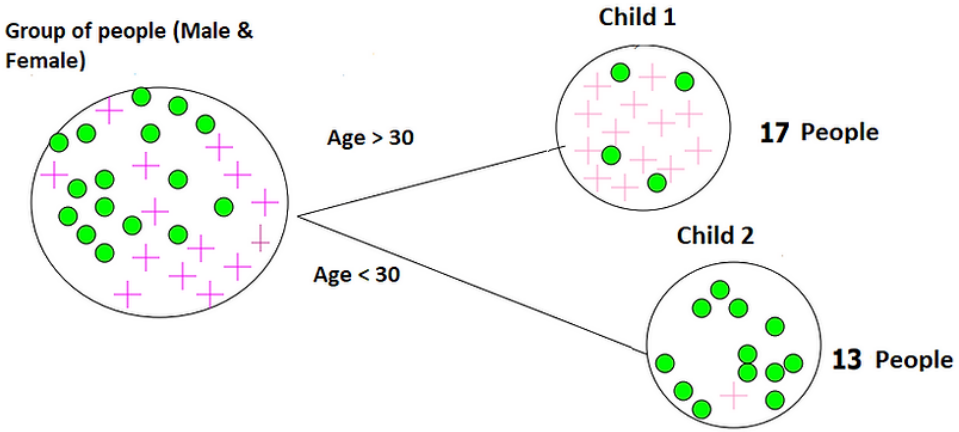
Minimum Impurity



Information Gain

Information Gain:

- Information gain is the measurement of changes in entropy after the segmentation of a dataset based on an attribute.
- It calculates how much information a feature provides us about a class.
- According to the value of information gain, we split the node and build the decision tree.
- A decision tree algorithm always tries to maximize the value of information gain, and a node/attribute having the highest information gain is split first. It can be calculated using the below formula:



parent entropy $-\left(\frac{14}{30}\cdot\log\frac{14}{30}\right)-\left(\frac{16}{30}\cdot\log\frac{16}{30}\right)=$ **0.996**

child **1** entropy $-\left(\frac{13}{17}\cdot\log\frac{13}{17}\right)-\left(\frac{4}{17}\cdot\log\frac{4}{17}\right)=$ **0.787**

child 2 entropy $-\left(\frac{1}{13}\cdot\log\frac{1}{13}\right)-\left(\frac{12}{13}\cdot\log\frac{12}{13}\right)=$ **0.391**

(Weighted) Average Entropy of children = $\left(\frac{17}{30}\cdot 0.787\right)+\left(\frac{13}{30}\cdot 0.391\right)=0.615$

Information Gain - 0.996 - 0.615 = 0.38 for this split

Gini Index

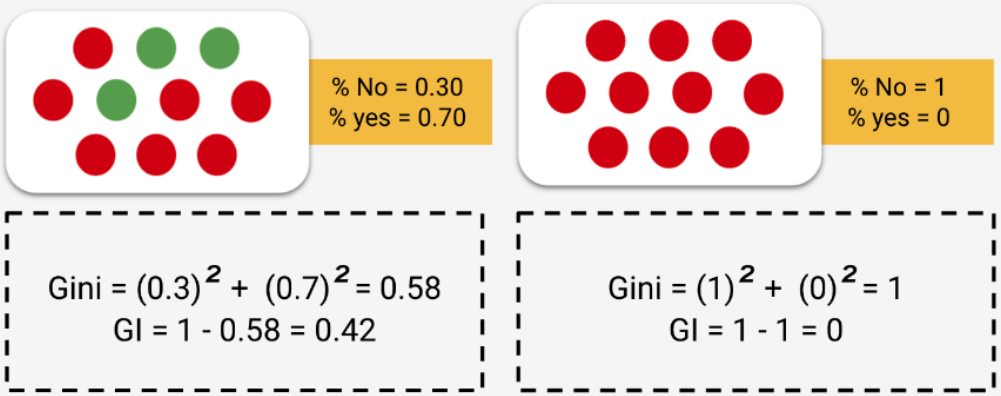
Gini Index

- Gini index is a measure of impurity used while creating a decision tree in the CART(Classification and Regression Tree) algorithm.
- An attribute with the low Gini index should be preferred as compared to the high Gini index.
- It only creates binary splits, and the CART algorithm uses the Gini index to create binary splits.
- Gini index can be calculated using the below formula:

