Decision Tree

Abdul Haseeb

BS(AI)-III

Outline

- What is Decision Tree?
- Terminologies related to Decision Tree
- Different Splitting criterion for Decision Tree
- Pros/Cons of Decision Tree
- Implementation of Decision Tree

What is Decision Tree

- Decision Tree is a supervised Learning Algorithm which uses tree like structure to classify data or make predictions

- **Characteristics:**
  - Tree Structure
  - Supervised Learning

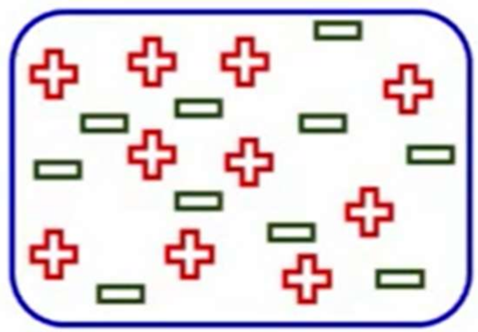
What is Decision Tree

- **Types:**
  - Classification Trees
  - Regression Trees

- **Applications:**
  - **Classification:**
    - Spam classification, Image Classification
  - **Regression:**
    - Predicting stock prices
  - **Feature Selection:**
    - Identifying relevant features

Decision Tree

- Height
- Performance in class
- Class



Total number of students = 20

Play cricket = 10

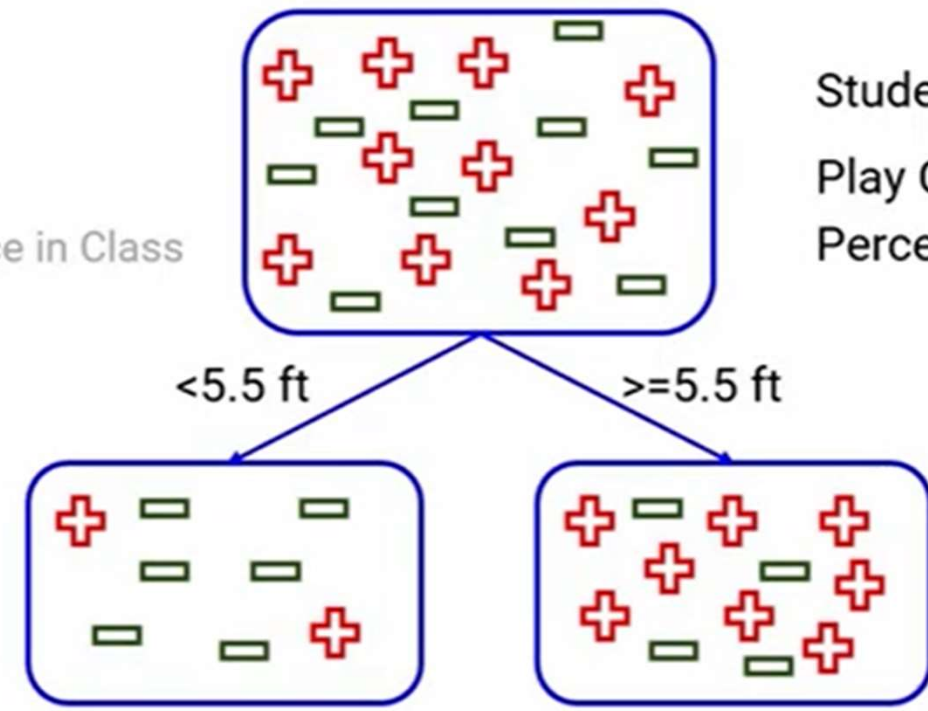
Do not play cricket = 10

Decision Tree

- **Teacher** wants to identify subgroups, that these subgroups are very much familiar with playing/not playing the cricket with the help of given attributes

