Decision Tree

Todays Agenda

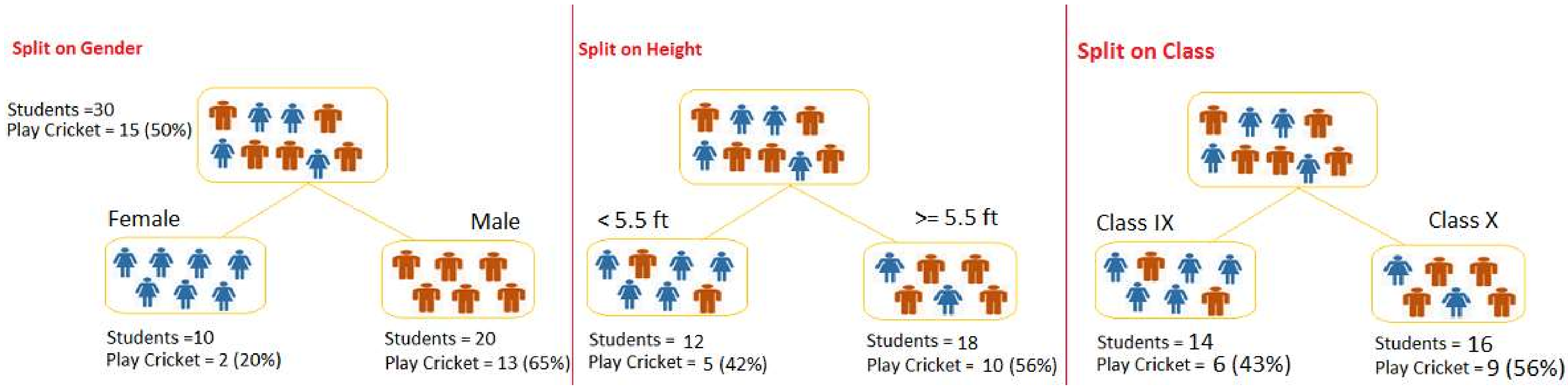
- What is Decision Tree
- Types of Decision Tree
- Terminologies in Decision Tree
- Advantages and Disadvantages
- How does a tree decide where to split
- Solution to Decision Tree issues
- Tree vs Linear based models

What is Decision Tree

Supervised learning algorithm.

Mostly used in classification problems

Works for both categorical and continuous input and output variables.


Split on Gender
Students =30
Play Cricket = 15 (50%)
Female
Male
Students =10
Play Cricket = 2 (20%)
Students = 20
Play Cricket = 13 (65%)
Split on Height
< 5.5 ft
>= 5.5 ft
Students = 12
Play Cricket = 5 (42%)
Students = 18
Play Cricket = 10 (56%)
Split on Class
Class IX
Class X
Students = 14
Play Cricket = 6 (43%)
Students = 16
Play Cricket = 9 (56%)

