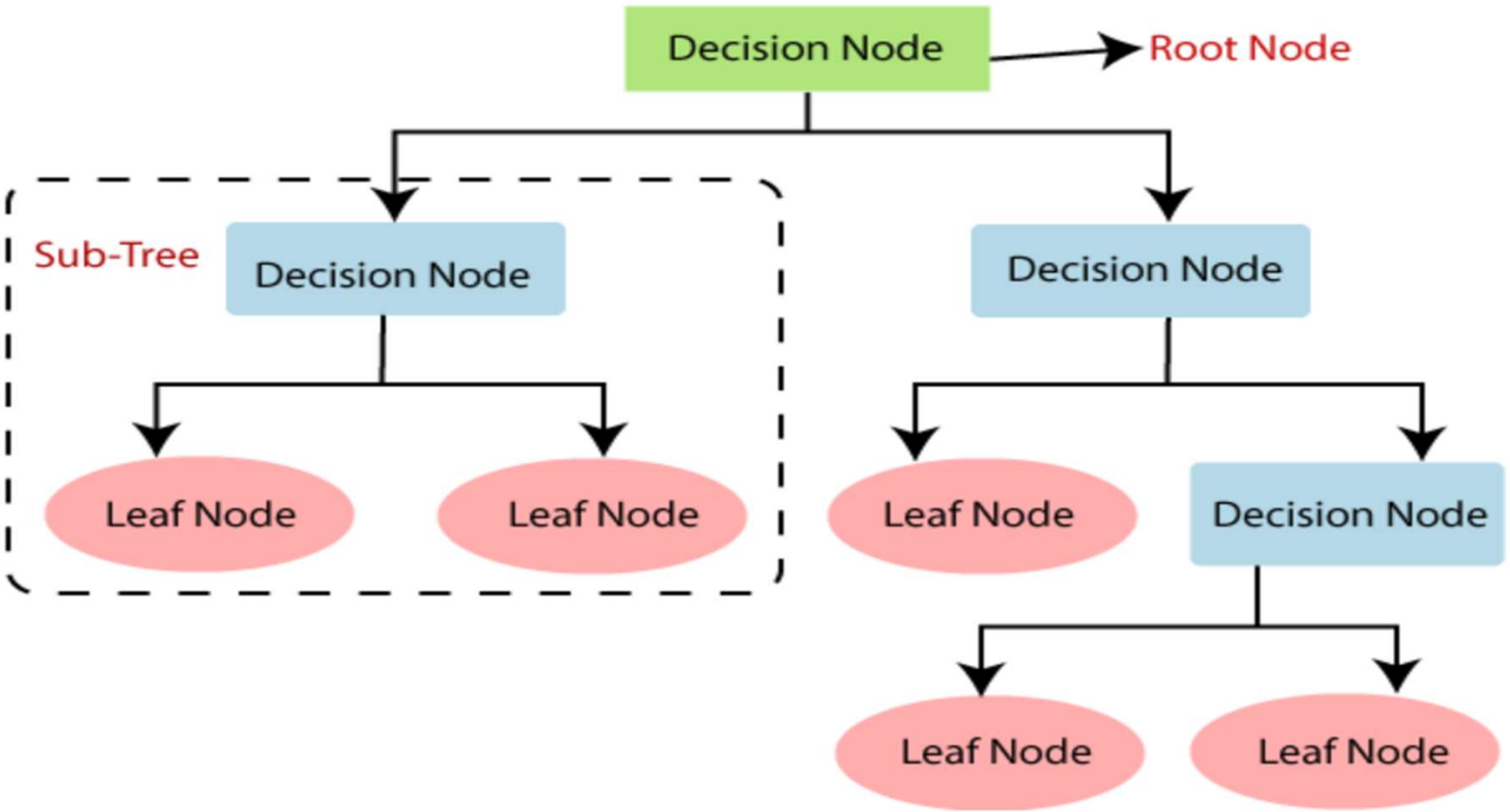
Decision Tree Machine Learning Model

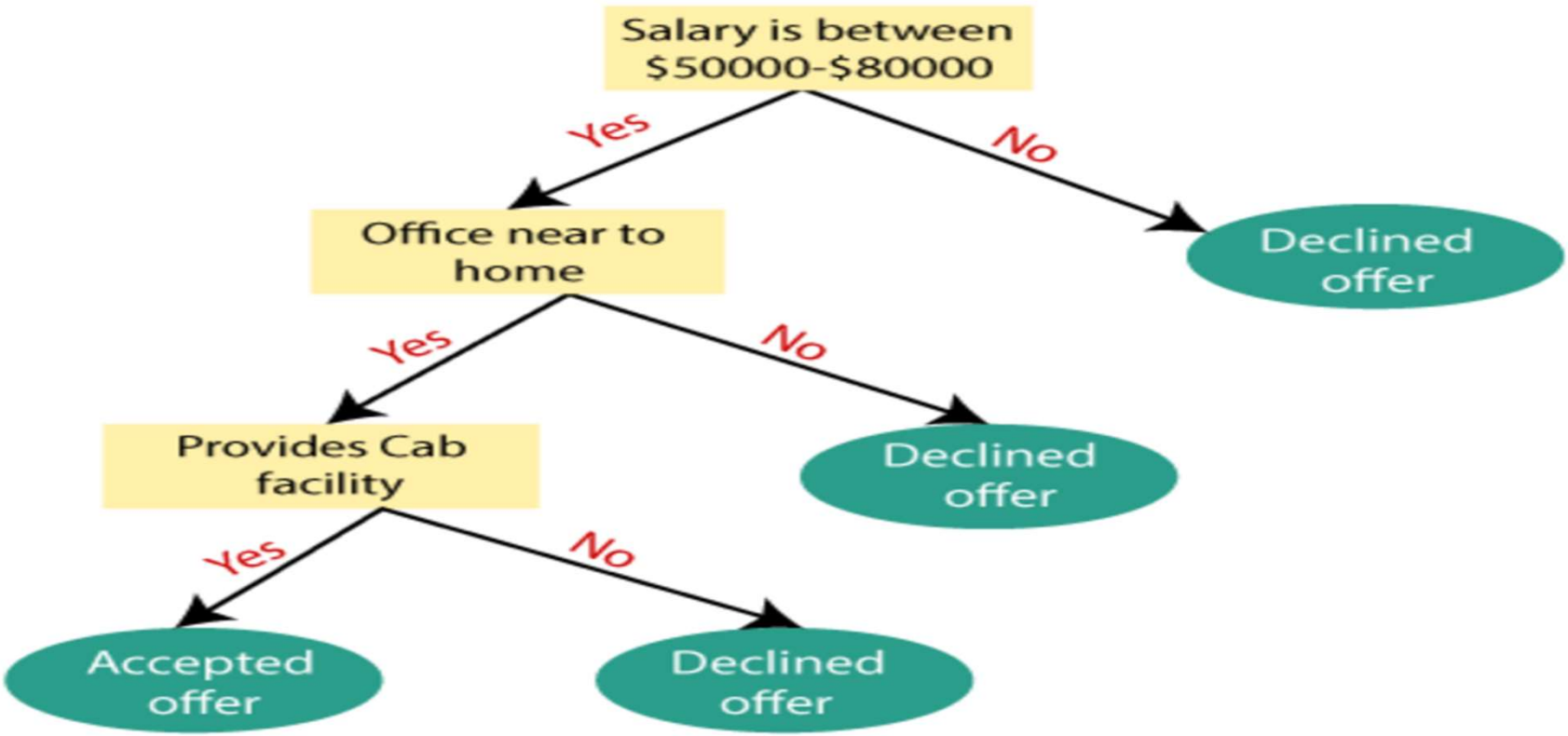
Definition

- Decision Tree is a classification and regression supervised ML model. It can be used for a classification problem and also a regression problem.
- But it is best suited for a classification problem
- In a decision tree the dataset is split into different groups based on the features.
- The aim is to use that feature for splitting which results in a purer group or in other words which results in a better classification.
- Decision Tree Model is based on ‘recursive partitioning’.
- There are different algorithms based on different splitting criteria.

Basic Terminologies

- Root Node: Main node (from where the splitting starts)
- Decision Node: A point where decision about splitting needs to be taken. It is also called ‘condition node’
- Parent / Child Nodes: In a split the split nodes are called child node, and the node from where they split is called parent node.
- Subnode: Same as child node
- Terminal Node/ Leaf Node: Last node of the tree that does not get divided further.
- Pruning: Once the decision tree is completed formed, it may be required to remove some of the splits and reshape the decision tree, this process is called pruning. It is completely opposite of splitting.
- In splitting the tree size increases, in pruning the tree size decreases.