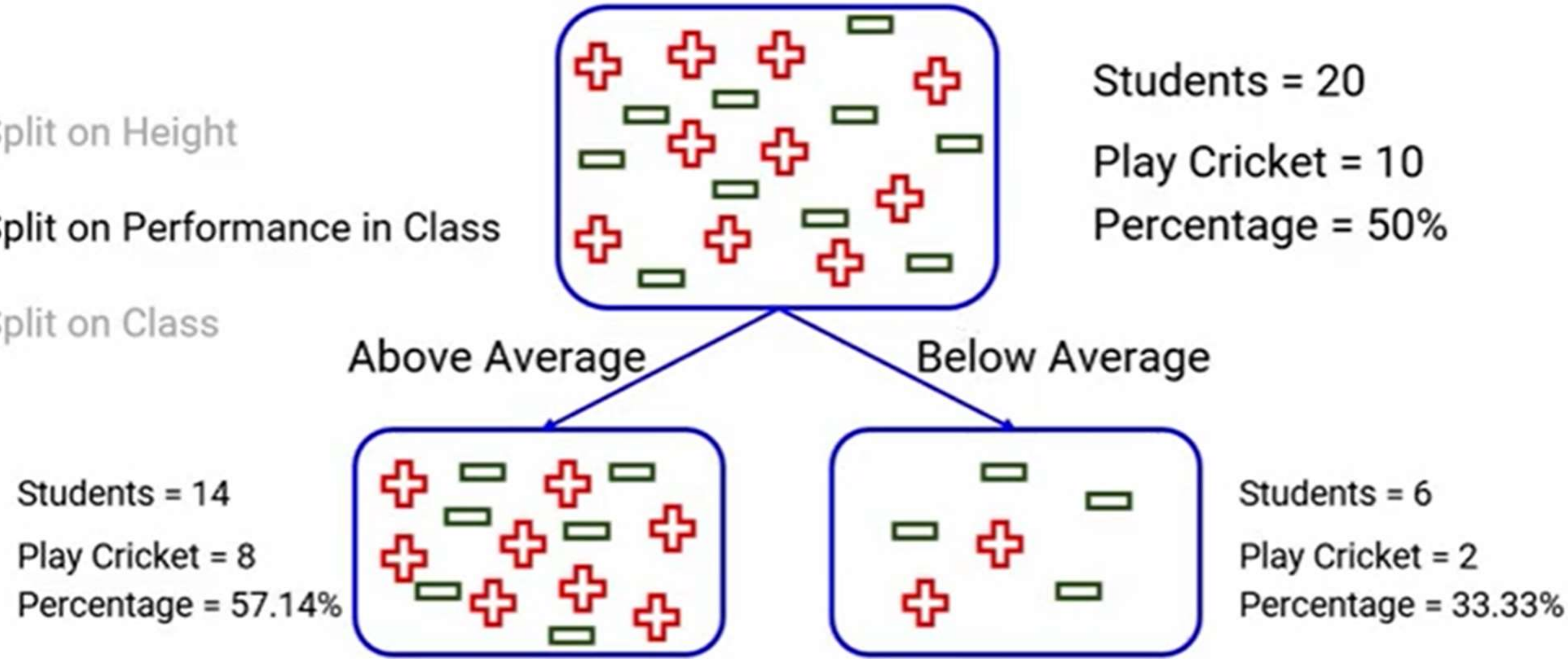
What is Decision Tree

- Split on Height
- Split on Performance in Class
- Split on Class



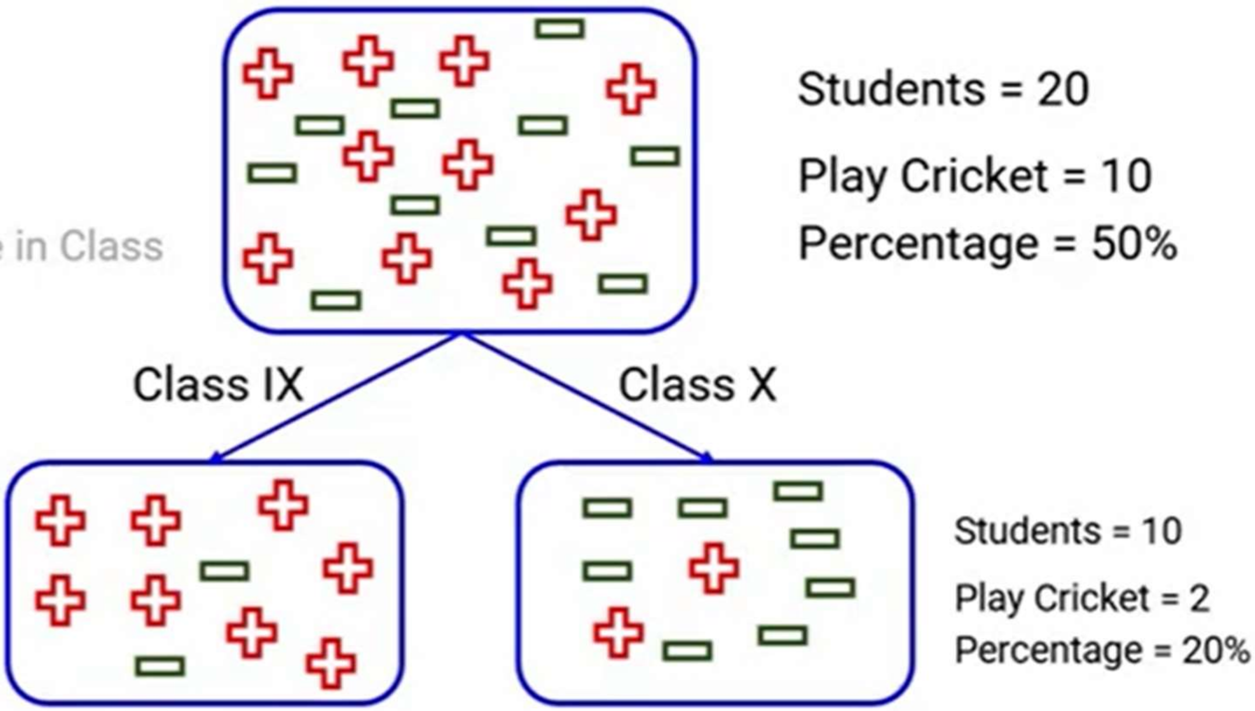
What is Decision Tree

- Split on Height
- Split on Performance in Class
- Split on Class

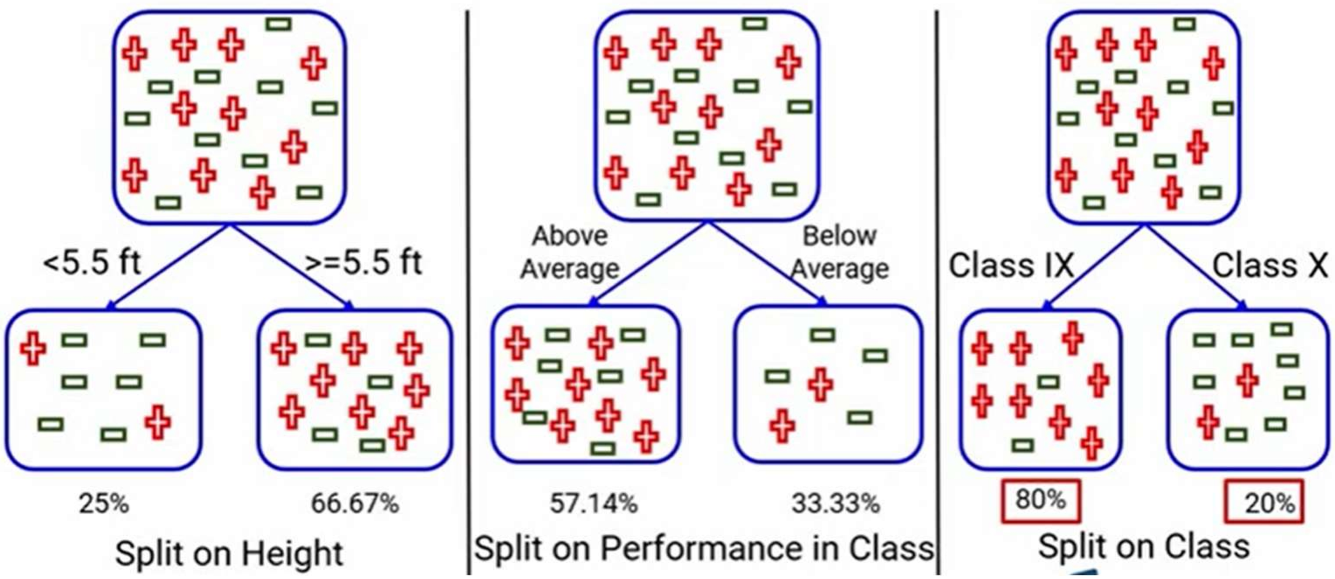


What is Decision Tree

- Split on Height
- Split on Performance in Class
- Split on Class



What is Decision Tree

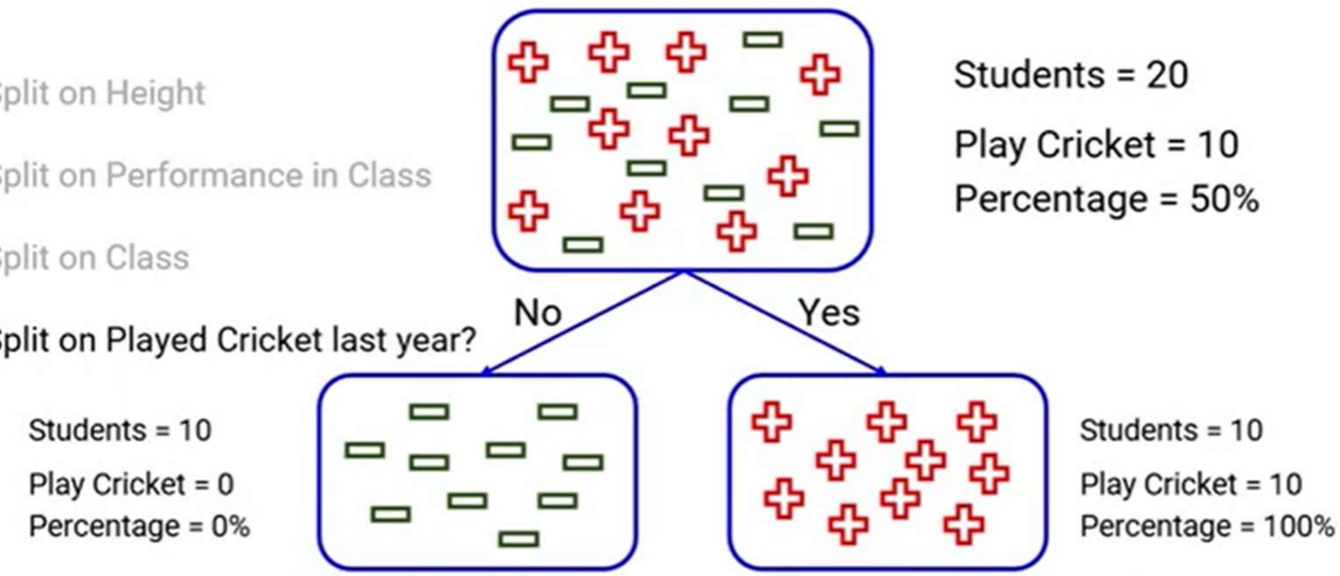


Look at the split on class:

It looks the best split as it segregates the most of students

Purity in Decision Tree

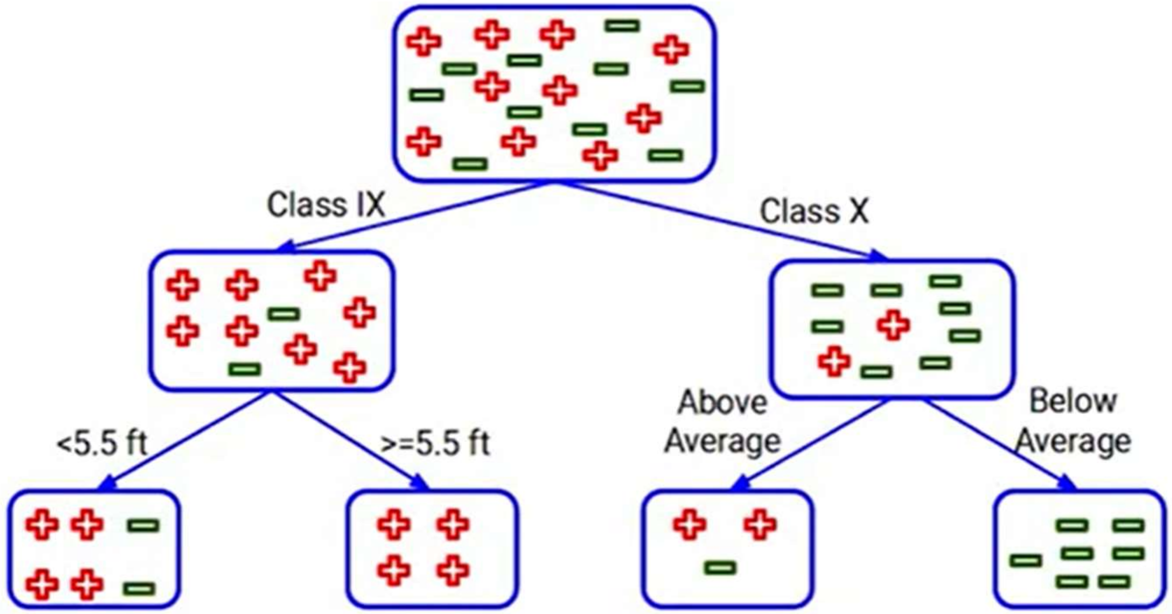
- Split on Height
- Split on Performance in Class
- Split on Class
- Split on Played Cricket last year?



Objective of Decision Tree is to produce pure nodes

In practical scenario we rarely will have such features which can produce best split like this

Purity in Decision Tree



We will have multiple splits and multiple decisions in a decision tree.

Purity in Decision Tree

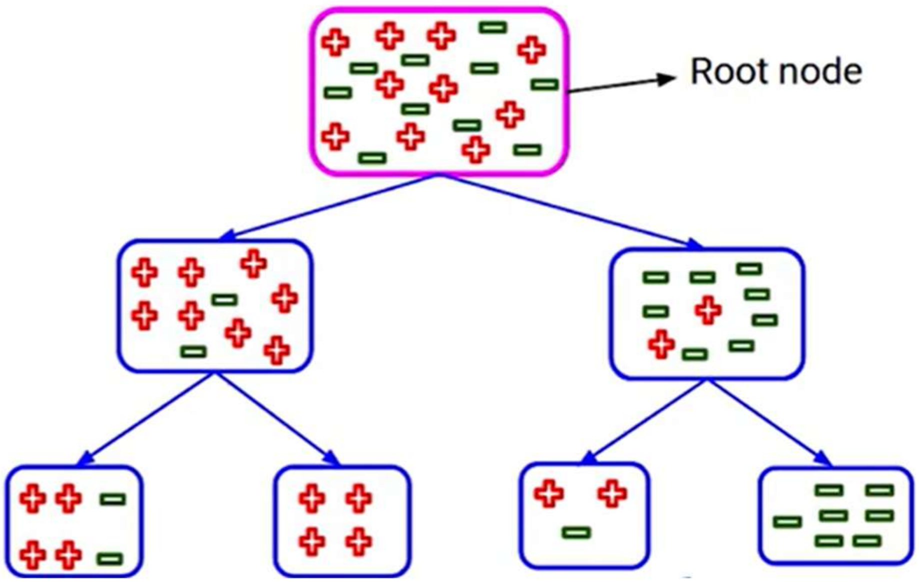
- But wait…
- What should be root node for split?
- What should be the sequence of split?
- How to select the node for split?