Example

Types of Decision Tree

1

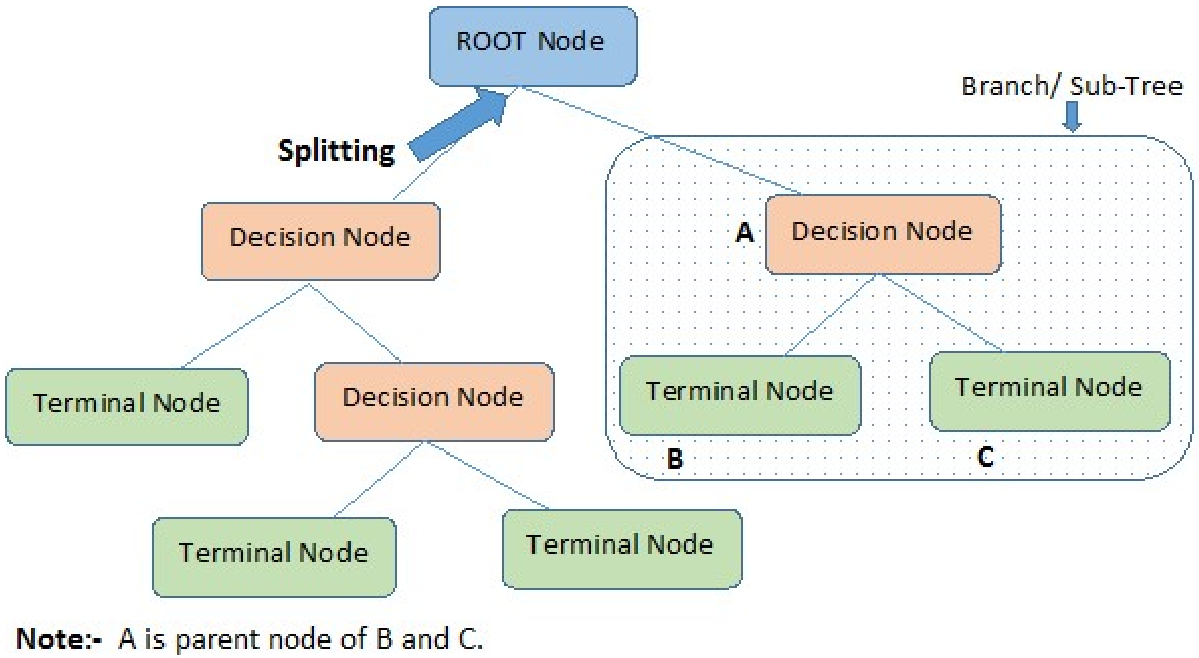
Categorical Variable Decision Tree:

Decision Tree with categorical target variable is called as categorical variable decision tree.

2

Continuous Variable Decision Tree:

Decision Tree with continuous target variable is called as Continuous Variable Decision Tree.



Terminologies

- **Root Node**
- **Splitting**
- **Decision Node**
- **Leaf/Terminal Node**
- **Pruning**
- **Branch/Sub-Tree**
- **Parent and Child Node**

Advantages

- Easy to Understand
- Useful in Data exploration
- Less Data cleaning is required
- Data type is not a constraint
- Non-Parametric method

Disadvantages

Overfitting: Solved by setting constraints on model parameters and pruning

Not fit for Continuous variables: While working with continuous numerical variables, decision tree looses information when it categorizes variables in different categories.

How does a tree decide where to split?

- Gini
- Chi-Square
- Information Gain
- Reduction in Variance

Gini

- It works with categorical target variable "Success" or "Failure".
- It performs only Binary splits
- Higher the value of Gini higher the homogeneity.
- CART (Classification and Regression Tree) uses Gini method to create binary splits.

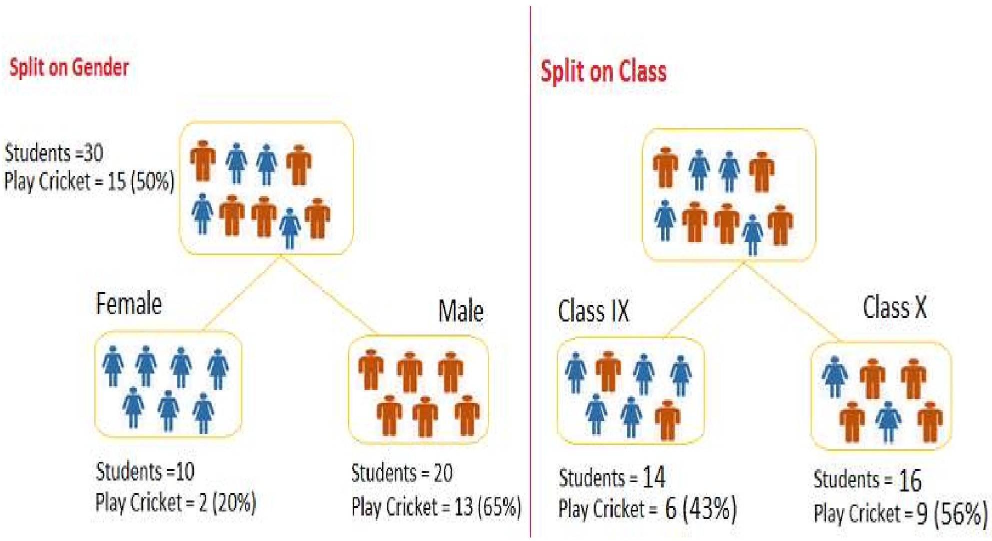
Gini

Steps to Calculate Gini for a split

Calculate Gini for sub-nodes, using formula sum of square of probability for success and failure **(p^2+q^2).**

