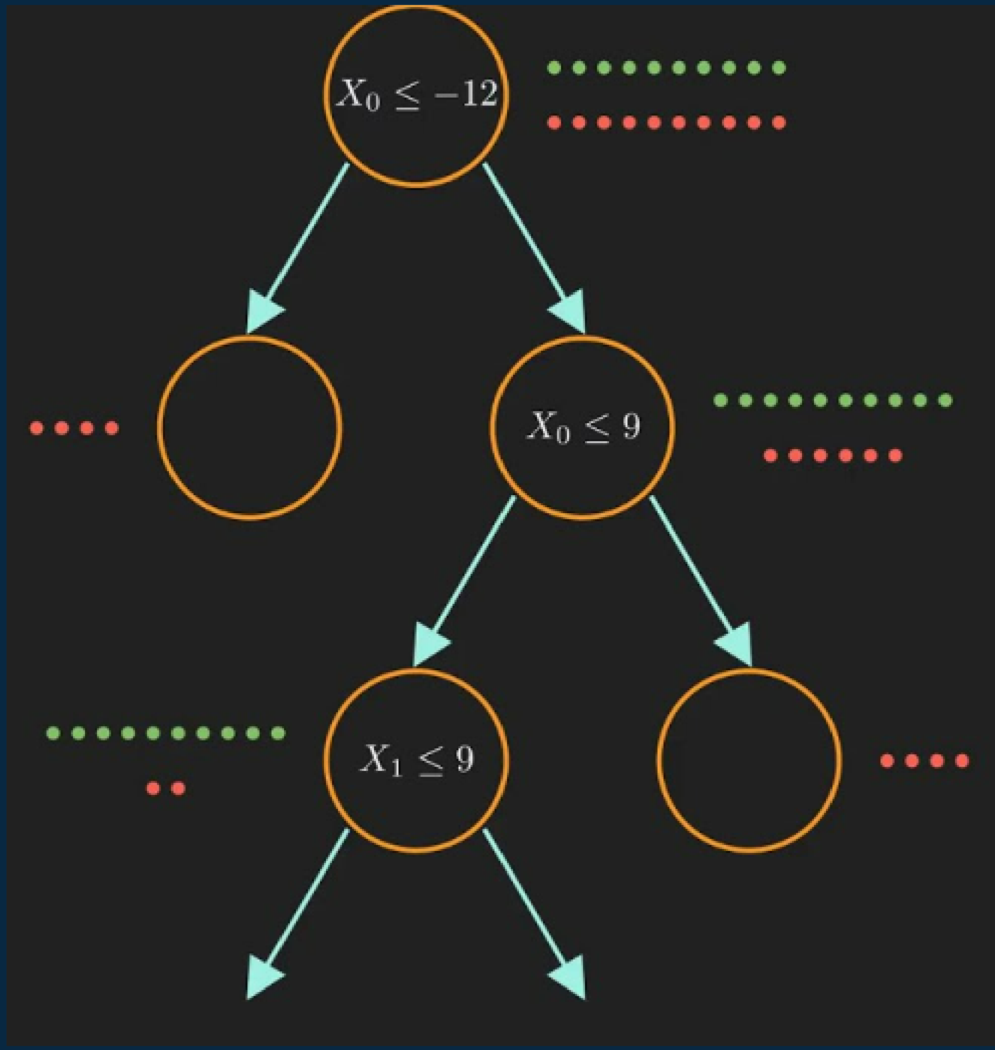
DECISION TREES

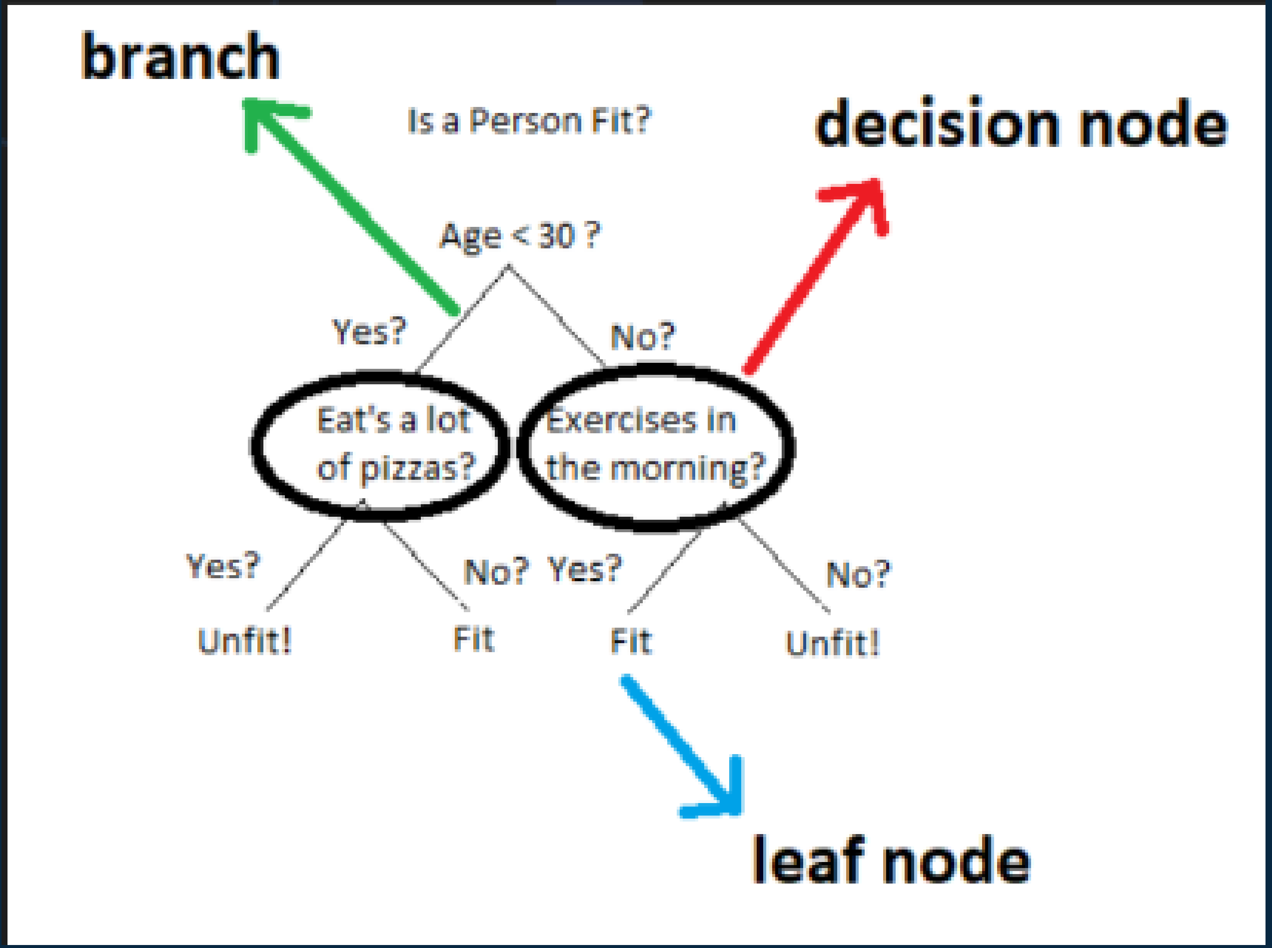
WWW.TANTROSTAV.CO.IN

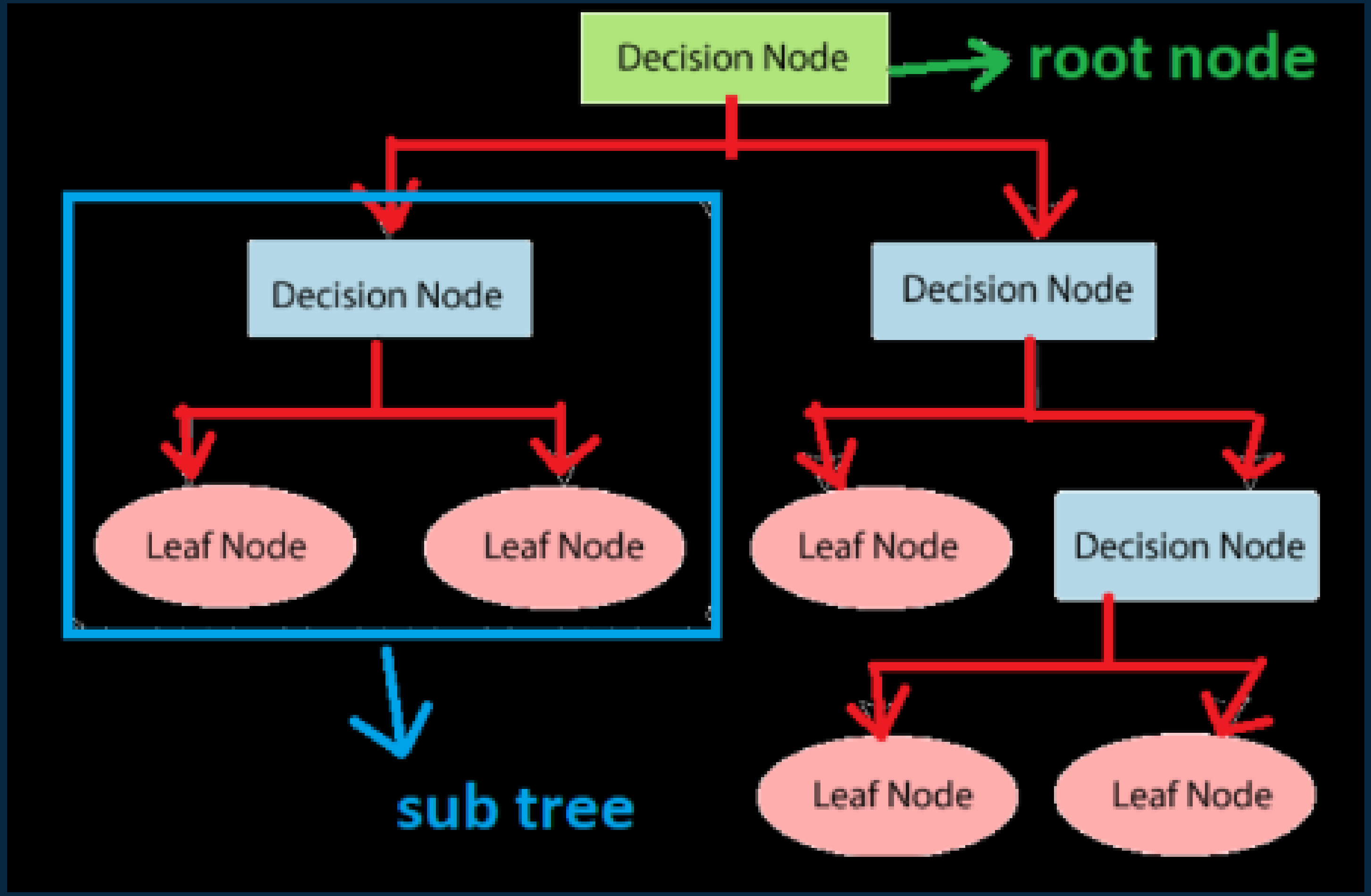
- Decision Trees are a type of Supervised Machine Learning where the data is continuously split according to a certain parameter.

- The tree can be explained by two entities, namely decision nodes and leaves. The leaves are the decisions or the final outcomes. And the decision nodes are where the data is split.