Purity in Decision Tree

- There are techniques to decide purity of nodes: the node which is purest among the others will be taken for the splitting

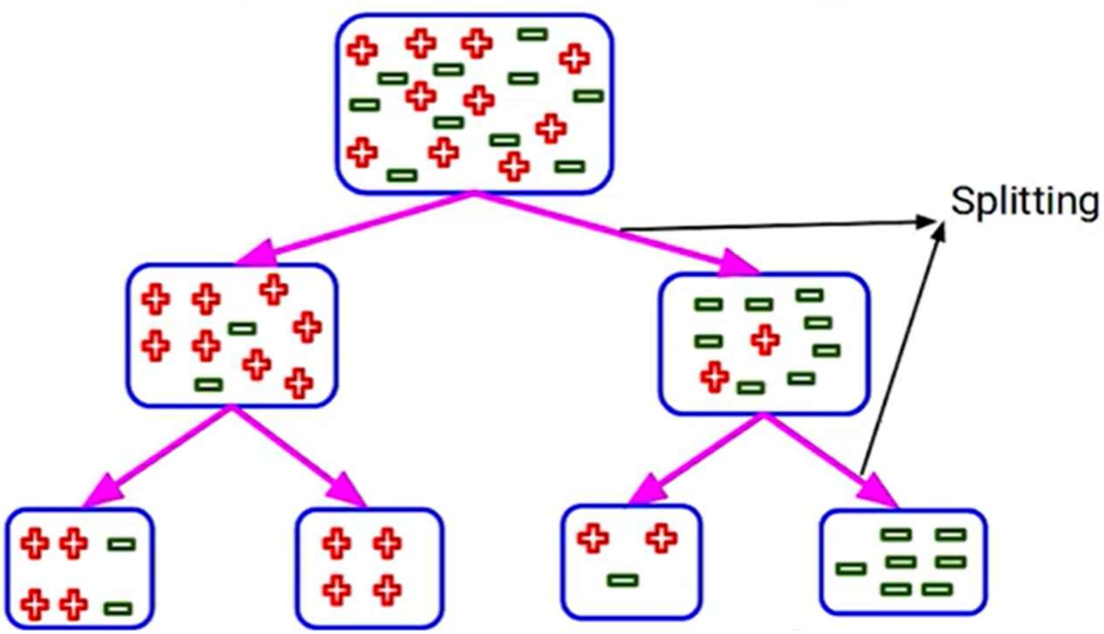
Terminologies related to Decision Tree

- Root Node



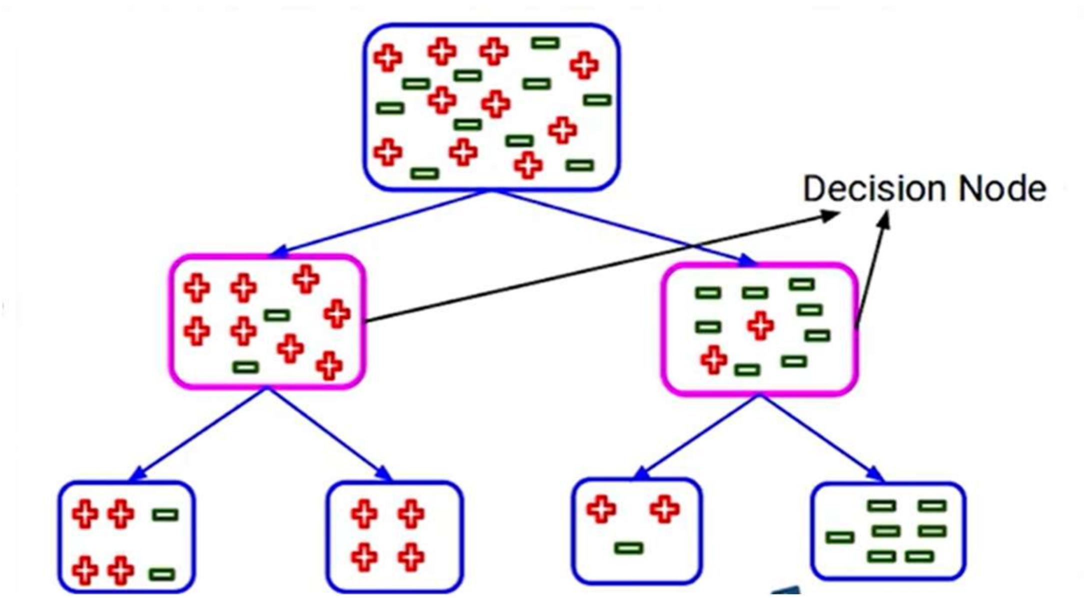
Root node describes the entire population

Terminologies related to Decision Tree



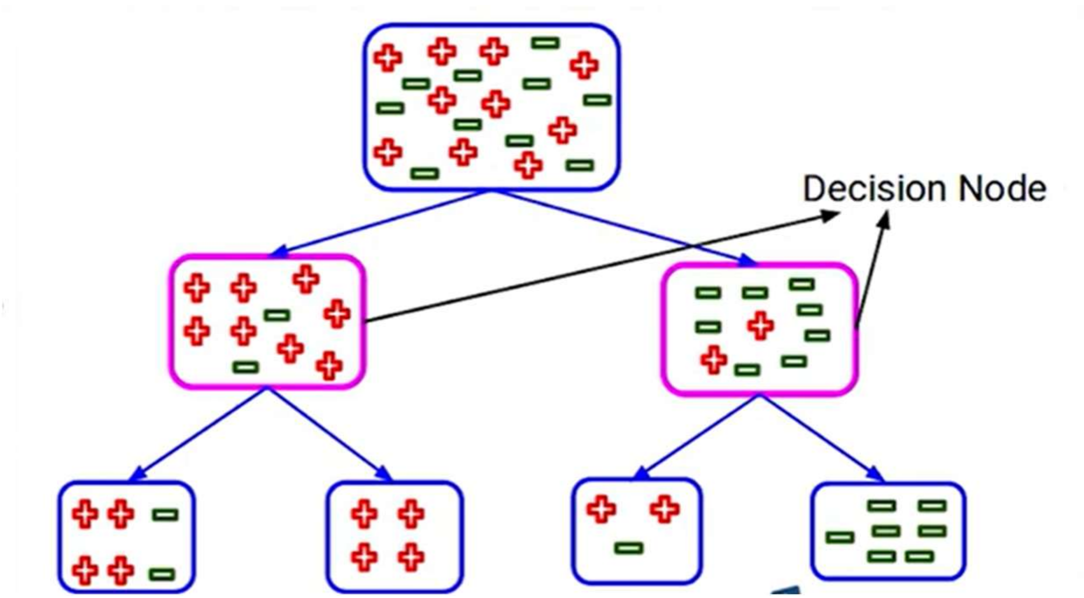
- Splitting is dividing a node into further sub nodes

Terminologies related to Decision Tree



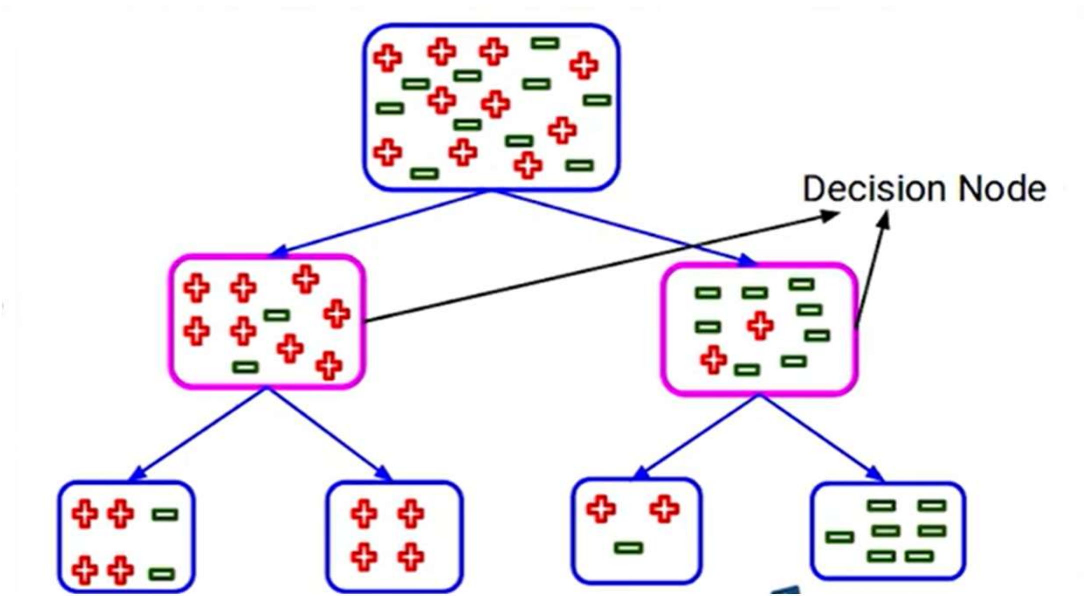
- Decision Nodes are those on which a split is performed