Calculate Gini for split using weighted Gini score of each node of that split

Example



Split on Gender:

Gini for sub-node Female = (0.2)*(0.2)+(0.8)*(0.8)=0.68

Gini for sub-node Male = (0.65)*(0.65)+(0.35)*(0.35)=0.55

Weighted Gini for Split Gender = (10/30)*0.68+(20/30)*0.55 = **0.59**

Split on Class:

Gini for sub-node Class IX = (0.43)*(0.43)+(0.57)*(0.57)=0.51

Gini for sub-node Class X = (0.56)*(0.56)+(0.44)*(0.44)=0.51

Weighted Gini for Split Class = (14/30)*0.51+(16/30)*0.51 = **0.51**

Chi-Square

- It works with categorical target variable “Success” or “Failure”.
- It can perform two or more splits.
- Higher the value of Chi-Square higher the statistical significance of differences between sub-node and Parent node.

Chi-Square

Steps to Calculate Chi-square for a split:

- Calculate Chi-square for individual node by calculating the deviation for Success and Failure both.
- Calculated Chi-square of Split using Sum of all Chi-square of success and Failure of each node of the split.
- Chi-Square of each node is calculated using formula,

Chi-square = ((Actual – Expected)^2 / Expected)^1/2

Example

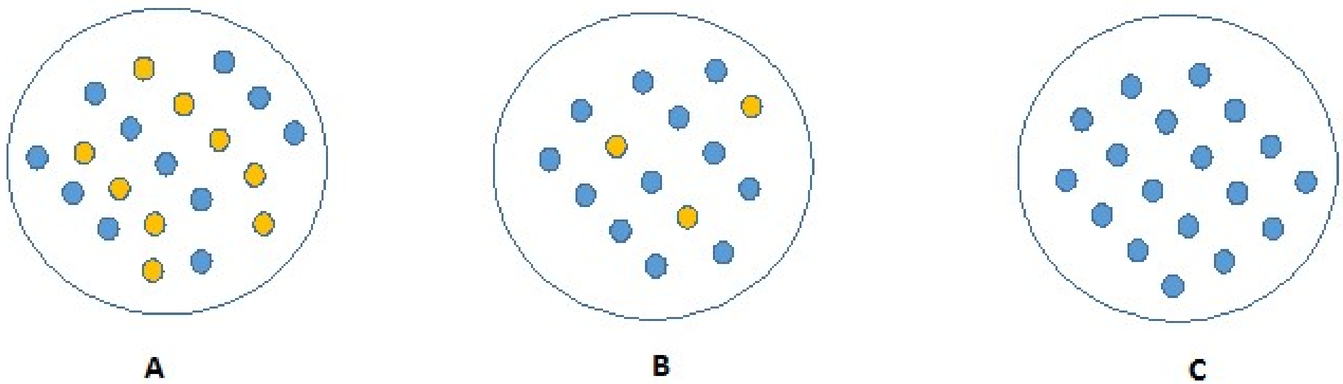
Split on Gender:

Node	Play Cricket	Not Play Cricket	Total	Expected Play Cricket	Expected Not Play Cricket	Deviation Play Cricket	Deviation Not Play Cricket	Chi-Square	
								Play Cricket	Not Play Cricket
Female	2	8	10	5	5	-3	3	1.34	1.34
Male	13	7	20	10	10	3	-3	0.95	0.95
							Total Chi-Square	**4.58**	

Split on Class:

Node	Play Cricket	Not Play Cricket	Total	Expected Play Cricket	Expected Not Play Cricket	Deviation Play Cricket	Deviation Not Play Cricket	Chi-Square	
								Play Cricket	Not Play Cricket
IX	6	8	14	7	7	-1	1	0.38	0.38
X	9	7	16	8	8	1	-1	0.35	0.35
							Total Chi-Square	**1.46**	

Information Gain



Information Gain

Information theory is a measure to define this degree of disorganization in a system known as Entropy.

