Machine Learning 520
Advanced Machine Learning

Lesson 2: Decision Trees

UNIVERSITY of WASHINGTON

Today’s Agenda

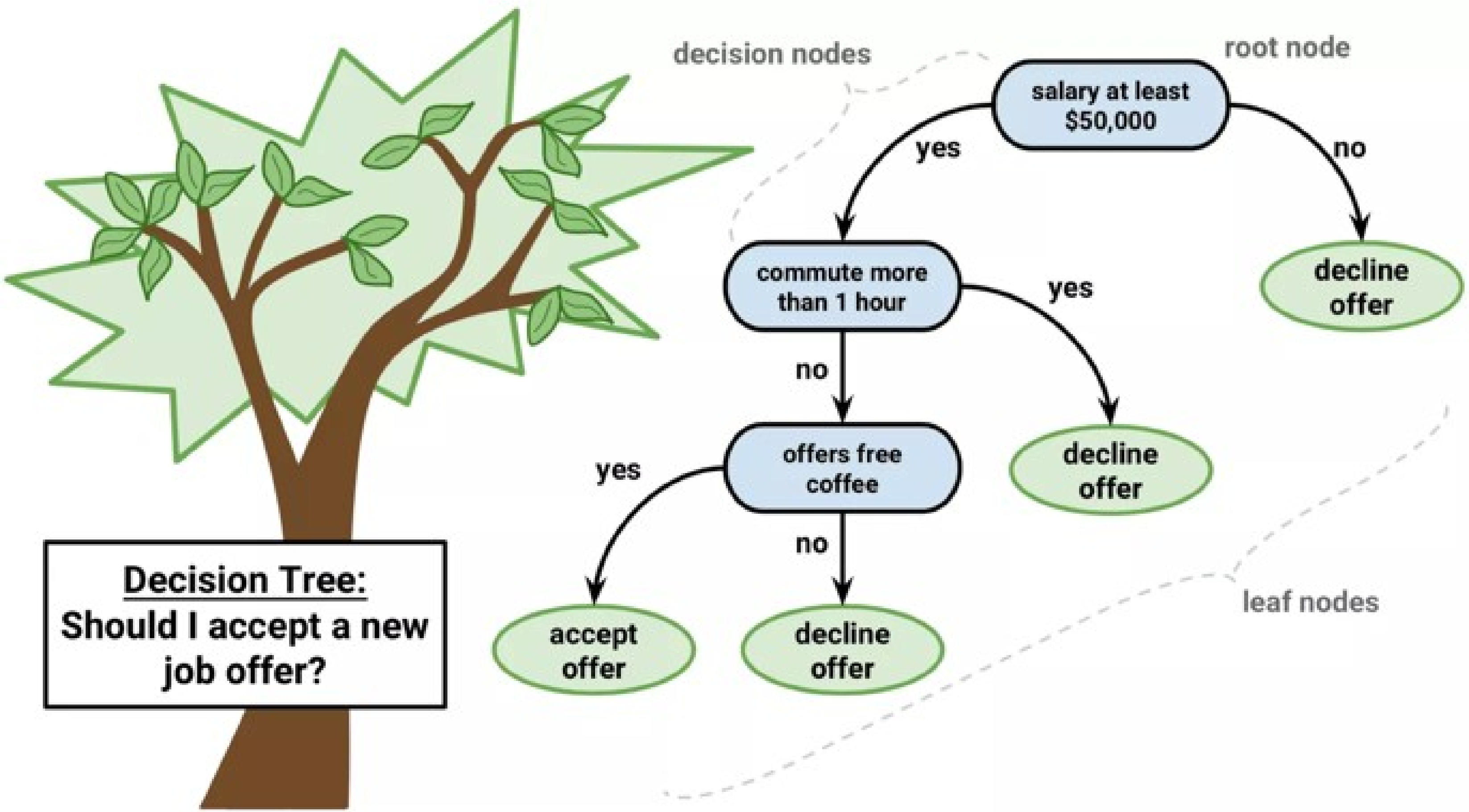
- Gini impurity
- Entropy
- Information gain
- Classification tree
- Regression tree
- Variable importance

Learning Objective

By the end of this session, you should be able to:

- Explain/demonstrate how Gini impurity is used for constructing decision tree.
- Compare entropy to Gini impurity.
- Explain/demonstrate how information gain is used for constructing decision tree.
- Differentiate between classification tree vs. regression tree.
- Apply recursive partitioning used for constructing classification / regression tree.
- Tune
- Apply regularization to decision tree to prevent overfitting.
- Extract variable importance to interpret the decision tree model.