Terminologies related to Decision Tree



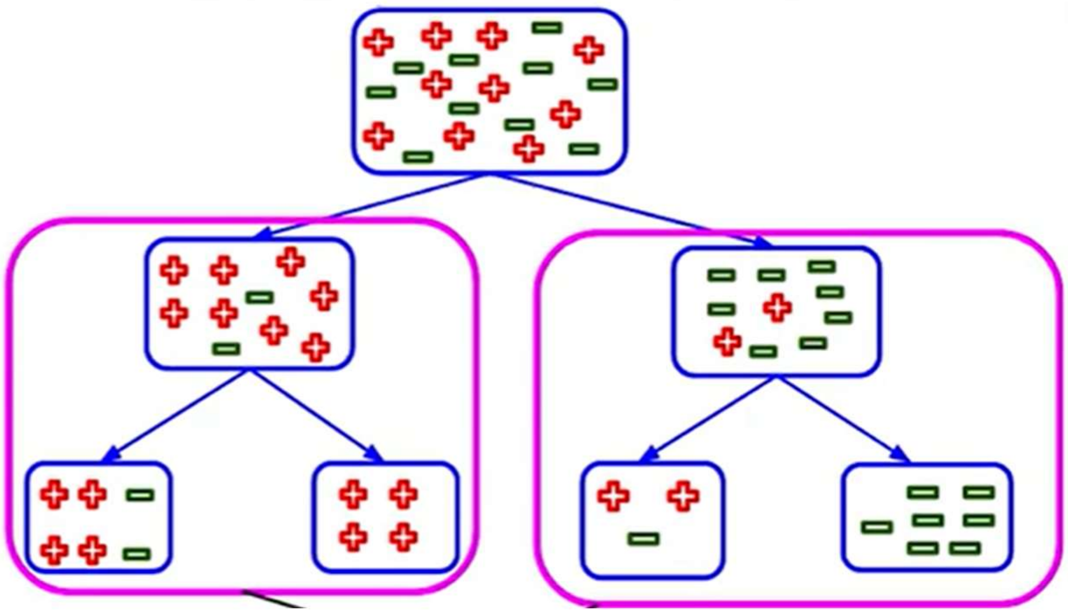
- Leaf nodes don't split further

Terminologies related to Decision Tree



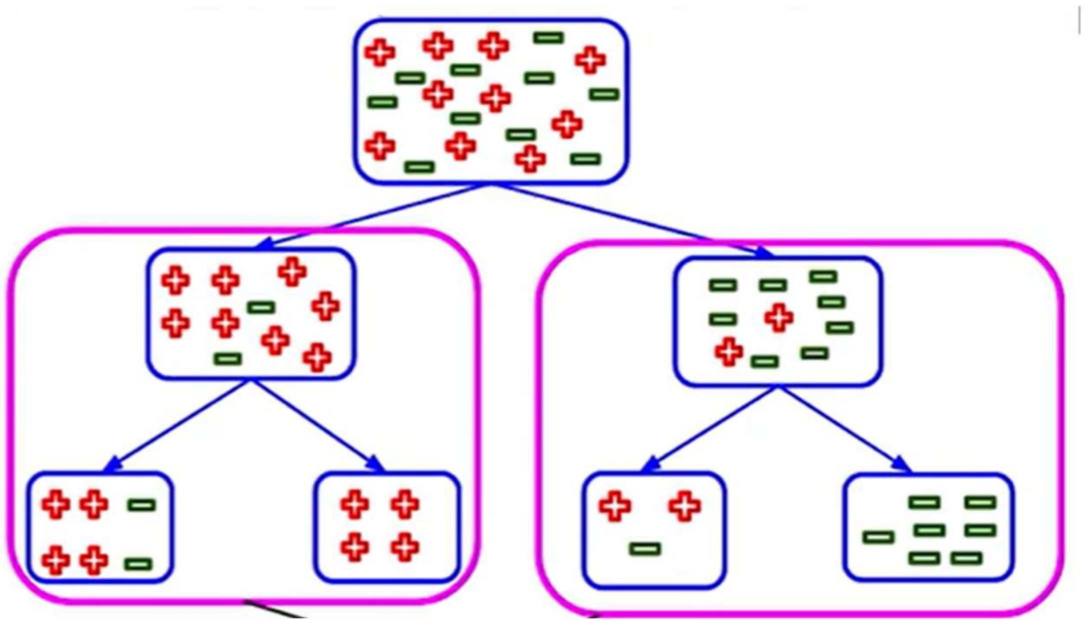
- Leaf nodes don’t split further

Terminologies related to Decision Tree



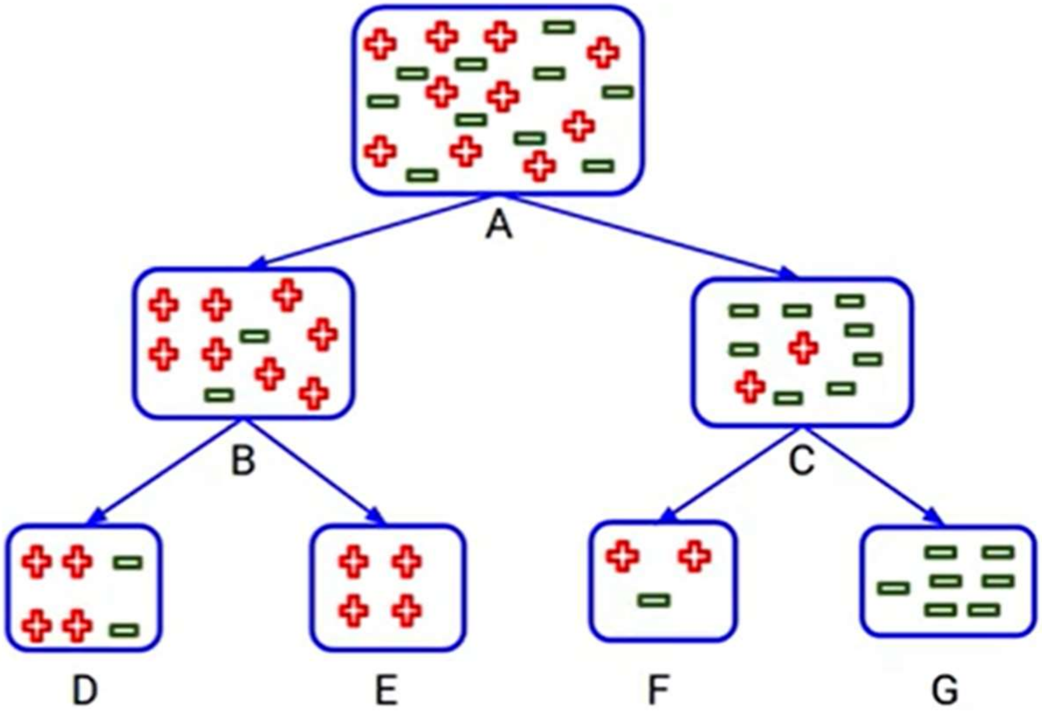
- Subtree is a subset/Part of original tree

Terminologies related to Decision Tree



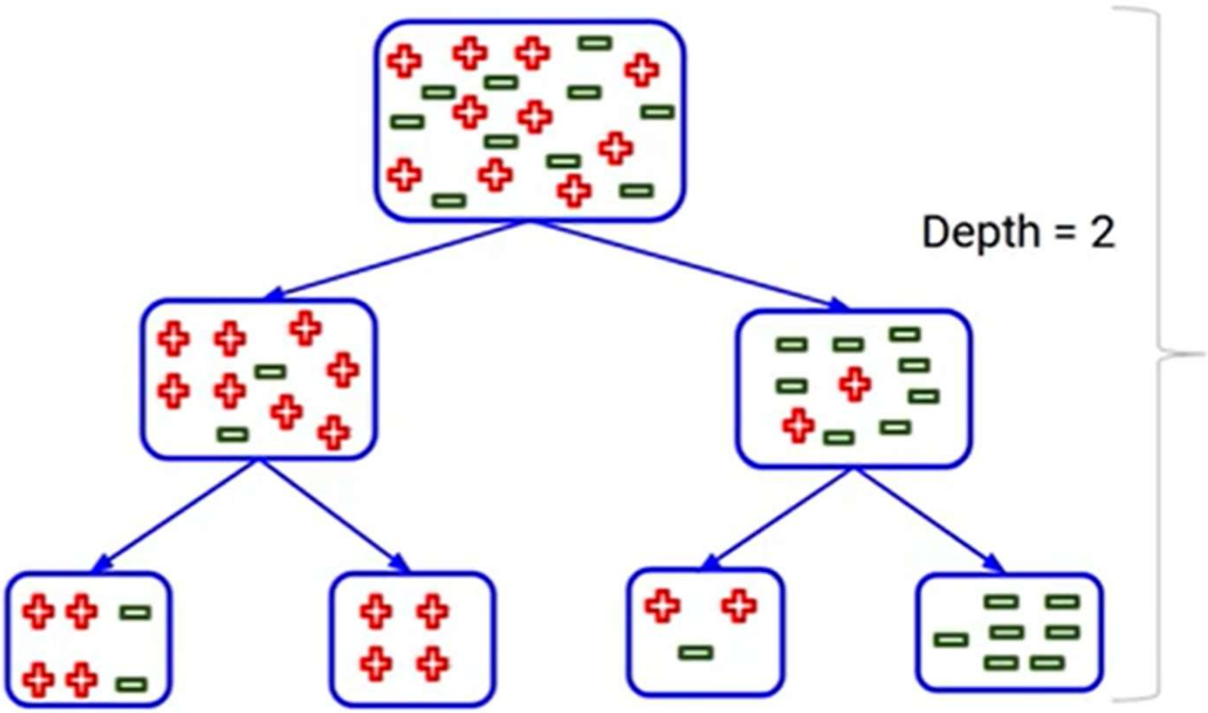
- Subtree is a subset/Part of original tree