$$Gini = 1 - \sum_j p_j^2$$

$$GI = 1 - Gini$$

$$Gini = p_1^2 + p_2^2 + \ldots + p_n^2$$



Example

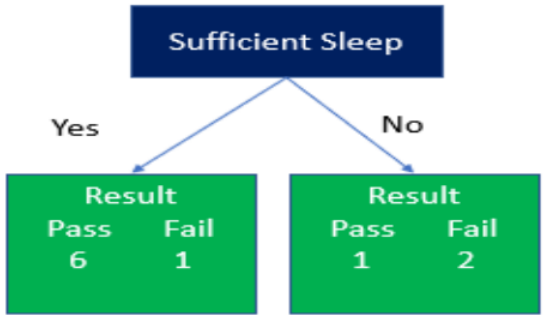
Here we have considered the example to build the tree.
Here we have 4 variables of which
Result is our dependent (or outcome) variable which is a categorical variable having 2 categories (Pass /fail) and
We have 3 independent (or predictor) variables which are all categorical variables.

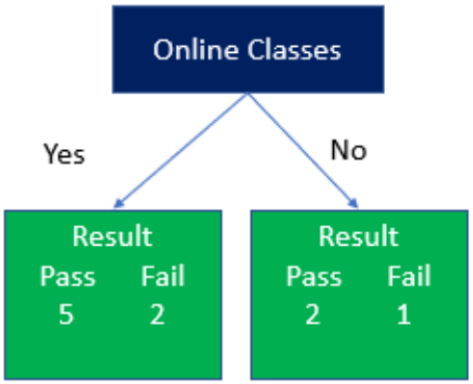
Sufficient Sleep	Online Classes	Outdoor Activity	Result
No	Yes	Yes	Fail
Yes	Yes	No	Pass
Yes	No	Yes	Pass
Yes	Yes	No	Pass
No	Yes	No	Pass
Yes	No	No	Fail
Yes	Yes	Yes	Pass
Yes	No	Yes	Pass
Yes	Yes	Yes	Pass
No	Yes	Yes	Fail

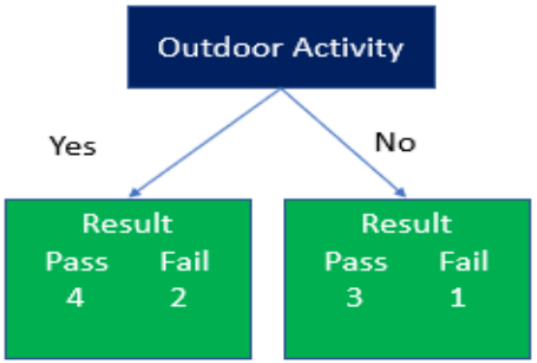
Decision of root node:
Steps:

1. First, we find out Gini for each leaf node using above formula:
2. Then we compute Gini index of feature by taking weighted average of Gini of each leaf node.
3. The feature having least Gini value is considered as root node

Example







Gini (Sufficient Sleep = Yes) = $1 - [(\frac{6}{7})^2 + (\frac{1}{7})^2] = 0.245$

Gini (Sufficient Sleep = No) = $1 - [(\frac{1}{3})^2 + (\frac{2}{3})^2] = 0.444$

Step 2: Calculate weighted average for sufficient sleep:

$= (\frac{7}{10}) * 0.245 + (\frac{3}{10}) * 0.444$

= 0.305

So, we have Gini of sufficient sleep as 0.305 .

Gini (Online Classes = Yes) = $1 - [(\frac{5}{7})^2 + (\frac{2}{7})^2] = 0.408$

Gini (Online Classes = No) = $1 - [(\frac{2}{3})^2 + (\frac{1}{3})^2] = 0.444$

weighted average for online classes:

$= (\frac{7}{10}) * 0.408 + (\frac{3}{10}) * 0.444$

= 0.419

So, we have Gini of online classes as 0.419 .