The Basics of Decision Trees



Types of Decision Trees

- **Categorical Variable Decision Tree:** Decision Tree which has categorical target variable

- **Continuous Variable Decision Tree:** Decision Tree which has continuous target variable

Pros and Cons

- Tree-based methods are simple and useful for interpretation.
- Tend to overfit leading to poor performance.
- We also discuss bagging, random forests, and boosting.
- Combining multiple trees to often yield improvements in prediction accuracy, at the expense of some loss interpretation.

How does a tree decide where to split?

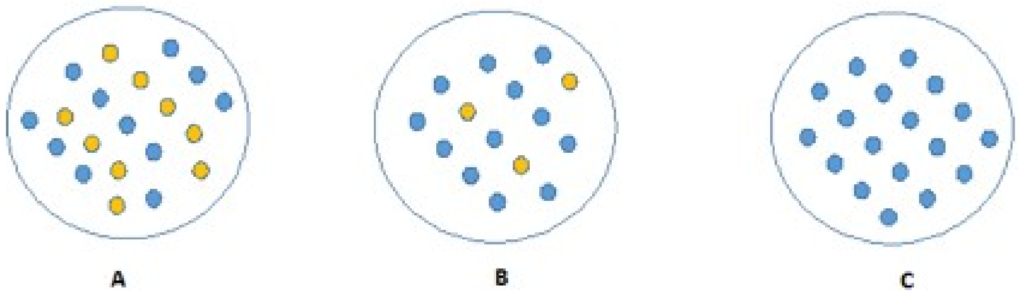
Four commonly used algorithms in decision tree:

Gini Impurity

- Compute Gini impurity for a set of items with J classes, where i in {1, 2, .., J} and p_i be the fraction of items labeled with class i in the set.
- It works with categorical target variable “Success” or “Failure”.
- Higher the value of Gini higher the homogeneity.
- CART (Classification and Regression Tree) uses Gini method to create binary splits.

How does a tree decide where to split?

Information Gain:



- Less impure node requires less information to describe it and, more impure node requires more information. Information theory is a measure to define this degree of disorganization in a system known as Entropy. If the sample is completely homogeneous, then the entropy is zero and if the sample is an equally divided (50% – 50%), it has entropy of one.

How does a tree decide where to split?

Entropy

- If you have two labels

Here p and q is probability of success and failure respectively in that node.

Entropy is also 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.