Terminologies related to Decision Tree



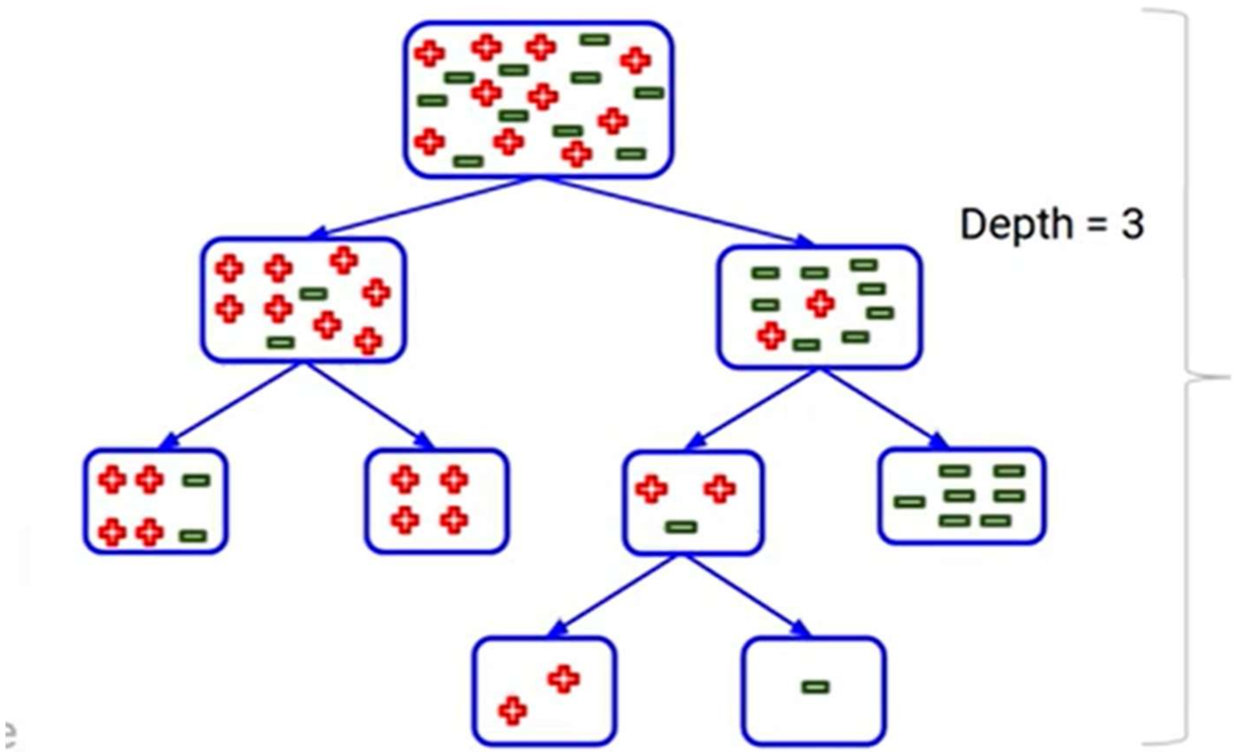
- Parent/Child Nodes

Terminologies related to Decision Tree



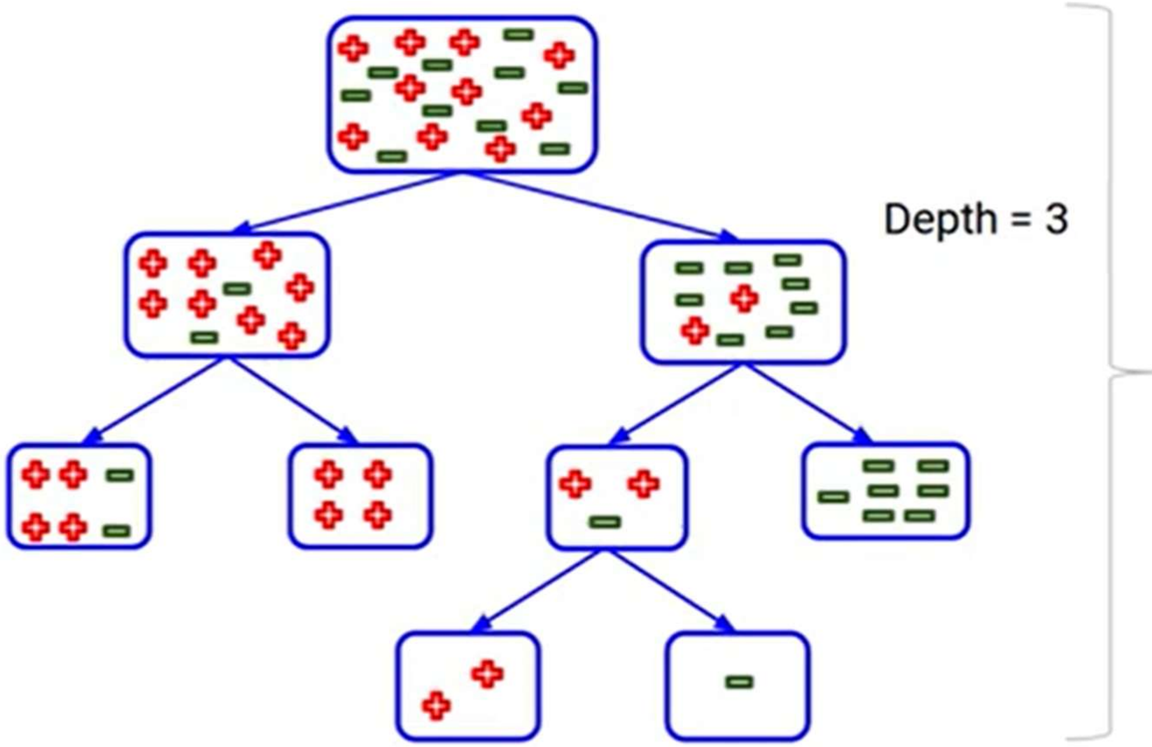
- Depth is the longest path from root to leaf

Terminologies related to Decision Tree



- How many leaf nodes in this tree??

How to select the best split point in Decision Tree?



- How many leaf nodes in this tree??

How to select the best split point in Decision Tree?

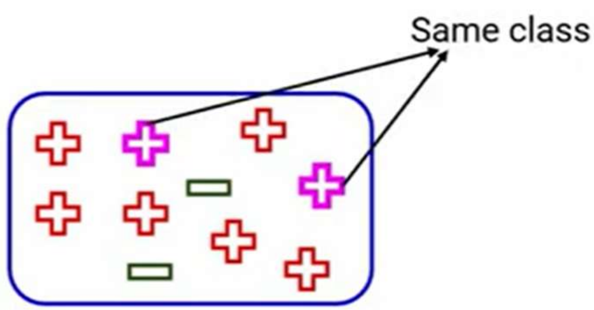
- Decision tree splits the nodes on all available variable
- Select the split which results in most homogenous sub-nodes
- Decision Tree Algorithms measure node Impurity, and two most common techniques for measuring node impurity are:
  - Gini
  - Entropy

How to select the best split point in Decision Tree?

- Gini Impurity:
  - 1-Gini

How to select the best split point in Decision Tree?

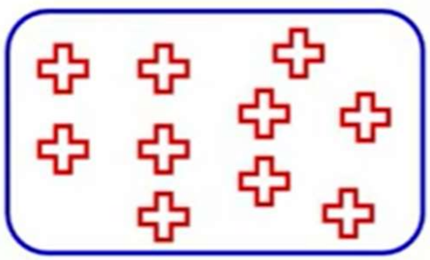
- Gini Impurity states:



If we select two items from a population at random, they must be of same class

How to select the best split point in Decision Tree?

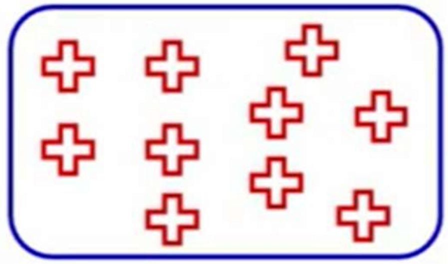
- Gini Impurity:



Probability that randomly picked points belong to same class?

How to select the best split point in Decision Tree?

- Gini Impurity:



Probability = 1

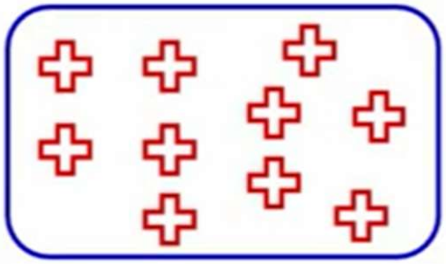
Probability is one, as all the samples belong to same class

Node is pure...

Gini ranges from 0-1, Highest the Gini: Highest the Node Purity

How to select the best split point in Decision Tree?

- Gini Impurity:



Probability = 1

Probability is one, as all the samples belong to same class

Node is pure…

Gini ranges from 0-1, Highest the Gini: Highest the Node Purity

Properties of Gini Impurity

- Node split is decided based on the gini impurity
- Lower the gini impurity, higher the homogeneity of the nodes
- Works only with categorical data
- Only performs binary splits

Steps to calculate Gini Impurity for a split

- Calculate gini impurity for subnode

- Gini=Sum of square probability of each class
  - *Gini*=$(p1^2+p2^2+p3....+pn^2)$

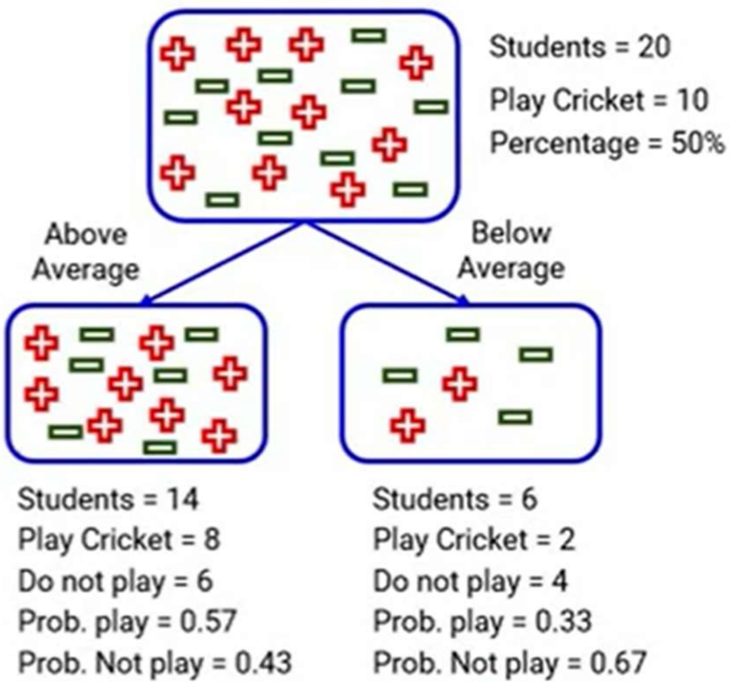
- To calculate gini impurity of a split, take weighted gini impurity of both sub-nodes of that split

Steps to calculate Gini Impurity for a split

Split on Performance in Class

- Gini Impurity: sub-node Above Average:
  1 - [(0.57)*(0.57) + (0.43)*(0.43)] = 0.49

- Gini Impurity: sub-node Below Average:
  1 - [(0.33)*(0.33) + (0.67)*(0.67)] = 0.44



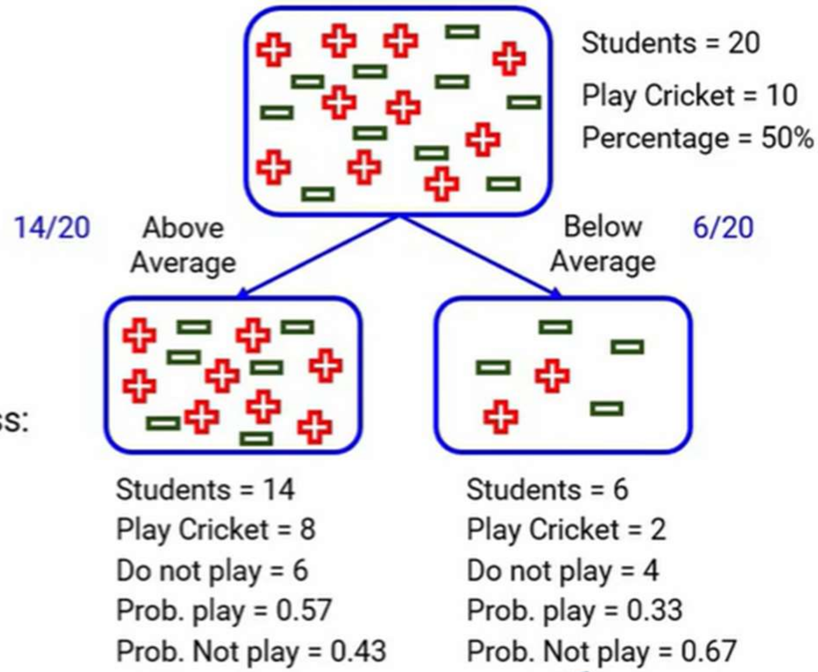
Steps to calculate Gini Impurity for a split

Split on Performance in Class

- Gini Impurity: sub-node Above Average:
  1 - [(0.57)*(0.57) + (0.43)*(0.43)] = 0.49

- Gini Impurity: sub-node Below Average:
  1 - [(0.33)*(0.33) + (0.67)*(0.67)] = 0.44