Entropy = 0; If Sample is completely homogeneous

Entropy = 1; If Sample is equally divided(50%-50%)

Information Gain = 1 - Entropy

Information Gain

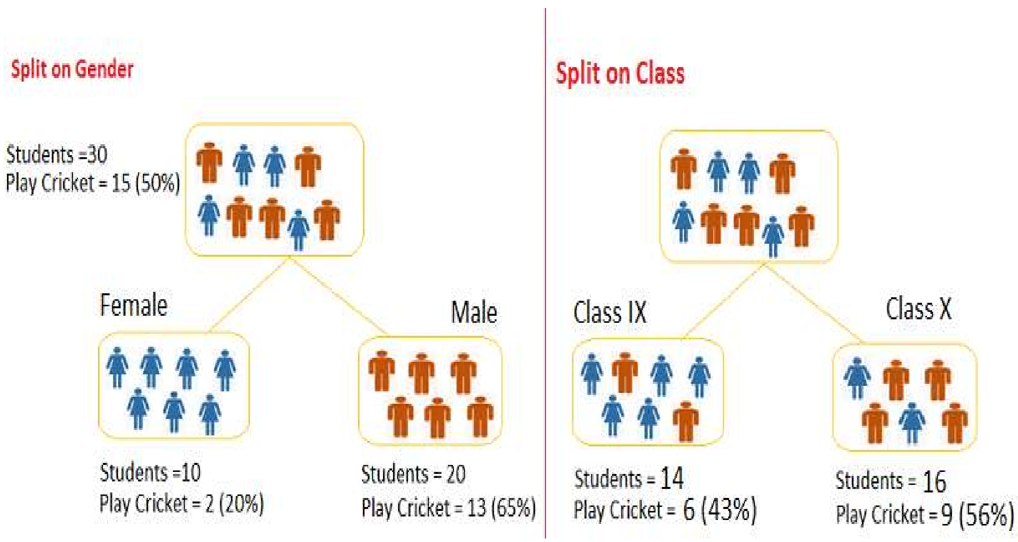
Steps to calculate entropy for a split:

- Calculate entropy of parent node
- Calculate entropy of each individual node of split and calculate weighted average of all sub-nodes available in split.

$$\textbf{Entropy} = -p \log_2 p - q \log_2 q$$

- p and q is probability of success and failure.
- Entropy is used with categorical target variable.
- It chooses the split which has lowest entropy compared to parent node and other splits. The lesser the entropy, the better it is.

Example



Entropy for parent node = -(15/30) log2 (15/30) – (15/30) log2 (15/30) = **1**.

Here 1 shows that it is an impure node.

Entropy for Female node = -(2/10) log2 (2/10) – (8/10) log2 (8/10) = 0.72

Entropy for Female node = -(13/20) log2 (13/20) – (7/20) log2 (7/20) = **0.93**

Entropy for split Gender = Weighted entropy of sub-nodes

= (10/30)*0.72 + (20/30)*0.93 = **0.86**

Entropy for Class IX node = -(6/14) log2 (6/14) – (8/14) log2 (8/14) = 0.99

Entropy for Class X node = -(9/16) log2 (9/16) – (7/16) log2 (7/16) = 0.99.

Entropy for split Class = (14/30)*0.99 + (16/30)*0.99 = **0.99**

Reduction in Variance

- Reduction in variance is an algorithm used for continuous target variables (regression problems).
- The split with lower variance is selected as the criteria to split the population:

$$\text{Variance} = \frac{\Sigma(X - \overline{X})^2}{n}$$

X-bar : Mean of the values,

X : Actual value

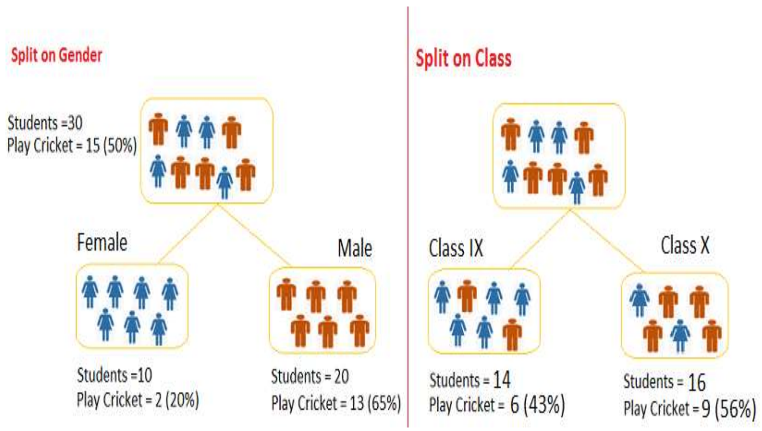
n : is number of values.

Reduction in Variance

Steps to calculate Variance:

- Calculate variance for each node.
- Calculate variance for each split as weighted average of each node variance.

Example



Let's assign play cricket=1 and not playing cricket=0.

Now follow the steps to identify the right split:

Mean = (15*1 + 15*0)/30 = 0.5.

Variance for Root node = ((1-0.5)^2+(1-0.5)^2+….15 times+(0-0.5)^2+(0-0.5)^2+…15 times) / 30 = (15*(1-0.5)^2+15*(0-0.5)^2) / 30 = **0.25**

Mean of Female node = (2*1+8*0)/10=0.2 and Variance = (2*(1-0.2)^2+8*(0-0.2)^2) / 10 = 0.16

Mean of Male Node = (13*1+7*0)/20=0.65 and Variance = (13*(1-0.65)^2+7*(0-0.65)^2) / 20 = 0.23

Variance for Split Gender = Weighted Variance of Sub-nodes = (10/30)*0.16 + (20/30) *0.23 = **0.21**

Mean of Class IX node = (6*1+8*0)/14=0.43 and Variance = (6*(1-0.43)^2+8*(0-0.43)^2) / 14= 0.24

Mean of Class X node = (9*1+7*0)/16=0.56 and Variance = (9*(1-0.56)^2+7*(0-0.56)^2) / 16 = 0.25

Variance for Split Class = (14/30)*0.24 + (16/30) *0.25 = **0.25**

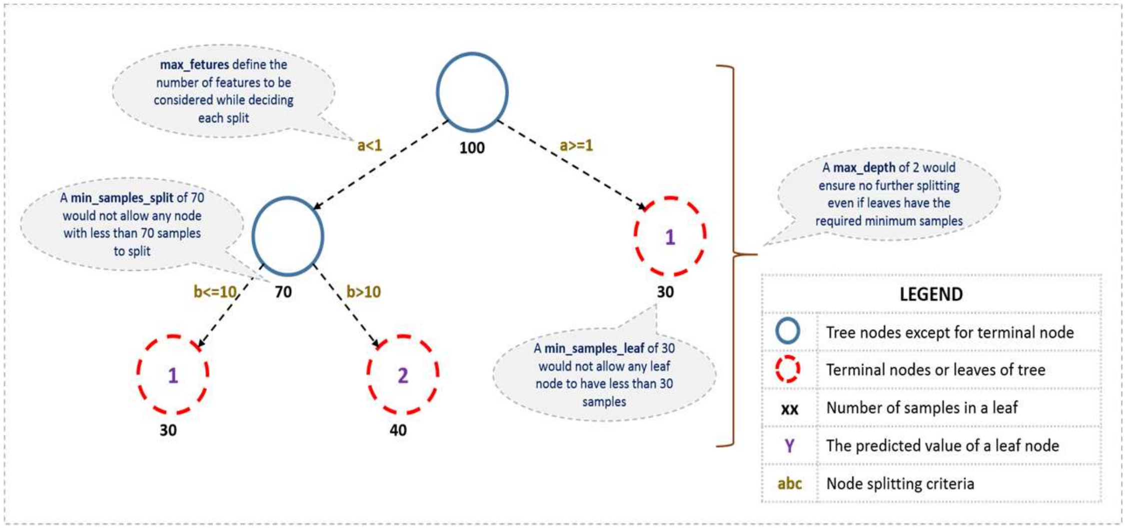
Solution to Decision Tree issue

Overfitting is one of the key challenges faced while modeling decision trees.