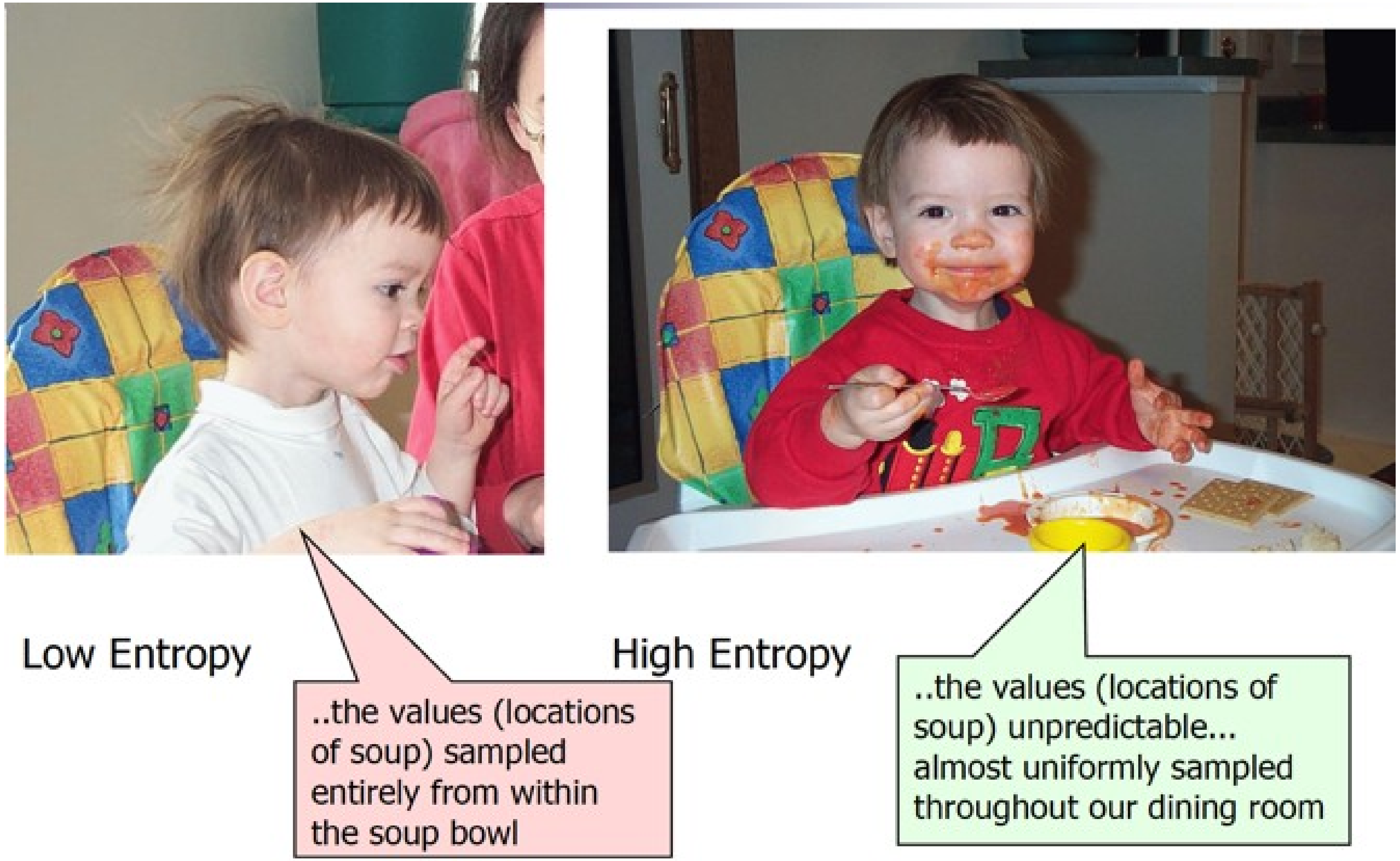
- More than two labels

Andrew Moore's Entropy in a nutshell

Low Entropy

High Entropy

Andrew Moore’s Entropy in a nutshell



How does a tree decide where to split?

- **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 of each node is calculated using formula,
  - Chi-square = ((Actual – Expected)^2 / Expected)^1/2
  - It generates tree called CHAID (Chi-square Automatic Interaction Detector)

How does a tree decide where to split?

Reduction in Variance

- Used for regression problems

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

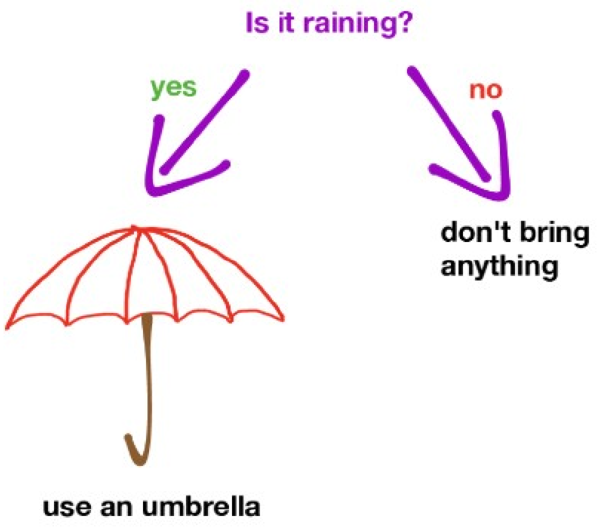
- Above X-bar is mean of the values, X is actual, and n is number of values.