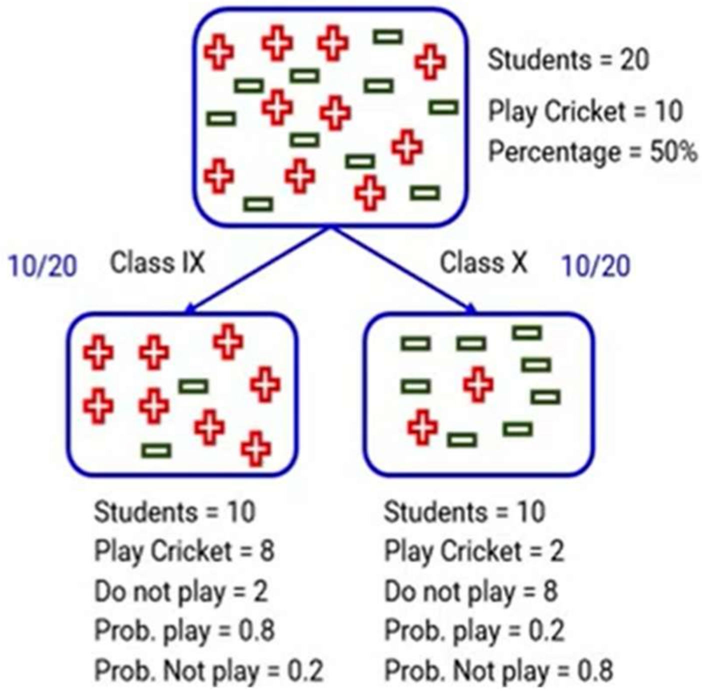
- Weighted Gini Impurity: Performance in Class:
  (14/20)*0.49 + (6/20)*0.44 = 0.475



Steps to calculate Gini Impurity for a split

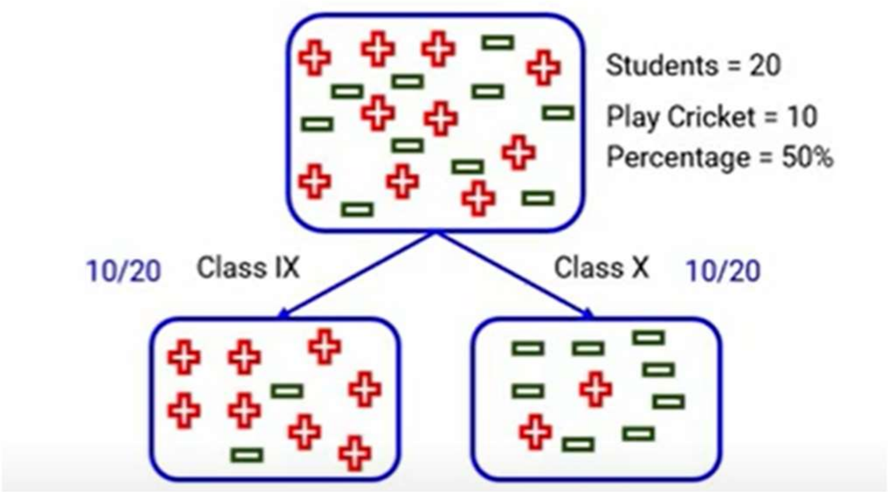
Split on Class

- Gini Impurity: sub-node Class IX:
  1 - [(0.8)*(0.8) + (0.2)*(0.2)] = 0.32
- Gini Impurity: sub-node Class X:
  1 - [(0.2)*(0.2) + (0.8)*(0.8)] = 0.32
- Weighted Gini Impurity: Class:
  (10/20)*0.32 + (10/20)*0.32 = 0.32



Weight of the node*Gini Impurity of the node

Steps to calculate Gini Impurity for a split



Calculate the Gini Impurity and Weighted Gini Impurity

Steps to calculate Gini Impurity for a split

- Gini Impurity: sub-node Class IX:
  1 - [(0.8)*(0.8) + (0.2)*(0.2)] = 0.32

- Gini Impurity: sub-node Class X:
  1 - [(0.2)*(0.2) + (0.8)*(0.8)] = 0.32

- Weighted Gini Impurity: Class:
  (10/20)*0.32 + (10/20)*0.32 = 0.32

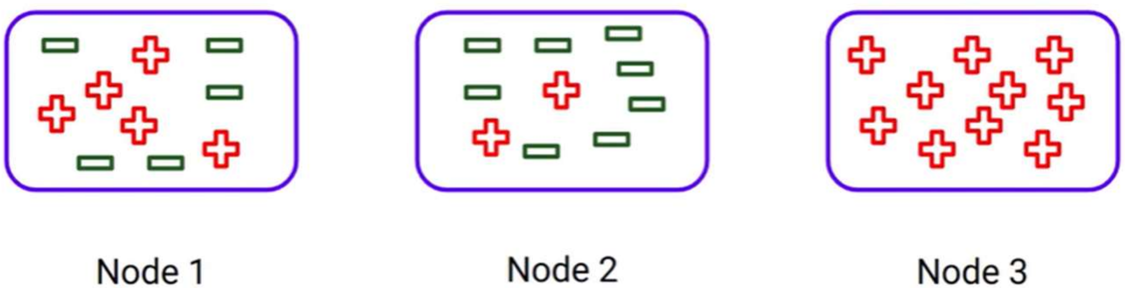
Steps to calculate Gini Impurity for a split

Split	Weighted Gini Impurity
Performance in Class	0.475
Class	0.32

Node Producing Minimum Weighted Gini Impurity will be selected as the Split

Another Algorithm for Deciding the Best Split

Information Gain:

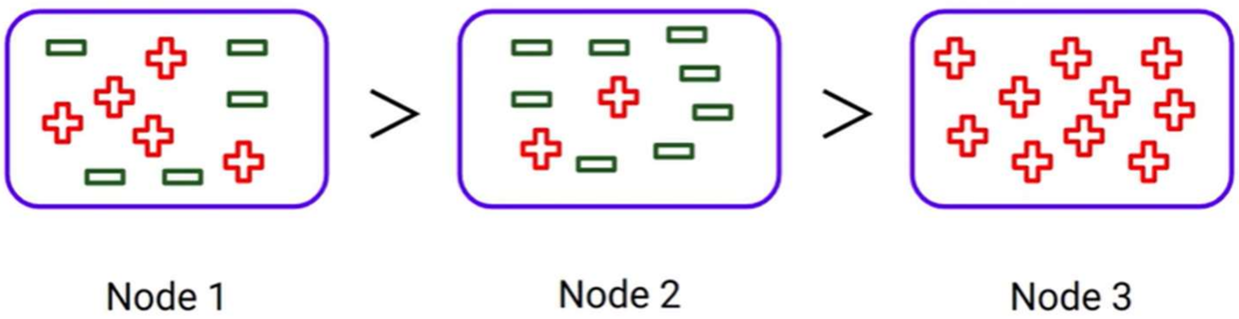


Which Node will require more explanation?

Which is the purest of the Nodes?

More Impure Nodes Require More Information

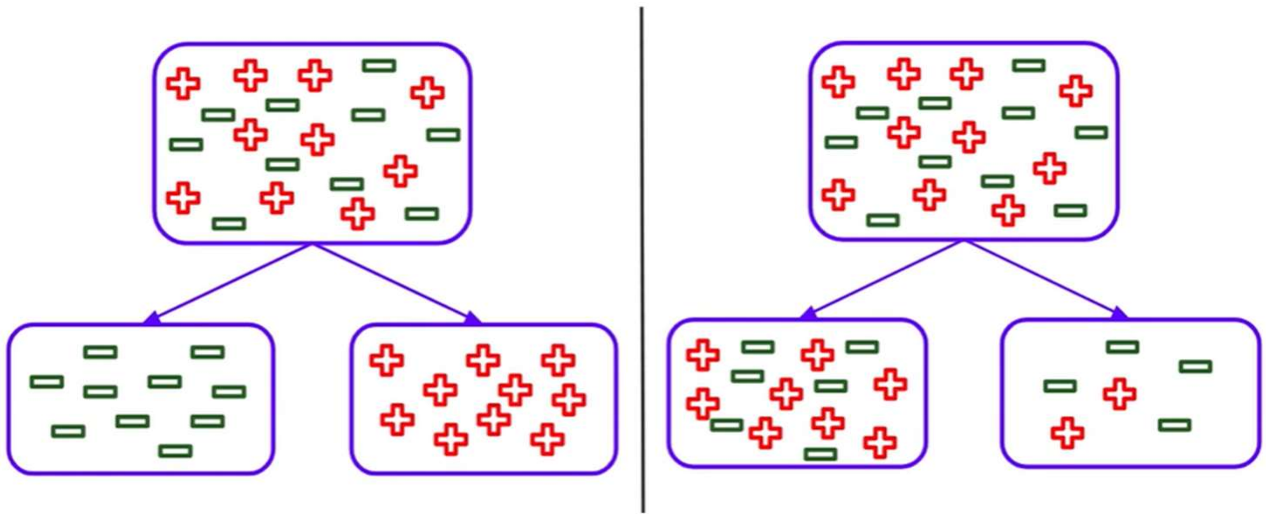
Information Gain:



Information required to describe the node

More Impure Nodes Require More Information

Information Gain:



What can you infer from this?

More Impure Nodes Require More Information

Information Gain:

- The split on the right is giving less information gain
- So, we can easily say: "***Higher the Information gain Higher the Homogeneity and lesser the impurity"***

More Impure Nodes Require More Information

Information Gain:

- The split on the right is giving less information gain
- So, we can easily say: "***Higher the Information gain Higher the Homogeneity and lesser the impurity***"

Formula for Information Gain

Information Gain = 1 - Entropy

Entropy

Entropy

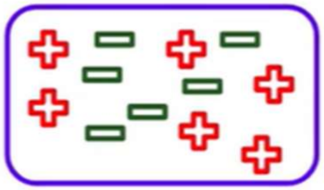
$$- p_1*\log_2 p_1 - p_2*\log_2 p_2 - p_3*\log_2 p_3 - \dots - p_n*\log_2 p_n$$

p refers to percentage of each class in the Node

Entropy

Entropy

$$- p_1 * \log_2 p_1 - p_2 * \log_2 p_2 - p_3 * \log_2 p_3 - \ldots\ldots - p_n * \log_2 p_n$$

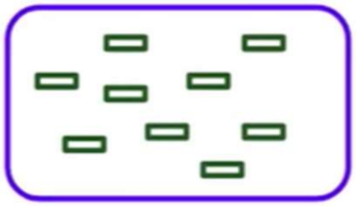


% Play = 0.50
% Not play = 0.50

Entropy = - (0.5) * log2(0.5) - (0.5) * log2(0.5)

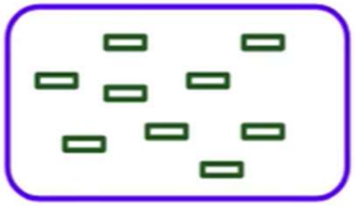
= 1

Calculate Entropy



% Play = 0
% Not play = 1

Solution

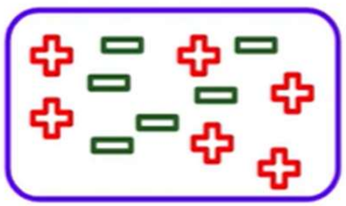


% Play = 0
% Not play = 1

Entropy = - (0) * log2(0) - (1) * log2(1)

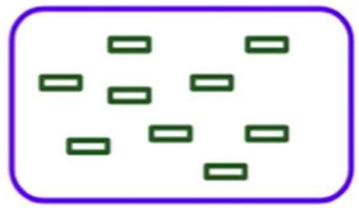
= 0

Entropy



% Play = 0.50
% Not play = 0.50

Entropy = 1



% Play = 0
% Not play = 1

Entropy = 0

Lower the Entropy means?