Decision Node
Root Node
Sub-Tree
Decision Node
Decision Node
Leaf Node
Leaf Node
Leaf Node
Decision Node
Leaf Node
Leaf Node

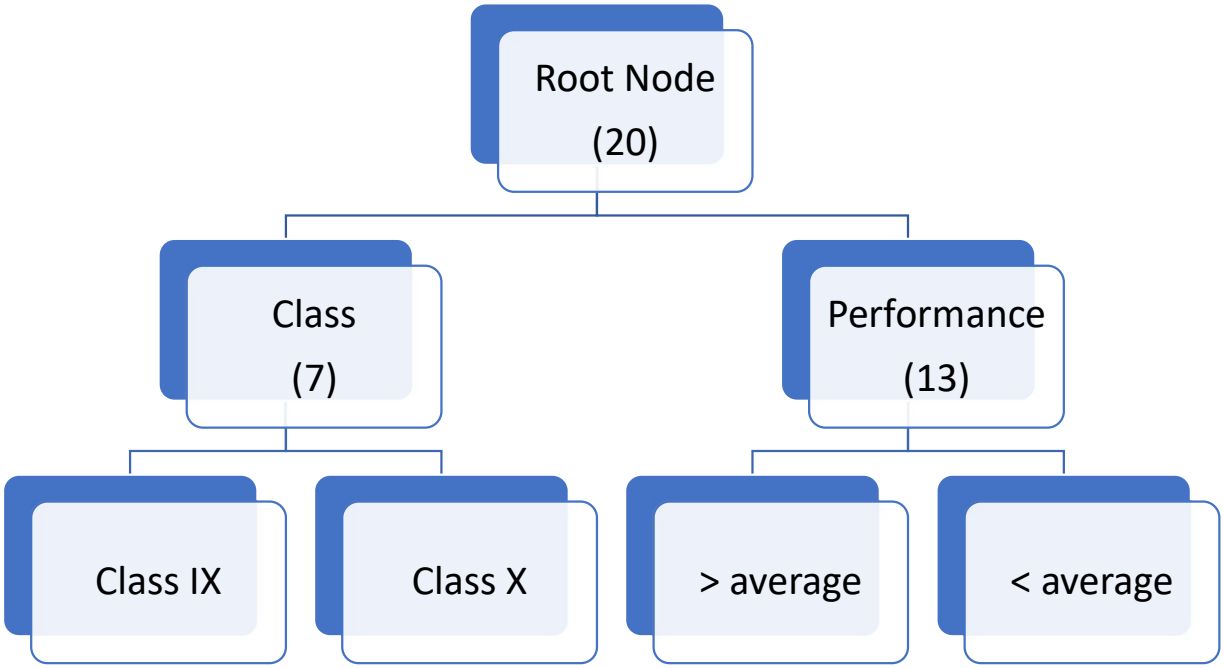

Salary is between $50000-$80000
Yes
No
Office near to home
Declined offer
Yes
No
Provides Cab facility
Declined offer
Yes
No
Accepted offer
Declined offer

Constraints of tree splitting

- A decision tree aims to obtain pure nodes. For this it may go on splitting till the time it gets 100% pure nodes. That is the reason it is also known as a ‘Greedy Algorithm’
- Since we do not want the tree to continue splitting till the last pure node, we predefine some constraints on when to stop splitting on when should it proceed to split.
- Minimum samples on a node to split
- Minimum samples for a terminal node
- Maximum depth
- Maximum number of terminal nodes
- Maximum features to be considered for a split