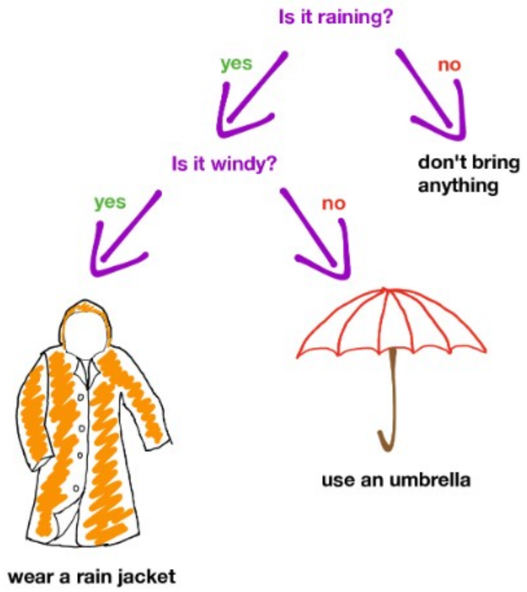
Steps to calculate Variance:

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

How to construct a Decision Tree?

- Choose an attribute (i.e. a feature) for root.
- Split data using chosen attribute into disjoint subsets.
- Recursive partitioning for each subset.

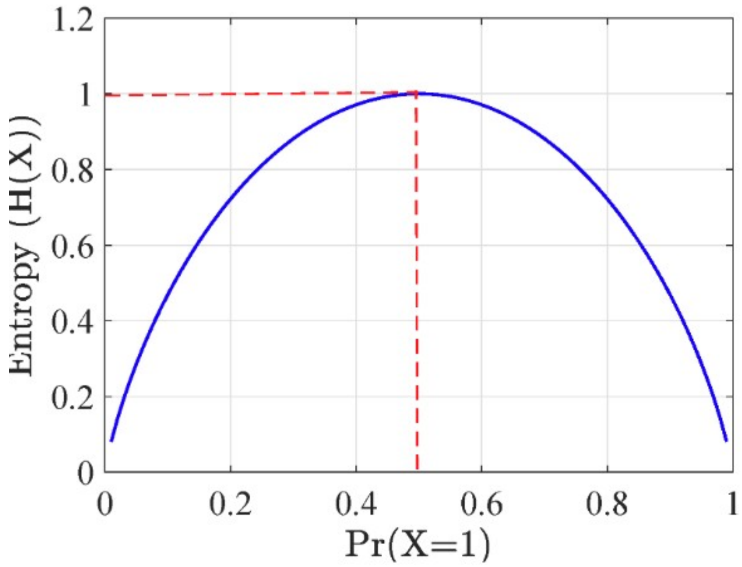




Classification Impurity Measure: Entropy

- **Entropy** measures the level of **impurity** in a group of examples for classification problems.

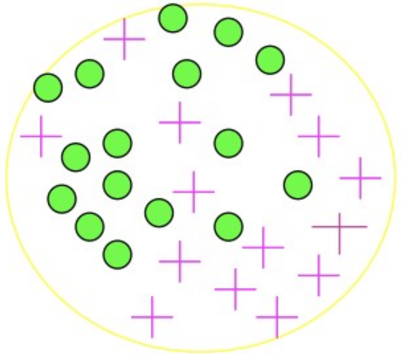
$$H(x) = -\sum_i p_i \log(p_i)$$



16/30 are green circles; 14/30 are pink crosses

$\log_2(16/30) = -.9$; $\log_2(14/30) = -1.1$

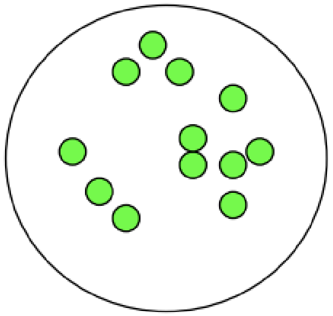
Entropy = -(16/30)(-.9) –(14/30)(-1.1) = .99