Higher the Entropy means?

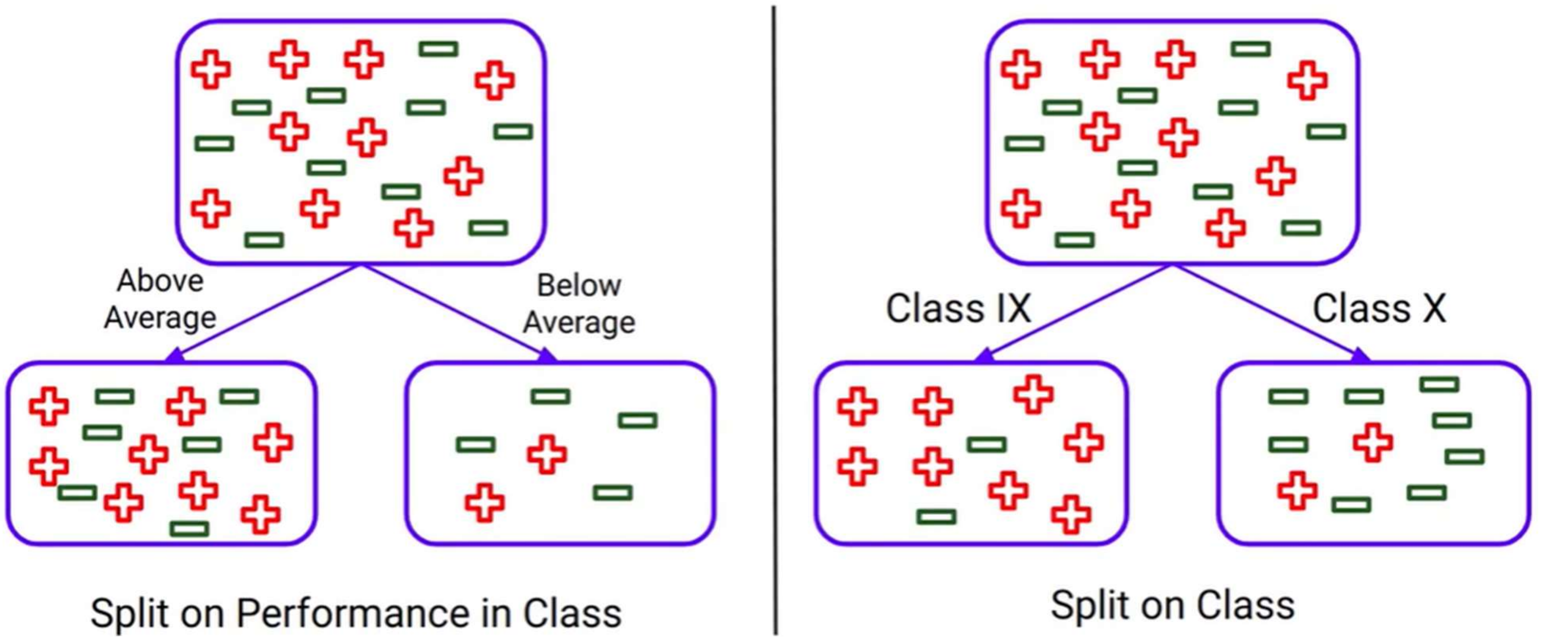
Properties of Entropy

- Works only with categorical Targets
- Lesser the entropy, higher the homogeneity of Nodes

Steps to Calculate Entropy of Nodes

- Calculate the entropy of the parent node
- Calculate the entropy of each child node
- Calculate the weighted average entropy of the split
- If weighted entropy of child node is greater than parent node, then we will ignore that node as it is returning more impure node than the parent

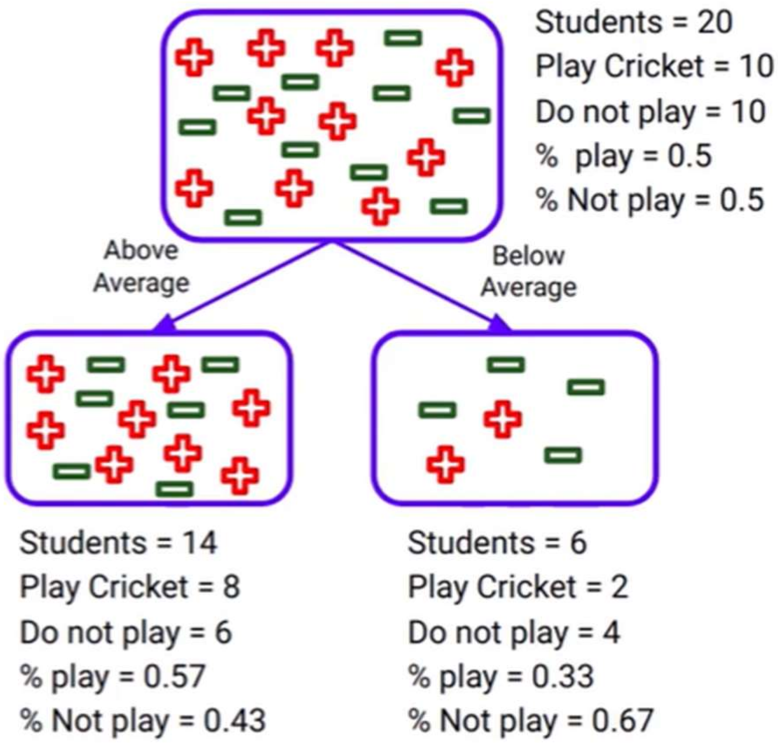
Steps to Calculate Entropy of Nodes



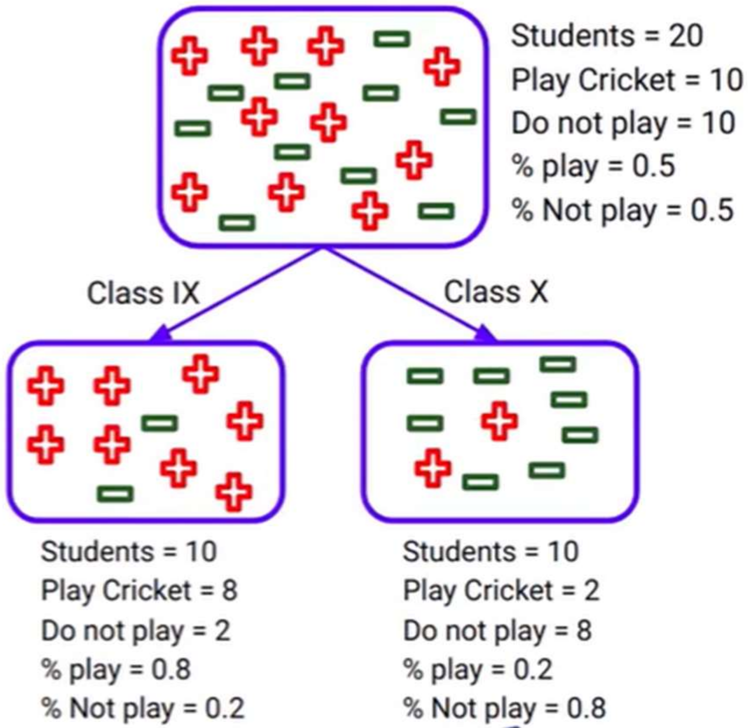
Steps to Calculate Entropy of Nodes

Split on Performance in Class

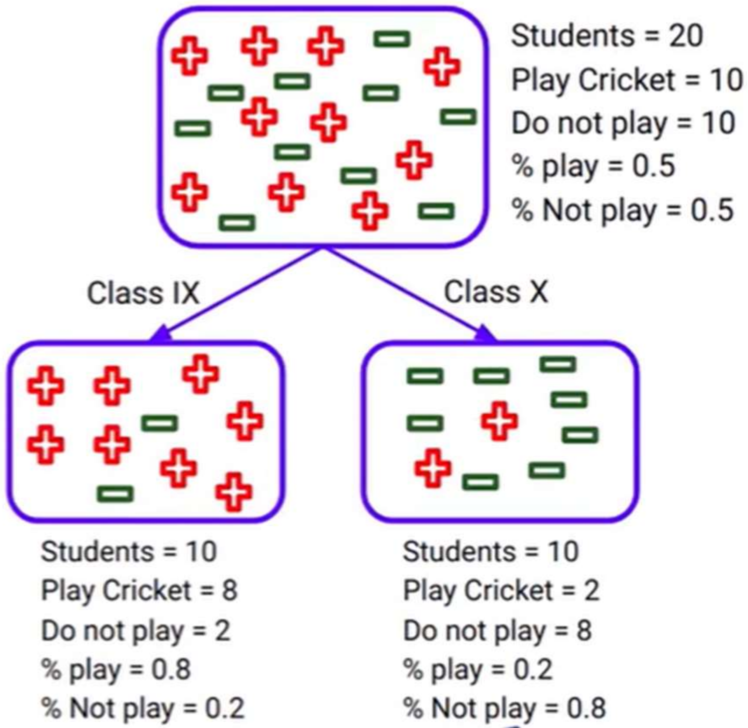
- Entropy for Parent node:
  $-(0.5)*\log_2(0.5) - (0.5)*\log_2(0.5) = 1$
- Entropy for sub-node Above Average:
  $-(0.57)*\log_2(0.57) - (0.43)*\log_2(0.43) = 0.98$
- Entropy for sub-node Below Average:
  $-(0.33)*\log_2(0.33) - (0.67)*\log_2(0.67) = 0.91$
- Weighted Entropy: Performance in Class:
  $(14/20)*0.98 + (6/20)*0.91 = 0.959$



Steps to Calculate Entropy of Nodes



Steps to Calculate Entropy of Nodes



Steps to Calculate Entropy of Nodes

Split on Class

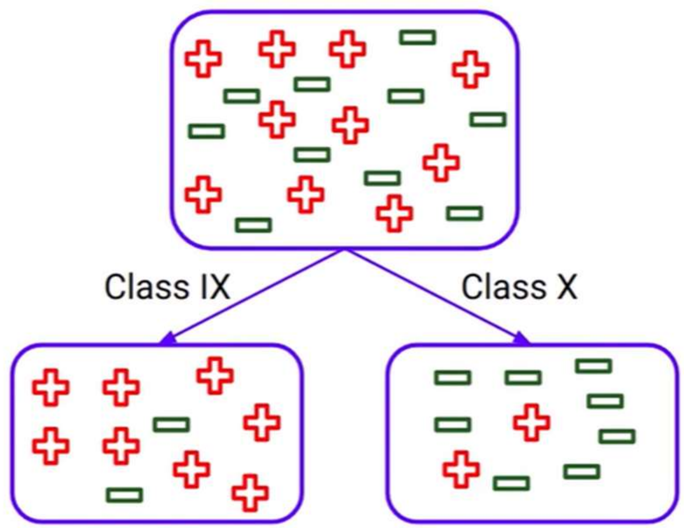
- Entropy for Parent node:
  $-(0.5)*\log_2(0,5) -(0.5)*\log_2(0.5) = 1$
- Entropy for sub-node Class IX:
  $-(0.8)*\log_2(0.8) -(0.2)*\log_2(0.2) = 0.722$
- Entropy for sub-node Class X:
  $-(0.2)*\log_2(0.2) -(0.8)*\log_2(0.8) = 0.722$
- Weighted Entropy: Class:
  $(10/20)*0.722 + (10/20)*0.722 = 0.722$

Steps to Calculate Entropy of Nodes

Split	Entropy	Information Gain
Performance in Class	0.959	0.041
Class	0.722	0.278

Higher Information Gain is Good or Lower Information Gain is Good?