Decision trees

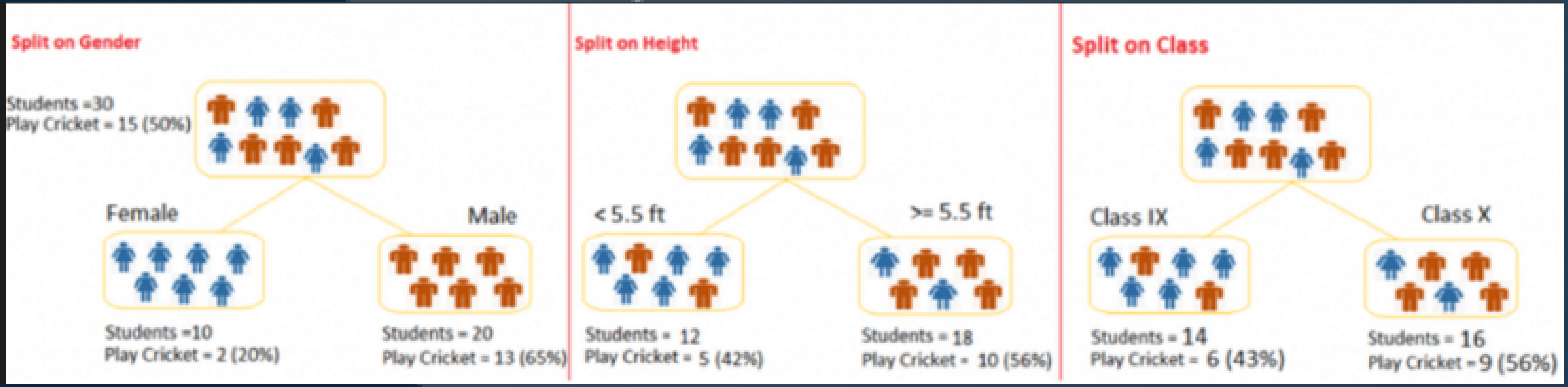




How does it work?

- Decision tree splits the sample on all available variables. Then it will select the variable which gives the most homogeneous sets.

- Let us understand this with an example. Suppose we have a 30 students and we want to classify them into groups - one group which likes cricket and other group which doesn't. Let us suppose we are given information about the gender of the students in the class, their heights and their class.



- So now let us observe the three possible cases, split based on gender, split based on height, and split based on class.

- We can notice that clearly split on gender is the winner because it performs better in segregating the students into cricket lovers and non-cricket lovers.

- This is what we deal with in decision trees, choosing the features which can best classify the given sample. Let us briefly look into the math involved in this process.

The Math

- To measure our decision tree performance, we use impurity. Impurity gives us an idea of how well the decision trees split the sample into homogeneous sets.

- Decision tree decides which feature to be considered for splitting based on the impurity associated with the split. Lesser the impurity better the split.

- There are two major impurities measures that are commonly used :- Gini Index and Entropy.

$$Gini = 1 - \sum_{i=1}^{C} (p_i)^2$$

$$Entropy = -\sum_{i=1}^{n} p_i * \log(p_i)$$

- Let us understand the working of decision trees using a classic example assisted by the math that we have understood so far.
- Predict whether you can play outside or not based on the weather conditions

Day	outlook	temperature	humidity	wind	Decision
1	sunny	hot	high	weak	No
2	sunny	hot	high	strong	No
3	overcast	hot	high	weak	Yes
4	rainfall	mild	high	weak	Yes
5	rainfall	cool	normal	weak	Yes
6	rainfall	cool	normal	strong	No
7	overcast	cool	normal	wtrong	Yes
8	sunny	mild	high	weak	No
9	sunny	cool	normal	weak	Yes
10	rainfall	mild	normal	weak	Yes
11	sunny	mild	normal	strong	Yes
12	overcast	mild	high	strong	Yes
13	overcast	hot	normal	weak	Yes
14	rainfall	mild	high	strong	No

- Now, in this case, we can see there are four parameters outlook, temperature, humidity, and wind. So the decision tree can split the sample on basis of either of the parameters. Let us figure out which combination of splits will result in the best outcome

- We can begin by calculating the Gini index for all four inputs independently, the input which yields least Gini index is the most desirable split.