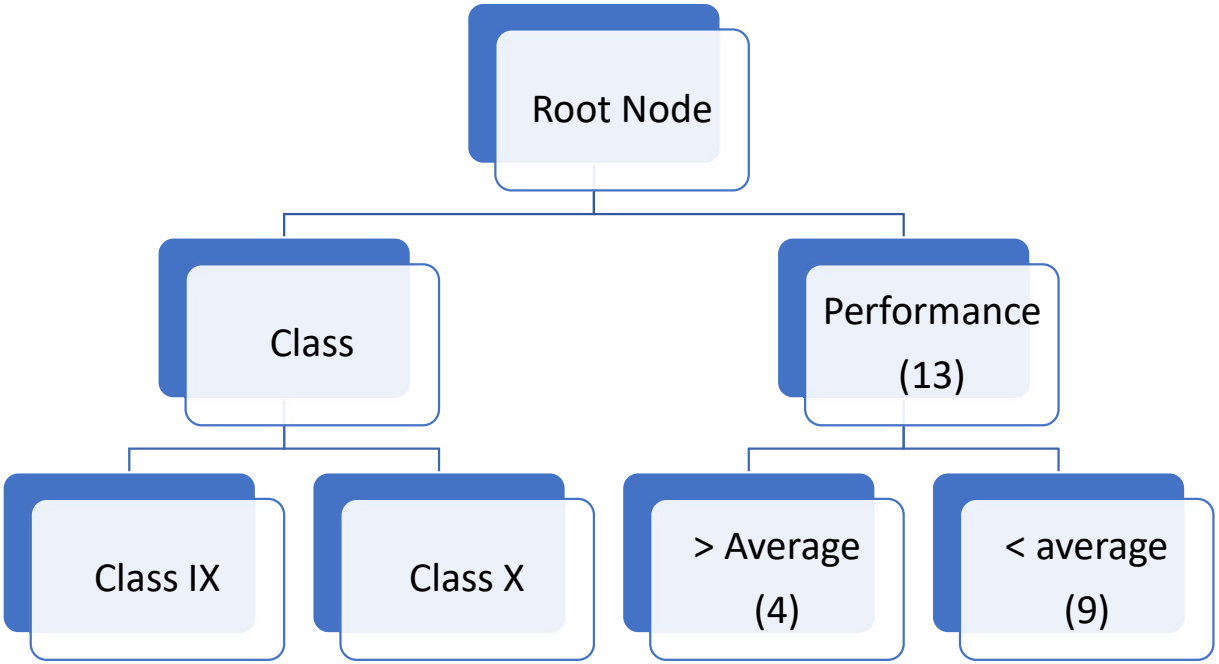
Minimum samples on a decision node to split

Lets say if the rule is that a node must have atleast 10 samples to go for further split



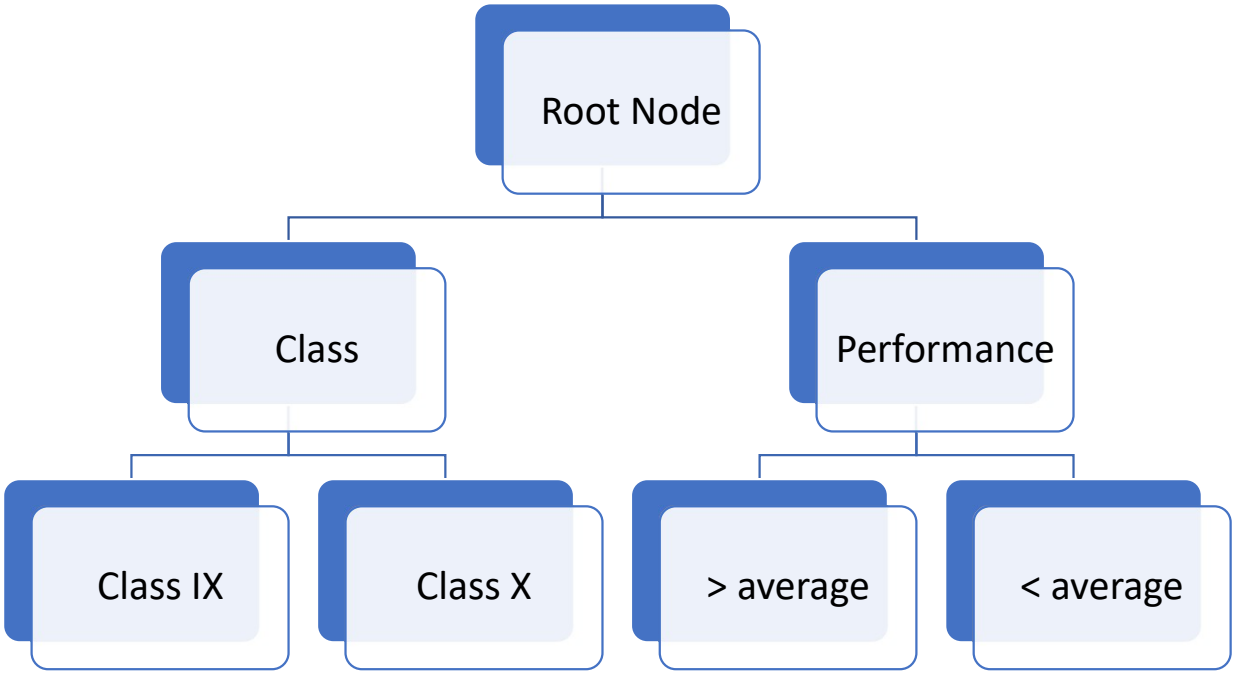
Minimum samples for a terminal node

- Suppose if the rule is that minimum samples in the terminal or leaf node must be 5