Steps to Calculate Entropy of Nodes



Continuous Values!!

- So far, we dealt with Categorical Values….

- What about continuous data?

Reduction in Variance

- **Formula:**

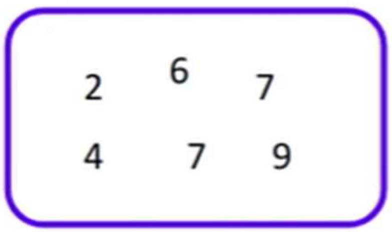
$$\text{Variance} = \Sigma\,[(X - \mu)^2] / n$$

Reduction in Variance

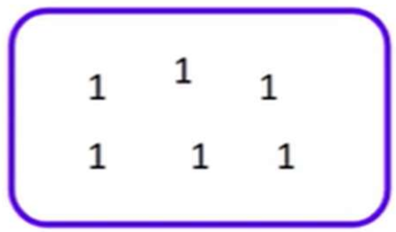
- **Formula:**

$$\text{Variance} = \Sigma \left[(X - \mu)^2\right] / n$$

Reduction in Variance



Variance ~ 6



Variance = 0

Lower Value of Variance is Good or Higher?

Properties of Variance

- Used when Target variable is Continuous

- Split with lower variance is selected

Steps to Calculate Variance

- Calculate the variance of each child node
- Variance = $\Sigma [(X - \mu)^2] / n$
- Calculate the variance of each split as weighted average variance of each child node

Steps to Calculate Variance

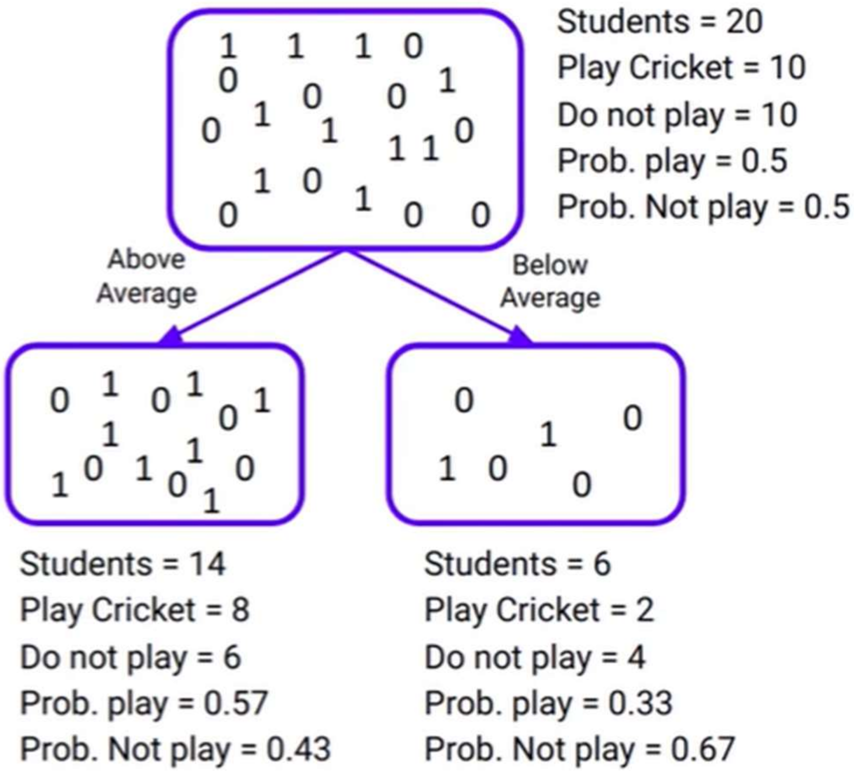
- Calculate the variance of each child node
- Variance = $\Sigma [(X - \mu)^2] / n$
- Calculate the variance of each split as weighted average variance of each child node

Steps to Calculate Variance

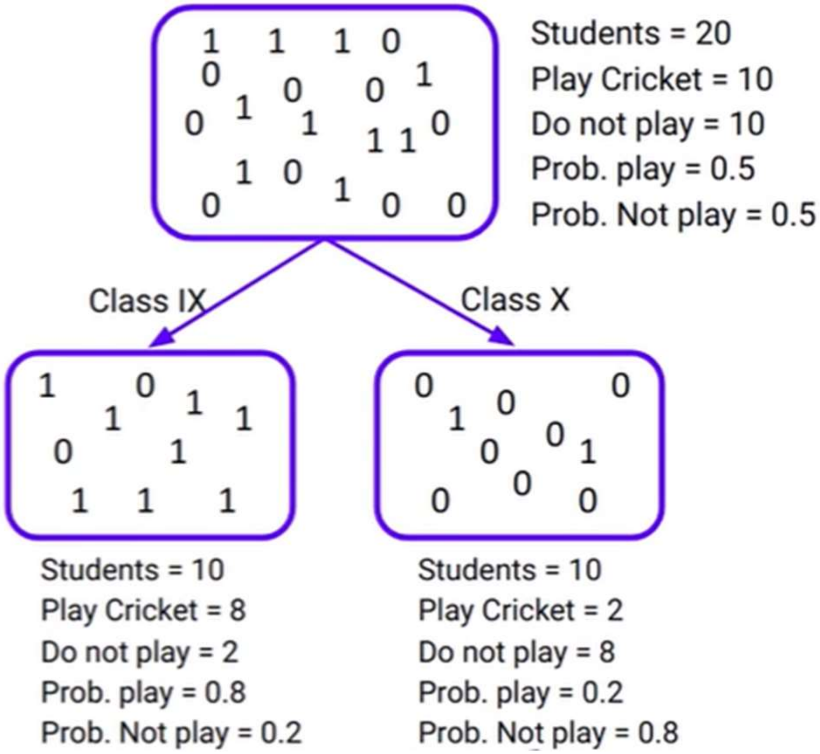
- Plays Cricket = 1
- Do not play Cricket = 0

Steps to Calculate Variance

- Above Average node:
  - Mean = (8*1 + 6*0) / 14 = 0.57
  - Variance =
    $[8*(1-0.57)^2 + 6*(0-0.57)^2] / 14 = 0.245$
- Below Average node:
  - Mean = (2*1 + 4*0) / 6 = 0.33
  - Variance =
    $[2*(1-0.33)^2 + 4*(0-0.33)^2] / 6 = 0.222$
- Variance: Performance in Class:
  (14/20)*0.245 + (6/20)*0.222 = 0.238



Steps to Calculate Variance



Steps to Calculate Variance

- Class IX node:
  - Mean = (8*1 + 2*0) / 10 = 0.8
  - Variance = $[8*(1-0.8)^2 + 2*(0-0.8)^2] / 10 = 0.16$

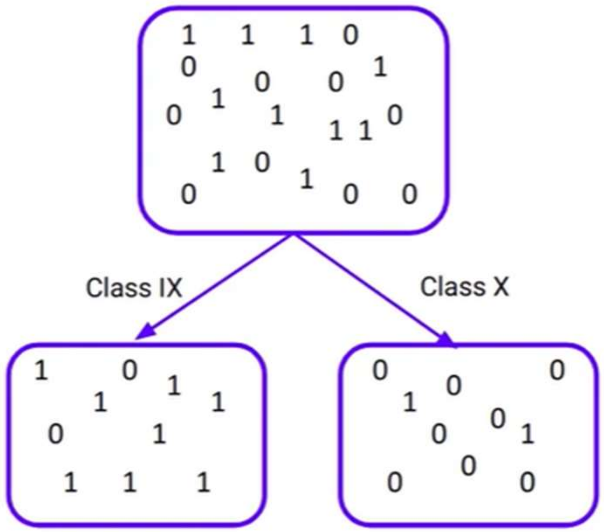
- Class X node:
  - Mean = (2*1 + 8*0) / 10 = 0.2
  - Variance = $[2*(1-0.2)^2 + 8*(0-0.2)^2] / 10 = 0.16$

- Variance: Class:
  (10/20)*0.16 + (10/20)*0.16 = 0.16

Steps to Calculate Variance

Split	Variance
Performance in Class	0.238
Class	0.16

Steps to Calculate Variance



Split on Class

Points to Remember

- Don’t grow too much deeper trees, because it leads to **overfitting (Applying pre-pruning and post pruning can save you)**

- Don’t grow too much shallow trees, because it leads to **underfitting (Increase Tree Depth, Use More Additional Features)**

Decision Tree Algorithms

- **CART (Classification and Regression Trees):**
  - Used for both classification and regression tasks
  - CART constructs binary trees by splitting data using the
    - Gini impurity for classification
    - Mean squared error for regression.

Decision Tree Algorithms

- **ID3 (Iterative Dichotomiser 3):**
  - Primarily used for classification
  - ID3 employs a top-down approach to select the best attribute at each node using information gain, which measures how well an attribute separates the classes.

Decision Tree Algorithms

- **C4.5:An extension of ID3:**
  - C4.5 handles both categorical and continuous data
  - It utilizes the gain ratio to select splits and allowing for the handling of missing values and pruning to avoid overfitting.