Outlook	Yes	No	# Instances
sunny	2	3	5
overcast	4	0	4
rainfall	3	2	5

Gini index (outlook=sunny)= $1-(2/5)^2-(3/5)^2 = 1- 0.16–0.36 = 0.48$

Gini index(outlook=overcast)= $1- (4/4)^2-(0/4)^2 = 1- 1- 0 = 0$

Gini index(outlook=rainfall)= $1- (3/5)^2 -(2/5)^2 = 1- 0.36- 0.16 = 0.48$

Now , we will calculate the weighted sum of Gini index for outlook features,

Gini(outlook) = $(5/14)*0.48 + (4/14) *0 + (5/14)*0.48 = 0.342$

Temperature	Yes	No	# Instances
hot	2	2	4
cool	3	1	4
mild	4	2	6

Gini(temperature=hot) = $1-(2/4)^2-(2/4)^2 = 0.5$

Gini(temperature=cool) = $1-(3/4)^2-(1/4)^2 = 0.375$

Gini(temperature=mild) = $1-(4/6)^2-(2/6)^2 = 0.445$

Now, the weighted sum of Gini index for temperature features can be calculated as,

Gini(temperature)= (4/14) *0.5 + (4/14) *0.375 + (6/14) *0.445 =0.439

Humidity	Yes	No	# Instances
high	3	4	7
Normal	6	1	7

Gini(humidity=high) = $1-(3/7)^2-(4/7)^2 = 0.489$

Gini(humidity=normal) = $1-(6/7)^2-(1/7)^2 = 0.244$

Now, the weighted sum of Gini index for humidity features can be calculated as,

Gini(humidity) = (7/14) *0.489 + (7/14) *0.244=0.367

wind	Yes	No	# Instances
weak	6	2	8
strong	3	3	6

Features	Gini Index
outlook	0.342
temperature	0.439
humidity	0.367
wind	0.428

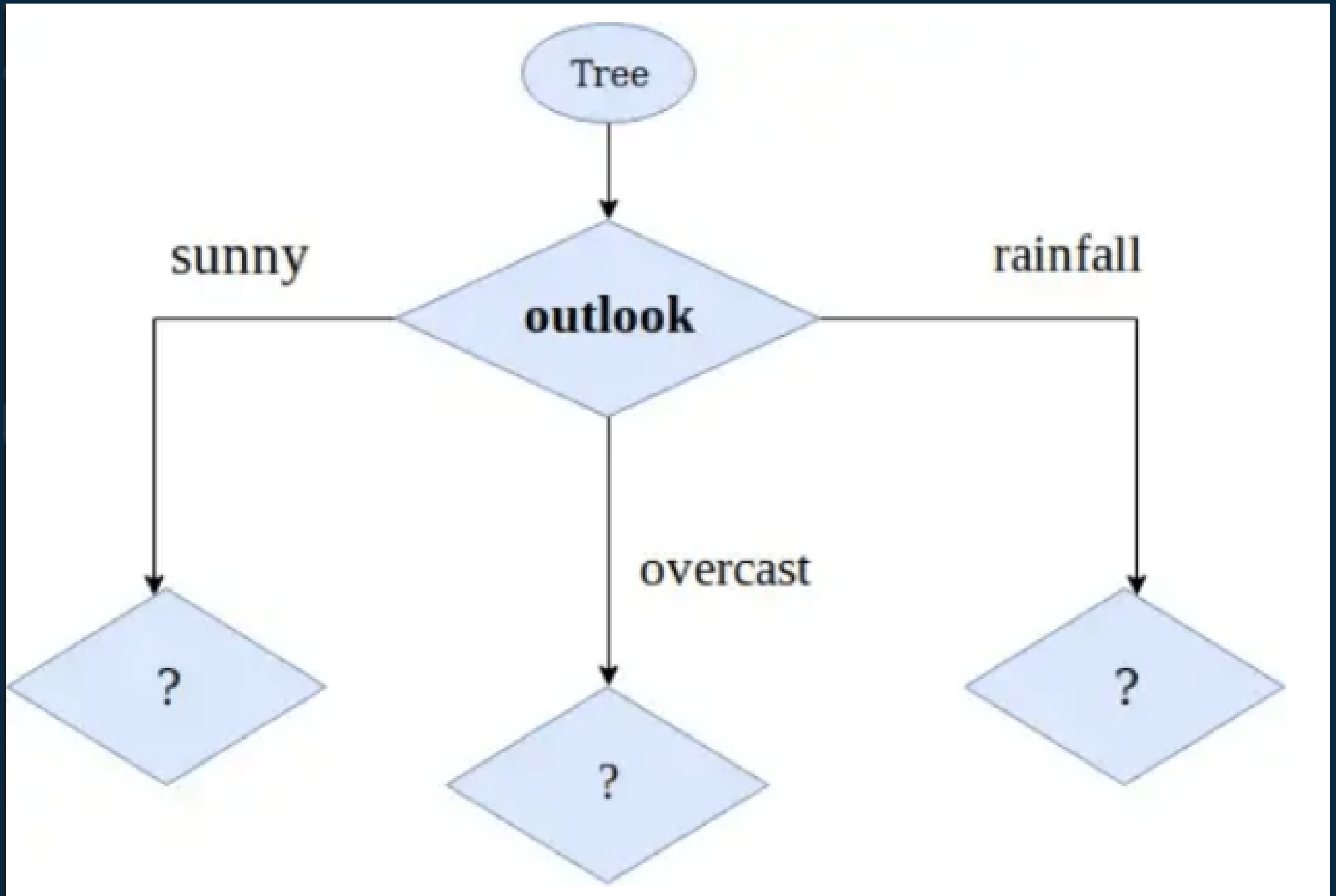
Gini(wind=weak)= $1-(6/8)^2-(2/8)^2 = 0.375$