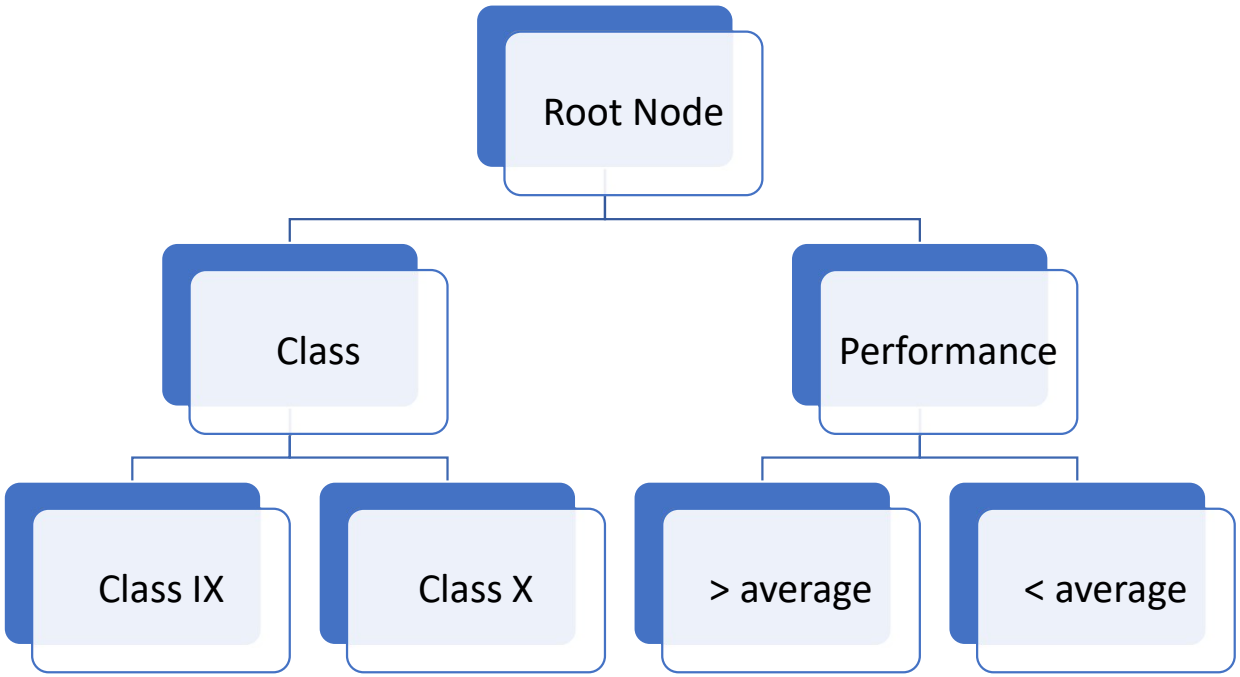


Maximum depth:

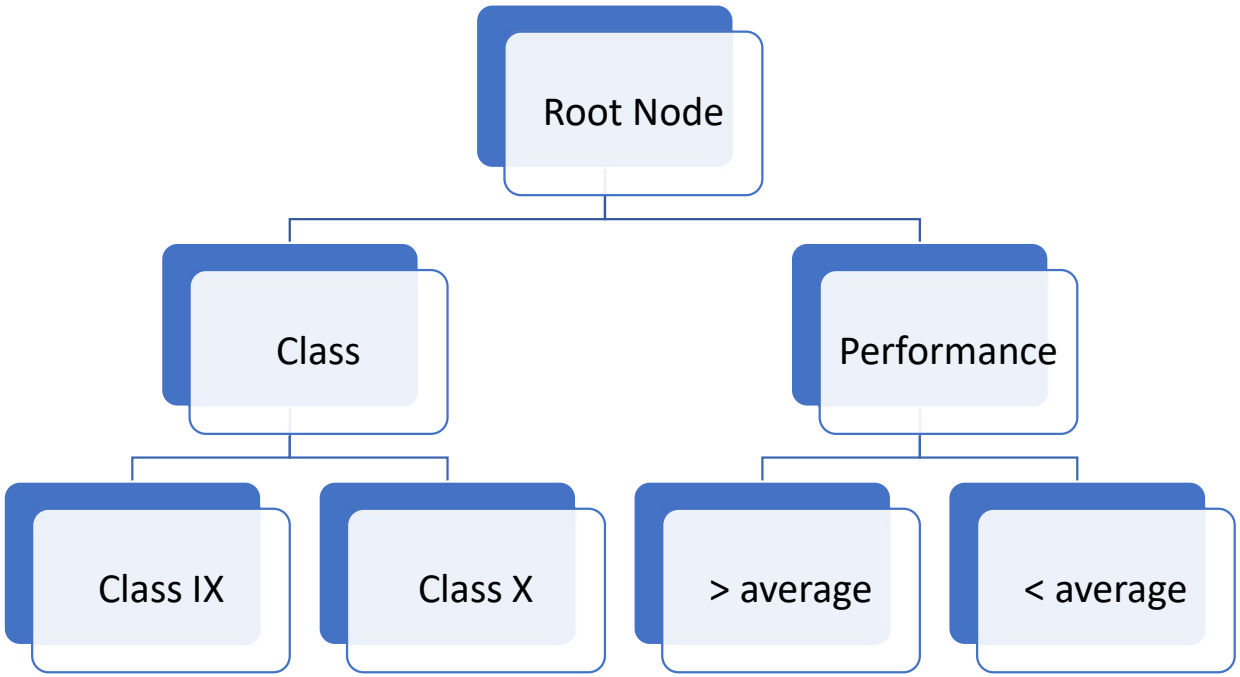
Depth is defined as the number of vertical splits the tree needs to do in order to give the final prediction. We can limit this depth by specifying the maximum depth acceptable for pruning purpose.