Entropy for 2-class Cases

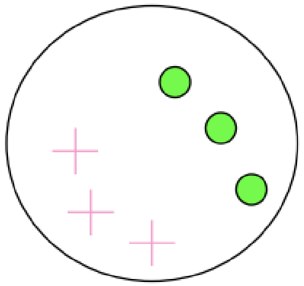
- What is the entropy of a group in which all examples belong to the same class?
  - entropy = $-\ 1\ \log_2 1 = 0$

Minimum impurity



- What is the entropy of a group with 50% in either class?
  - entropy = $-0.5\ \log_2 0.5 - 0.5\ \log_2 0.5 = 1$

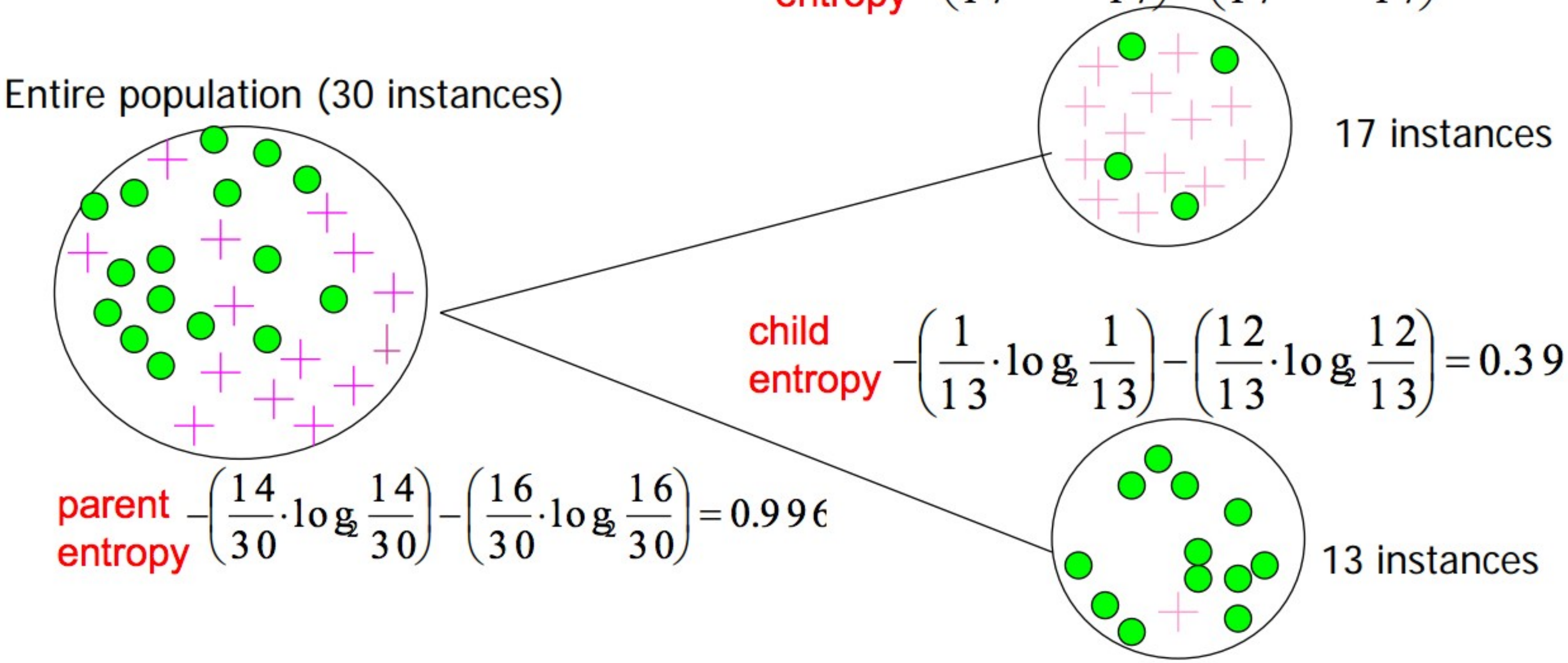
Maximum impurity



Information Gain Example

Information Gain = entropy(parent) – [average entropy(children)]