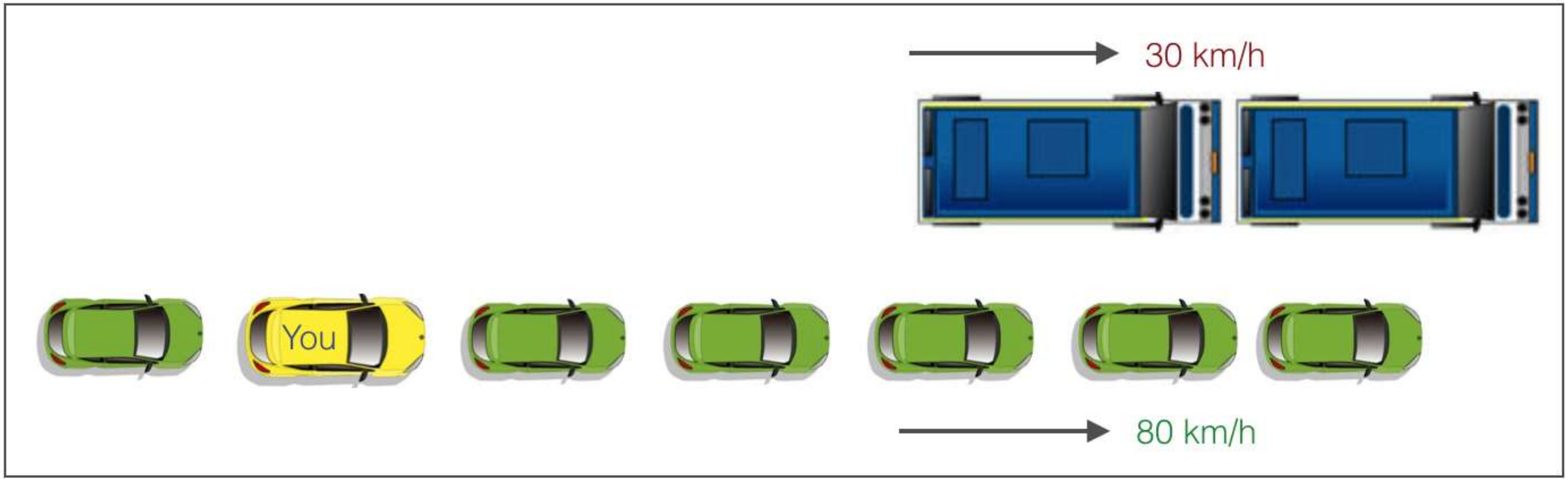
Prevention of over-fitting is done in 2 ways:

- Setting constraints on tree size
- Tree pruning

Setting Constraints on tree Size



- Minimum samples for a node split
- Minimum samples for a terminal node (leaf)
- Maximum depth of tree (vertical depth)
- Maximum number of terminal nodes
- Maximum features to consider for split



Tree Pruning

How to implement pruning in Decision Tree

- We first make the decision tree to a large depth.
- Then we start at the bottom and start removing leaves which are giving us negative returns when compared from the top.
- Suppose a split is giving us a gain of say -10 (loss of 10) and then the next split on that gives us a gain of 20. A simple decision tree will stop at step 1 but in pruning, we will see that the overall gain is +10 and keep both leaves.

Important points

- Sklearn’s decision tree classifier does not currently support pruning.
- Advanced packages like **xgboost** have adopted tree pruning in their implementation.
- The library ***rpart, tree*** in R, provides a function to prune. Good for R users!

Tree vs Linear based models

Key factors in deciding which algorithm to choose

- If the relationship between dependent & independent variable is well approximated by a linear model, linear regression will outperform tree-based model.
- If there is a high non-linearity & complex relationship between dependent & independent variables, a tree model will outperform a classical regression method.
- If you need to build a model which is easy to explain to people, a decision tree model will always do better than a linear model. Decision tree models are even simpler to interpret than linear regression.