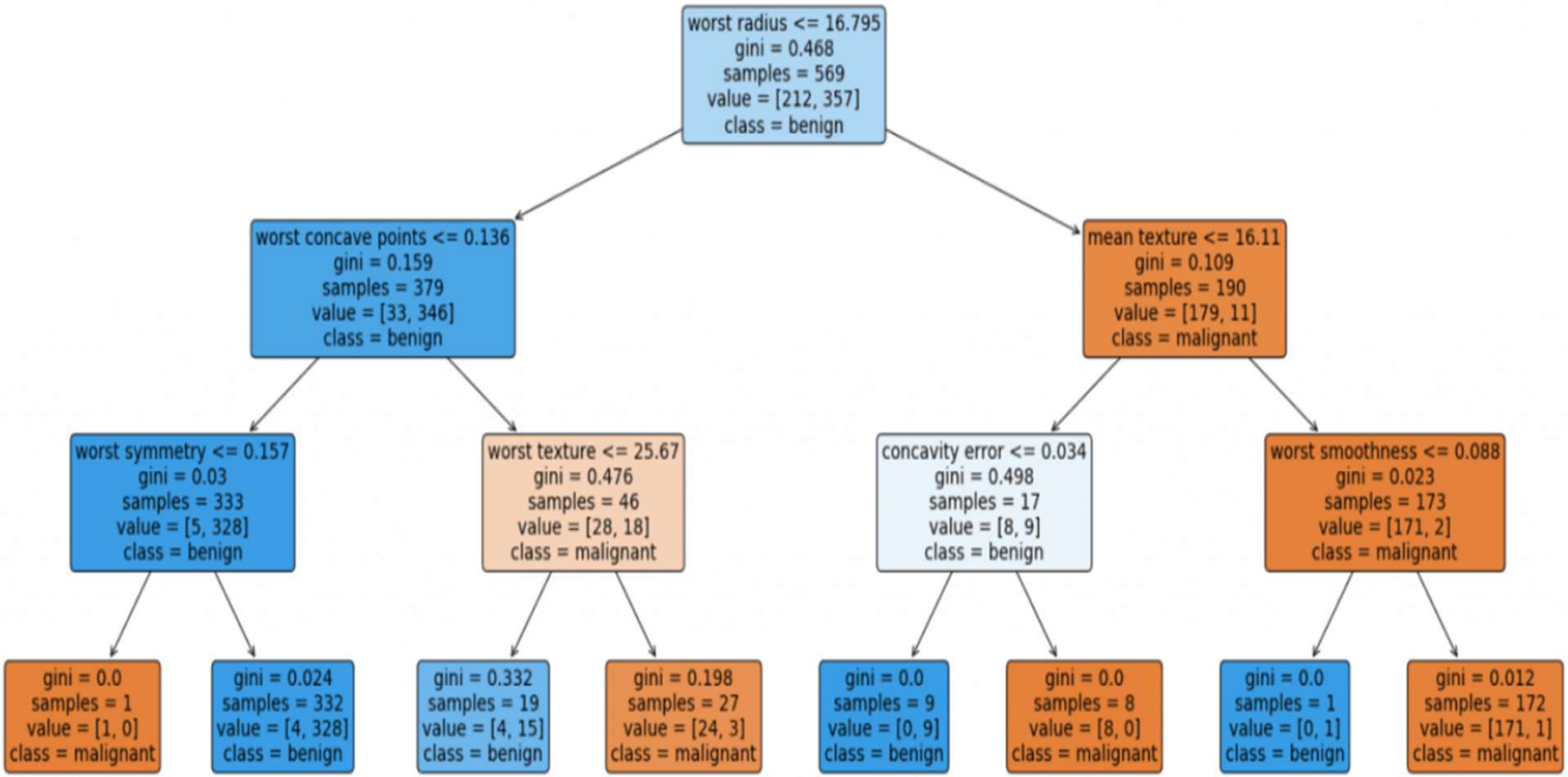


Maximum number of terminal nodes



Maximum features to be split: Generally when there are n features in a dataset, the maximum features to be split is set to sqrt(n) or log(n)




worst radius <= 16.795
gini = 0.468
samples = 569
value = [212, 357]
class = benign
worst concave points <= 0.136
gini = 0.159
samples = 379
value = [33, 346]
class = benign
mean texture <= 16.11
gini = 0.109
samples = 190
value = [179, 11]
class = malignant
worst symmetry <= 0.157
gini = 0.03
samples = 333
value = [5, 328]
class = benign
worst texture <= 25.67
gini = 0.476
samples = 46
value = [28, 18]
class = malignant
concavity error <= 0.034
gini = 0.498
samples = 17
value = [8, 9]
class = benign
worst smoothness <= 0.088
gini = 0.023
samples = 173
value = [171, 2]
class = malignant
gini = 0.0
samples = 1
value = [1, 0]
class = malignant
gini = 0.024
samples = 332
value = [4, 328]
class = benign
gini = 0.332
samples = 19
value = [4, 15]
class = benign
gini = 0.198
samples = 27
value = [24, 3]
class = malignant
gini = 0.0
samples = 9
value = [0, 9]
class = benign
gini = 0.0
samples = 8
value = [8, 0]
class = malignant
gini = 0.0
samples = 1
value = [0, 1]
class = benign
gini = 0.012
samples = 172
value = [171, 1]
class = malignant

Different splitting algorithms

- CART: Classification And Regression Tree. It uses GINI as splitting criteria
- ID3: It uses Entropy as splitting criteria

- 99% of the time CART will be used in all DT models.

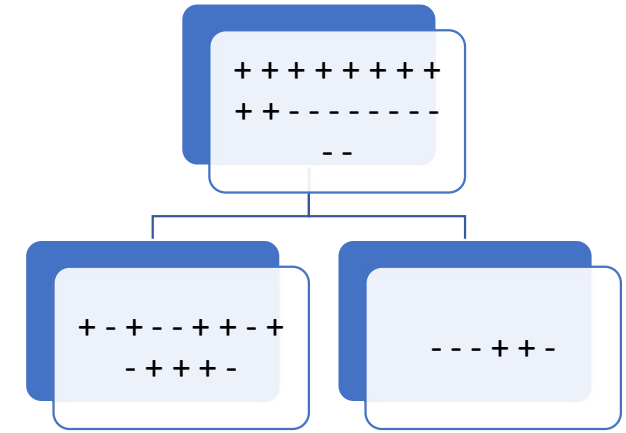
Four splitting methods are as follows:

- GINI
- ENTROPY

Example

Let the decision node have 20 students. + denotes those who play cricket and – denotes those who do not play cricket