Gini(wind=strong)= $1-(3/6)^2-(3/6)^2= 0.5$

Now, the weighted sum of Gini index for wind features can be calculated as,

Gini(wind) = $(8/14) *0.375 + (6/14) *0.5=0.428$

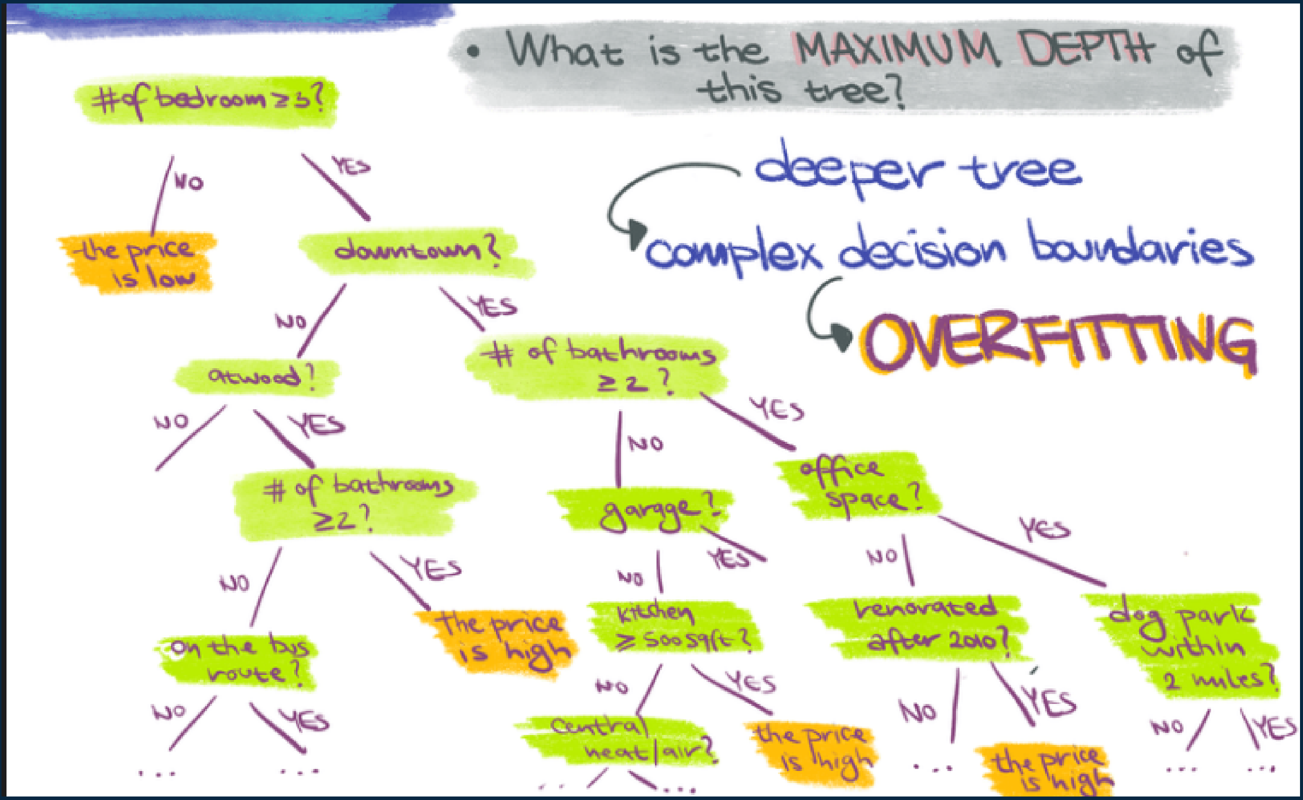
- The Gini index is least for outlook so that will be our root node. In similar way proceed until we obtain the leaf nodes.


Tree
outlook
sunny
rainfall
overcast
?
?
?

Drawbacks of Decision Trees

- Decision Trees generally tend to overfit very easily because the tree branches extend until all the leaf nodes are obtained.

- Decision Trees in general do not perform very well on continuous data i.e they are preferably more used for classification than regression.



How can we get rid of overfitting?

- The best solution to overfitting is hyperparameter optimization but what is hyperparameter optimization?

- For a learning algorithm, there are a few parameters that are to be set, and the process of choosing these parameters to optimize the learning algorithm is referred to as hyperparameter optimization.

HYPERPARAMETER TUNING

HYPERPARAMETER TUNING

- MAX DEPTH:- The maximum depth of the tree. If None, then nodes are expanded until all leaves are pure which often causes overfitting.

- MIN SAMPLES SPLIT:- The minimum number of samples required to split an internal node.