Gini (Outdoor Activity = Yes) = $1 - [(\frac{4}{6})^2 + (\frac{2}{6})^2] = 0.444$

Gini (Outdoor Activity = No) = $1 - [(\frac{3}{4})^2 + (\frac{1}{4})^2] = 0.375$

weighted average for online classes:

$= (\frac{6}{10}) * 0.444 + (\frac{4}{10}) * 0.375$

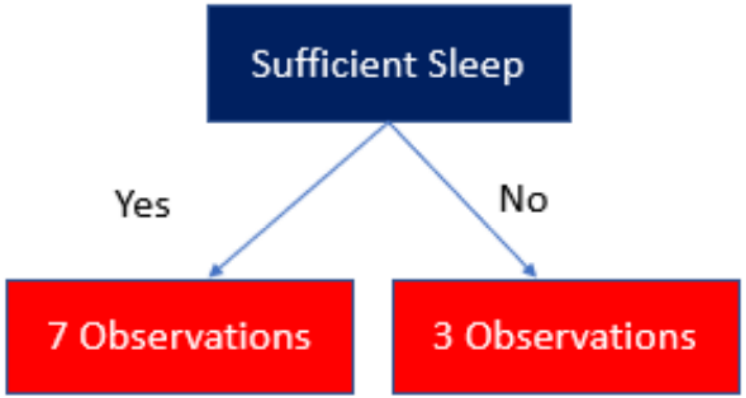
= 0.416

So, we have Gini of Outdoor Activity as 0.416 .

Example

Predictor	Gini
Sufficient Sleep	0.305
Online Classes	0.416
Outdoor Activity	0.343

So, we can clearly see that Sufficient sleep has least value of Gini and it will be our Root node.



Further splits on Sufficient Sleep = "Yes"

First let's consider the students who have "sufficient Sleep = Yes"

Again, we will follow the same approach but now only with remaining 2 features (online Classes , Outdoor Activity)

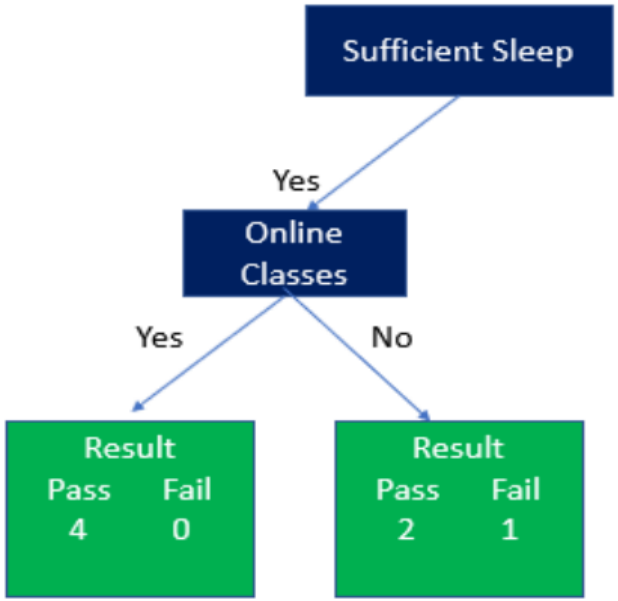
Gini for "Sufficient sleep = Yes" = $1 - [(\frac{6}{7})^2 + (\frac{1}{7})^2] = 0.245$

If further splits of "Sufficient sleep = Yes" node gives less value of Gini index, then we will make a split else we won't.

Let's calculate Gini for online Classes at internal node.

Sufficient Sleep	Online Classes	Outdoor Activity	Result
Yes	Yes	No	Pass
Yes	No	Yes	Pass
Yes	Yes	No	Pass
Yes	No	No	Fail
Yes	Yes	Yes	Pass
Yes	No	Yes	Pass
Yes	Yes	Yes	Pass

Further splits on Sufficient Sleep = "Yes"



Gini for "Sufficient sleep = Yes" = $1 - [(\frac{6}{7})^2 + (\frac{1}{7})^2] = 0.245$