child entropy $-\left(\frac{13}{17}\cdot\log_2\frac{13}{17}\right)-\left(\frac{4}{17}\cdot\log_2\frac{4}{17}\right)=0.787$



child entropy $-\left(\frac{1}{13}\cdot\log_2\frac{1}{13}\right)-\left(\frac{12}{13}\cdot\log_2\frac{12}{13}\right)=0.391$

parent entropy $-\left(\frac{14}{30}\cdot\log_2\frac{14}{30}\right)-\left(\frac{16}{30}\cdot\log_2\frac{16}{30}\right)=0.996$

(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

Toy Example

Training Set: 3 features and 2 classes

X	Y	Z	C
1	1	1	I
1	1	0	I
0	0	1	II
1	0	0	II

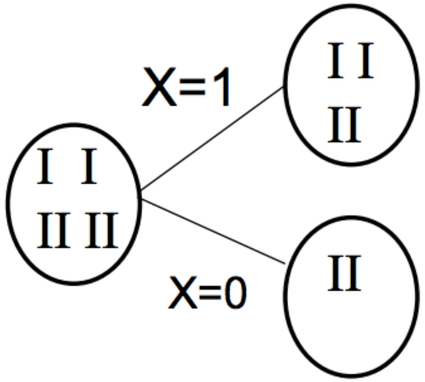
How would you distinguish class I from class II?

Toy Example

X	Y	Z	C
1	1	1	I
1	1	0	I
0	0	1	II
1	0	0	II

Split on attribute X

If X is the best attribute,
this node would be further split.