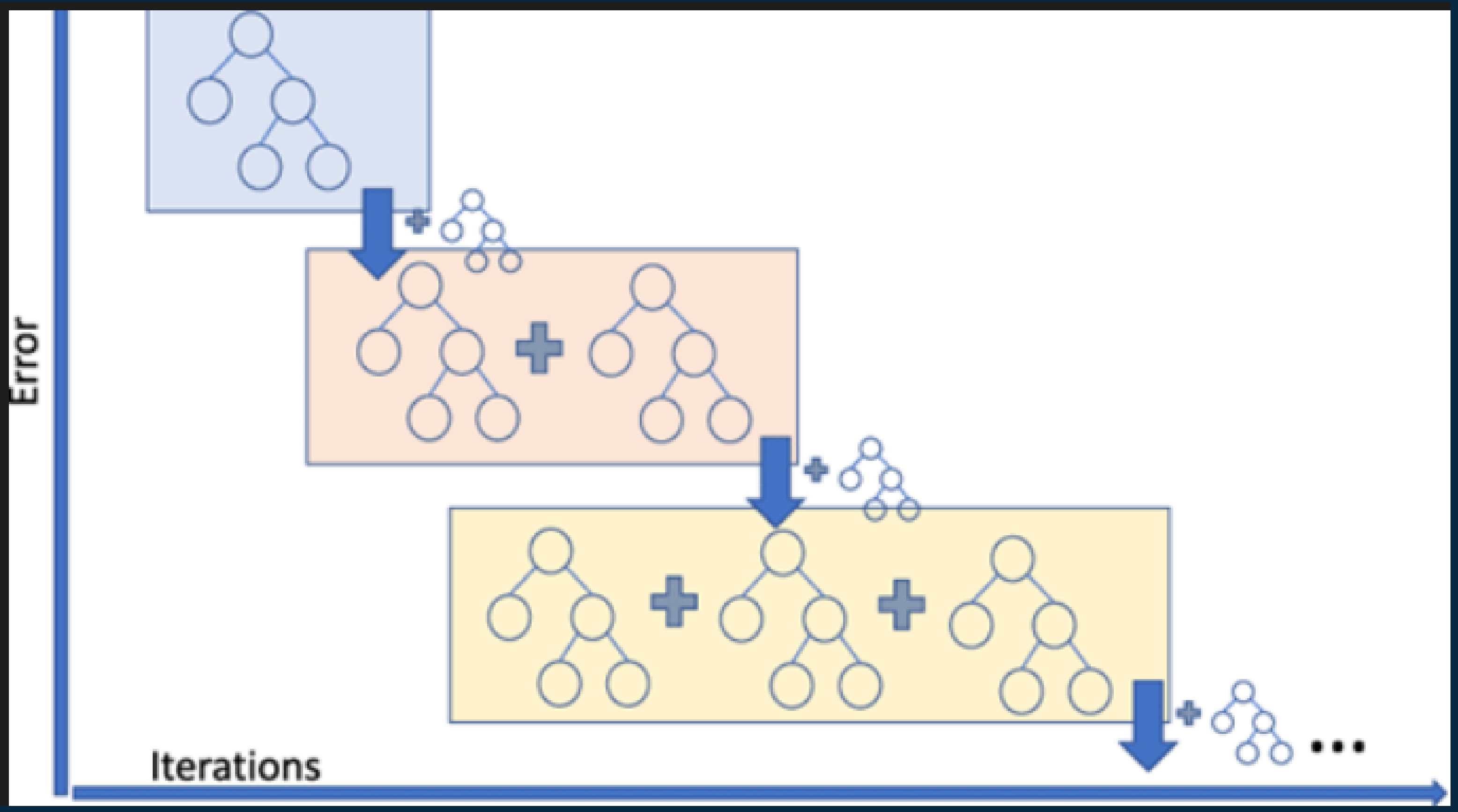
- MIN SAMPLES LEAF:- The minimum number of samples required to be at a leaf node.

- MAX FEATURES:- The number of features to consider when looking for the best split.

Random Forest

- Random forest, like its name implies, consists of a large number of individual decision trees that operate as an ensemble.
- The fundamental concept behind random forest is a simple but powerful one — the wisdom of crowds.
- The additional hyperparameter in the case of Random Forest is the number of estimators i.e. the number of trees that will operate as an ensemble.

Gradient Boosting



- Gradient Boosting is a popular boosting algorithm. In gradient boosting, each predictor corrects its predecessor's error.
- The term " gradient" refers to the fact that each decision tree is trained to reduce the loss from the previous iteration.
- Gradient Boosting models generally outperforms decision trees and random forest models.