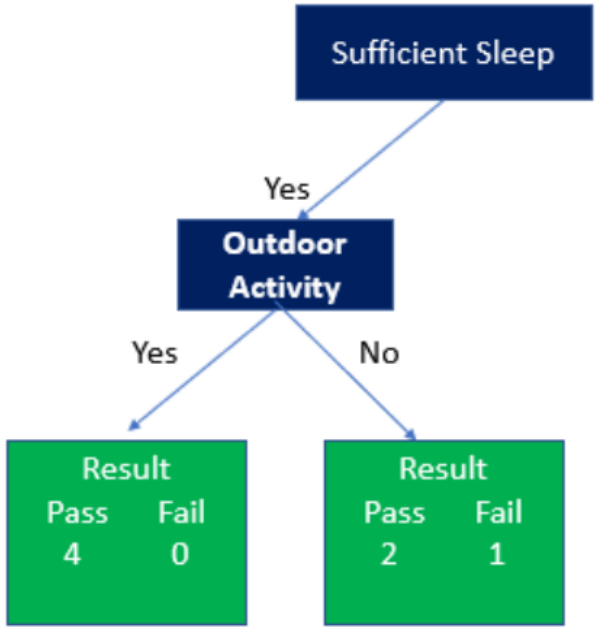
Gini (Sufficient Sleep = Yes & online Classes = Yes) = $1 - [(\frac{4}{4})^2 + (\frac{0}{4})^2] = 0$

Gini (Sufficient Sleep = Yes & online Classes = No) = $1 - [(\frac{2}{3})^2 + (\frac{1}{3})^2] = 0.444$

weighted average for online classes:

$= (\frac{4}{7}) * 0 + (\frac{3}{7}) * 0.444$

$= 0.191$



Gini (Sufficient Sleep = Yes & Outdoor Activity = Yes) = $1 - [(\frac{4}{4})^2 + (\frac{0}{4})^2] = 0$

Gini (Sufficient Sleep = Yes & Outdoor Activity = No) = $1 - [(\frac{2}{3})^2 + (\frac{1}{3})^2] = 0.444$

weighted average for online classes:

$= (\frac{4}{7}) * 0 + (\frac{3}{7}) * 0.444$

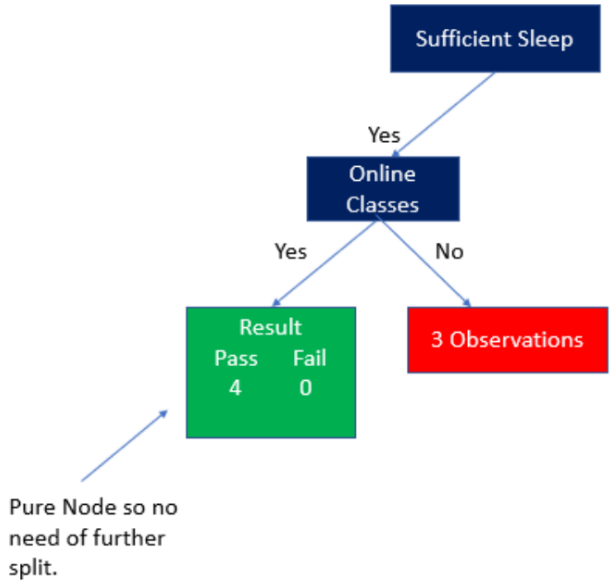
$= 0.191$

Further splits on Sufficient Sleep = "Yes"

Predictor	Gini
Online Classes	0.191
Outdoor Activity	0.191

Here both Online classes and Outdoor activity give us equal separation so we can take any of these as our 1st internal node.

Let us take Online classes as our 1st internal node.



So, when "sufficient sleep = Yes & Online classes = Yes" we get a pure node and we don't have to split that node any further

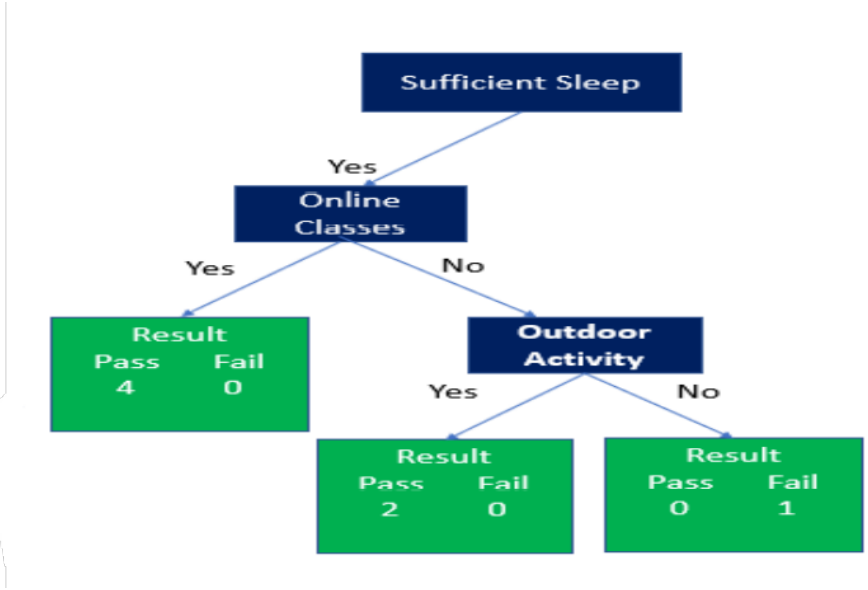
Further splits on sufficient sleep = Yes & Online classes = No