$E_{child1} = -(1/3)\log_2(1/3)-(2/3)\log_2(2/3)$
$= .5284 + .39$
$= .9184$

$E_{child2} = 0$

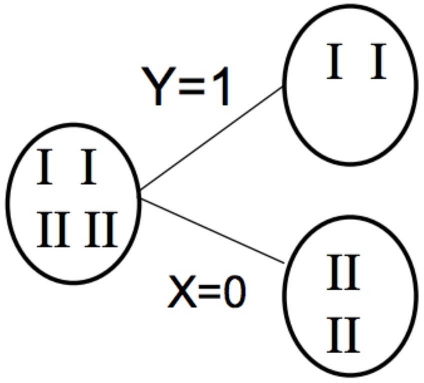
$E_{parent} = 1$

GAIN = 1 – (3/4)(.9184) – (1/4)(0) = .3112

Toy Example

X	Y	Z	C
1	1	1	I
1	1	0	I
0	0	1	II
1	0	0	II

Split on attribute Y



$E_{child1} = 0$

$E_{child2} = 0$

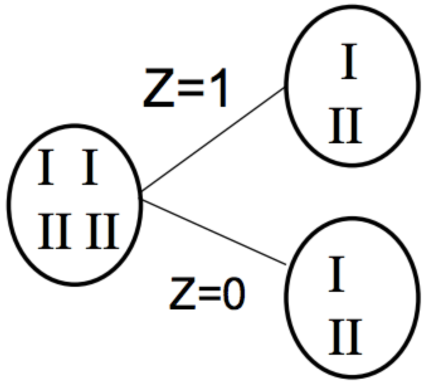
$E_{parent} = 1$

GAIN = 1 –(1/2) 0 – (1/2)0 = 1; BEST ONE

Toy Example

X	Y	Z	C
1	1	1	I
1	1	0	I
0	0	1	II
1	0	0	II

Split on attribute Z



$E_{child1} = 1$

$E_{child2} = 1$

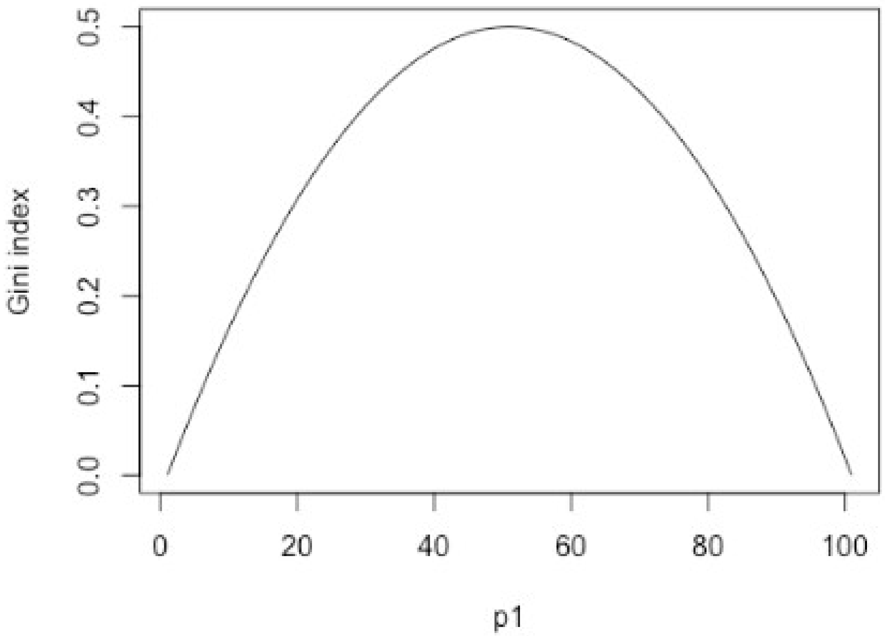
$E_{parent} = 1$

GAIN = 1 – (1/2)(1) – (1/2)(1) = 0   ie. NO GAIN; WORST

Classification Impurity Measure: Gini Impurity

- **Gini impurity/index** is another measure to quantify the level of impurity in a group of examples.
  - I(A) = 0 when all cases belong to the same class.
  - Max value when all classes are equally represented.

$$I(x) = 1 - \sum_i p\, i$$



Split of a Numerical Variable

> For each numerical attribute:
  - Sort the attribute from the smallest to the largest.
  - Linearly scan these values and choose the split position leading to the maximum impurity reduction (i.e. information gain).

	Cheat	No	No	No	Yes	Yes	Yes	No	No	No	No
		Taxable Income									
Sorted Values →		60	70	75	85	90	95	100	120	125	220

Split Positions →	55		65		72		80		87		92		97		110		122		172		230	
	<=	>	<=	>	<=	>	<=	>	<=	>	<=	>	<=	>	<=	>	<=	>	<=	>	<=	>
Yes	0	3	0	3	0	3	0	3	1	2	2	1	3	0	3	0	3	0	3	0	3	0
No	0	7	1	6	2	5	3	4	3	4	3	4	3	4	4	3	5	2	6	1	7	0
Gini	0.420		0.400		0.375		0.343		0.417		0.400		0.300		0.343		0.375		0.400		0.420	

Over-fitting in decision trees?