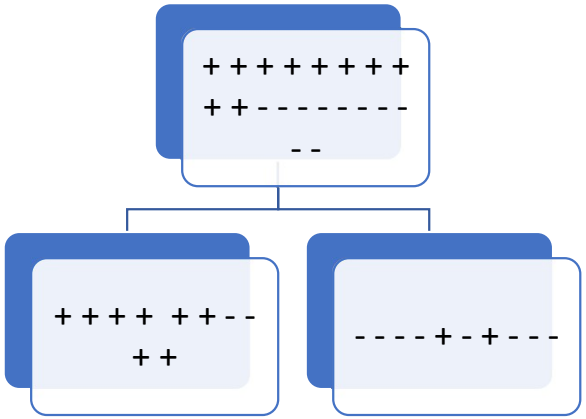
Split by performance



Above Average

Below Average

Split by class



Class IX

Class X

GINI

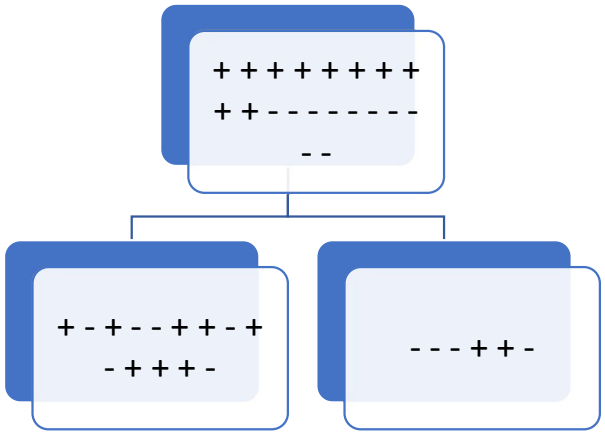
- GINI is a measure that talks about purity of a node.

$$GINI = P_1^2 + P_2^2 + P_3^2 + P_4^2 + ...+ P_n^2$$

$$GI = 1 - GINI$$

- Steps of calculation:
  1. Calculate GINI for each subnode
  2. Calculate GINI Impurity of each subnode
  3. Calculate the overall weight GINI impurity of the split.

GINI Calculation for split by performance



Above Average

- Total = 14
- Play cricket = 8
- Not play cricket = 6
- P(+) = 8/14 = 0.57
- P(-) = 6/14 = 0.43

Below Average

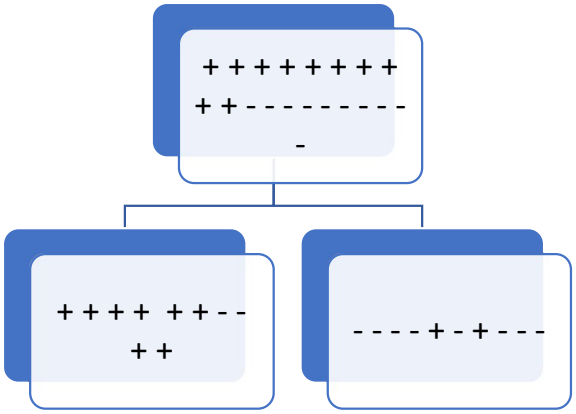
- Total = 6
- Play cricket = 2
- Not play cricket = 4
- P(+) = 2/6 = 0.33
- P(-) = 4/6 = 0.67

- ***<u>Step I: GINI calculation of each sub node</u>***
- GINI (AA) = (0.57)^2 + 0.43)^2 = 0.5098
- GINI (BA) = (0.33)^2 + (0.67)^2 = 0.5578

- ***<u>Step II: GINI Impurity calculation for each sub node</u>***
- GI(AA) = 1 – 0.5098 = 0.4902
- GI(BA) = 1 – 0.5578 = 0.4422

- ***<u>Step III: Weight GI of entire split</u>***
- WGI = (14/20)*0.4902 + (6/20) * 0.4422 = 0.4758

GINI Calculation for split by class



Class IX

- Total =
- Play cricket =
- Not play cricket =
- P(+) =
- P(-) =

Class X

- Total =
- Play cricket =
- Not play cricket =
- P(+) =
- P(-) =

- ***<u>Step I: GINI calculation of each sub node</u>***
- GINI (IX) =
- GINI (X) =

- ***<u>Step II: GINI Impurity calculation for each sub node</u>***
- GI(IX) =
- GI(X) =

- ***<u>Step III: Weight GI of entire split</u>***
- WGI = (W class IX) * GI(IX) + (W class X) * GI(X)
- WGI =