Let's consider the node where when "sufficient sleep = Yes & Online classes = No"

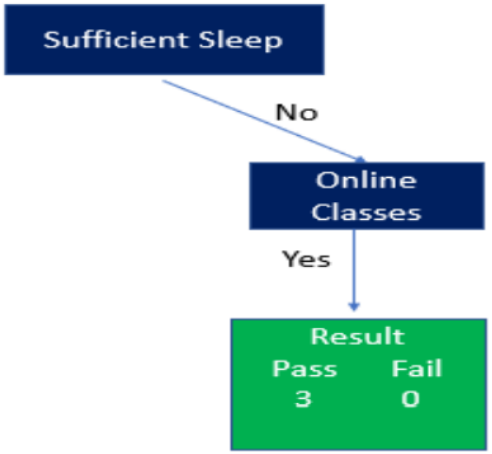
Here we are left with only one variable outdoor activity .

If gives better split than "sufficient sleep = Yes & Online classes = No" then we will split the node else, we will stop.

Sufficient Sleep	Online Classes	Outdoor Activity	Result
Yes	No	Yes	Pass
Yes	No	No	Fail
Yes	No	Yes	Pass

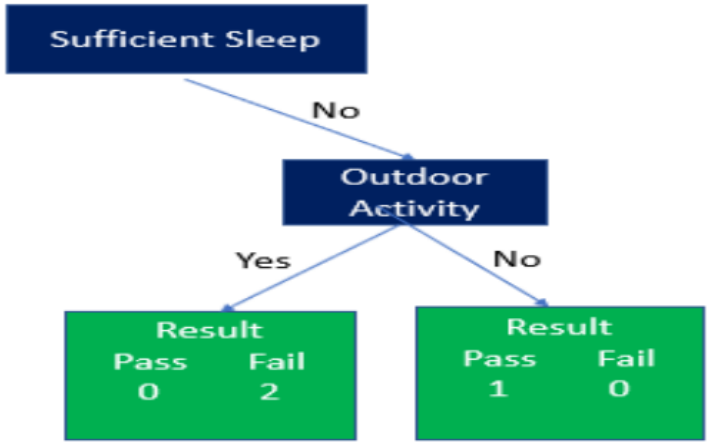


Further splits on sufficient sleep = No



Gini (Sufficient Sleep = No & online Classes = Yes) = $1 - [(\frac{3}{3})^2 + (\frac{0}{3})^2] = 0$