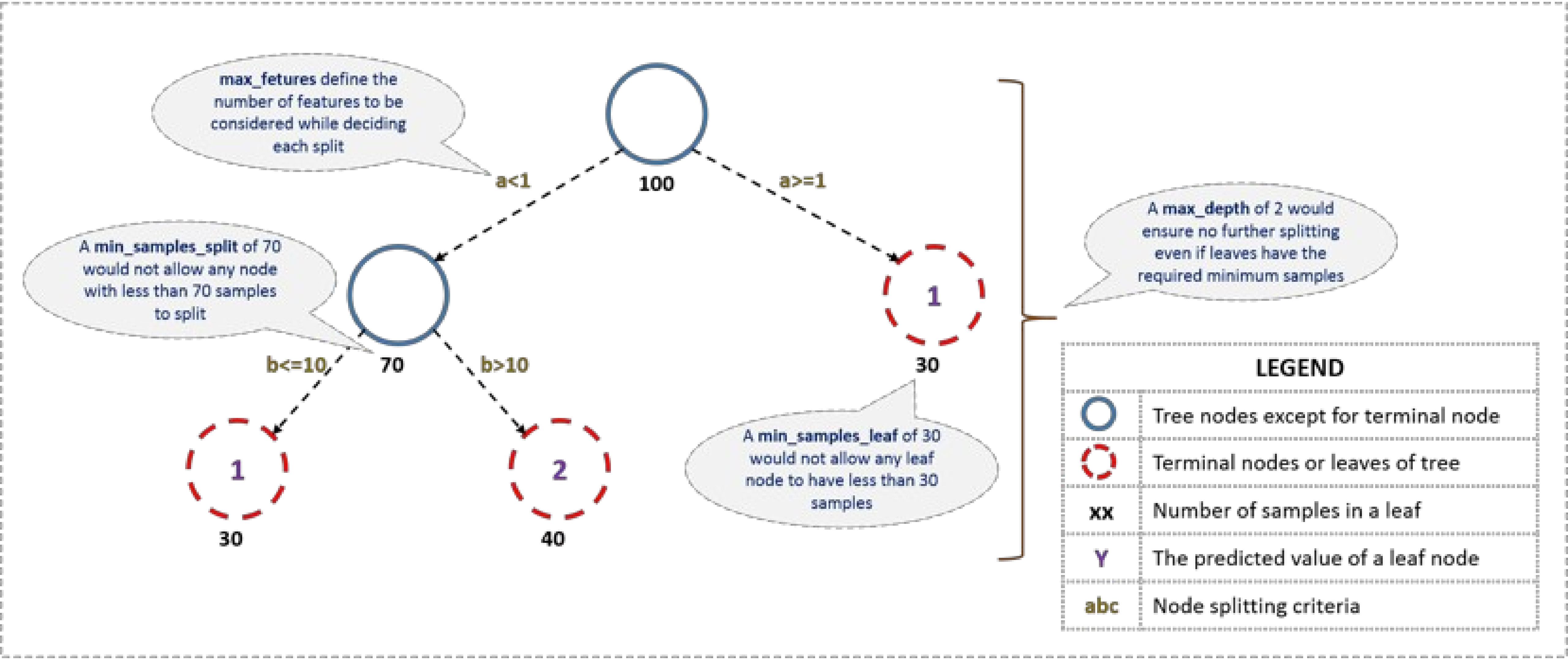
Overfitting is one of the key challenges faced while modeling decision trees.

If there is no limit set of a decision tree, it will give you 100% accuracy on training set because in the worse case it will end up making 1 leaf for each observation.

Thus, preventing overfitting is pivotal while modeling a decision tree and it can be done in 2 ways:

- **Setting constraints on tree size**
- **Tree pruning**

Setting Constraints on Tree Size



Setting Constraints on Tree Size

Minimum samples for a node split

- Defines the minimum number of samples (or observations) which are required in a node to be considered for splitting.
- Used to control over-fitting. Higher values prevent a model from learning relations which might be highly specific to the particular sample selected for a tree.
- Too high values can lead to under-fitting hence, it should be tuned using CV.

Setting Constraints on Tree Size

Minimum samples for a terminal node (leaf)

Defines the minimum samples (or observations) required in a terminal node or leaf.

- Used to control over-fitting similar to min_samples_split.

- Generally lower values should be chosen for imbalanced class problems because the regions in which the minority class will be in majority will be very small.

Setting Constraints on Tree Size

Maximum depth of tree (vertical depth)

- The maximum depth of a tree.
- Used to control over-fitting as higher depth will allow model to learn relations very specific to a particular sample.
- Should be tuned using CV.

Tree Pruning

The technique of setting constraint is a greedy-approach. In other words, it will check for the best split instantaneously and move forward until one of the specified stopping conditions are reached.

Pruning works differently:

- 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.

Pruning Example