and as we have only one leaf so weighted average Gini = 0



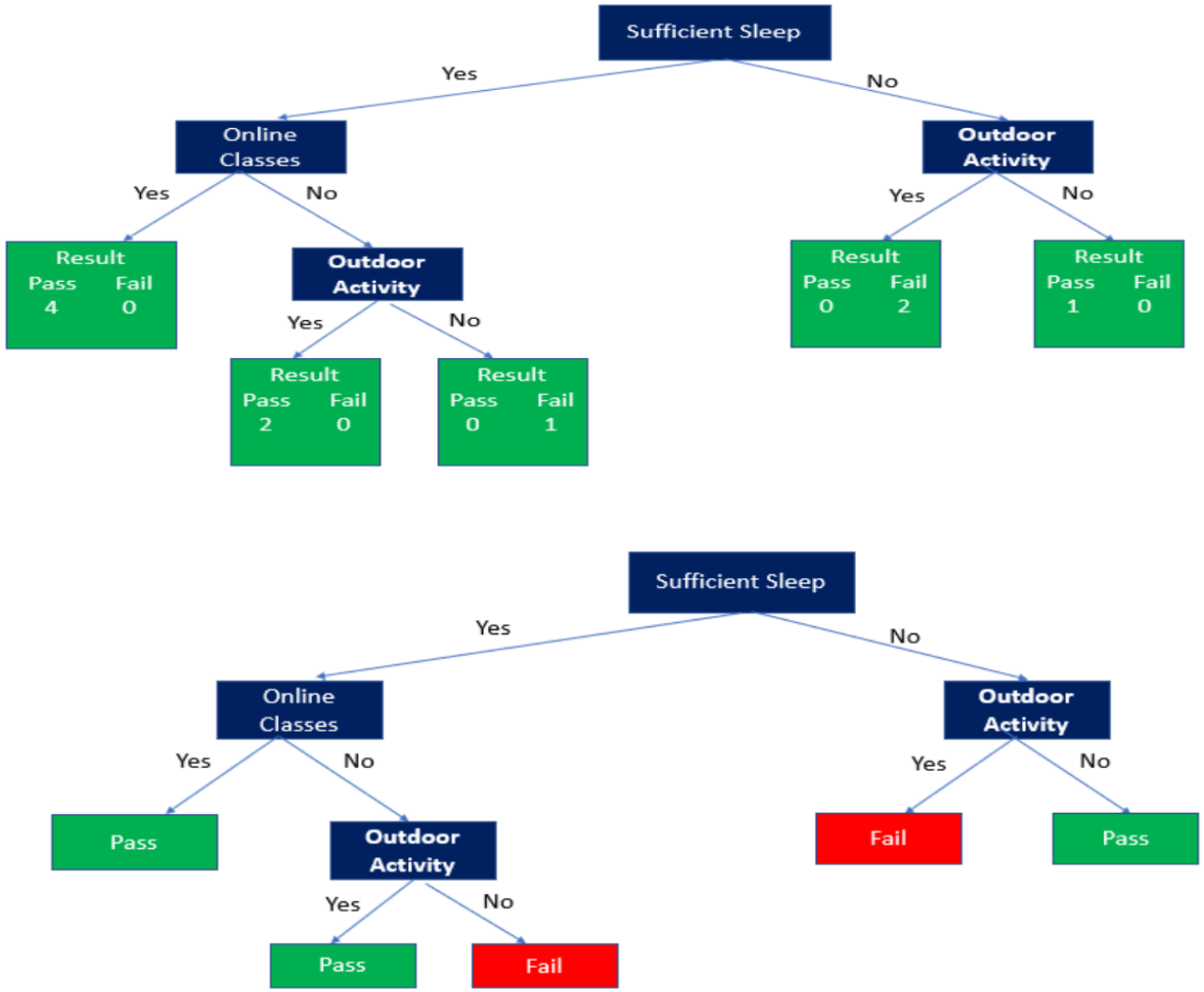
Gini (Sufficient Sleep = No & Outdoor Activity = Yes) = $1 - [(\frac{0}{2})^2 + (\frac{2}{2})^2] = 0$

Gini (Sufficient Sleep = No & Outdoor Activity = No) = $1 - [(\frac{1}{1})^2 + (\frac{0}{1})^2] = 0$

So, this also gives perfect separation with weighted Gini = 0

So ,we can either take Online classes or Outdoor activity as our 1st internal node and that will be our final node as either of them will give complete separation.
Let's take Outdoor activity as our internal node.
We are done with our tree.

Further splits on sufficient sleep = No



Pruning in Decision Tree

- Pruning is one of the techniques that is used to overcome our problem of Overfitting.
- Pruning, in its literal sense, is a practice which involves the selective removal of certain parts of a tree(or plant), such as branches, buds, or roots, to improve the tree's structure, and promote healthy growth. This is exactly what Pruning does to our Decision Trees as well.
- It reduces the size of a Decision Tree which might slightly increase your training error but drastically decrease your testing error, hence making it more adaptable.
- **Minimal Cost-Complexity Pruning** is one of the types of Pruning of Decision Trees.

This algorithm is parameterized by $\alpha(\geq 0)$ known as the complexity parameter.
The complexity parameter is used to define the cost-complexity measure, $R\alpha(T)$ of a given tree T:

$$R_\alpha(T)=R(T)+\alpha|T|$$

where $|T|$ is the number of terminal nodes in T and R(T) is traditionally defined as the total misclassification rate of the terminal nodes.