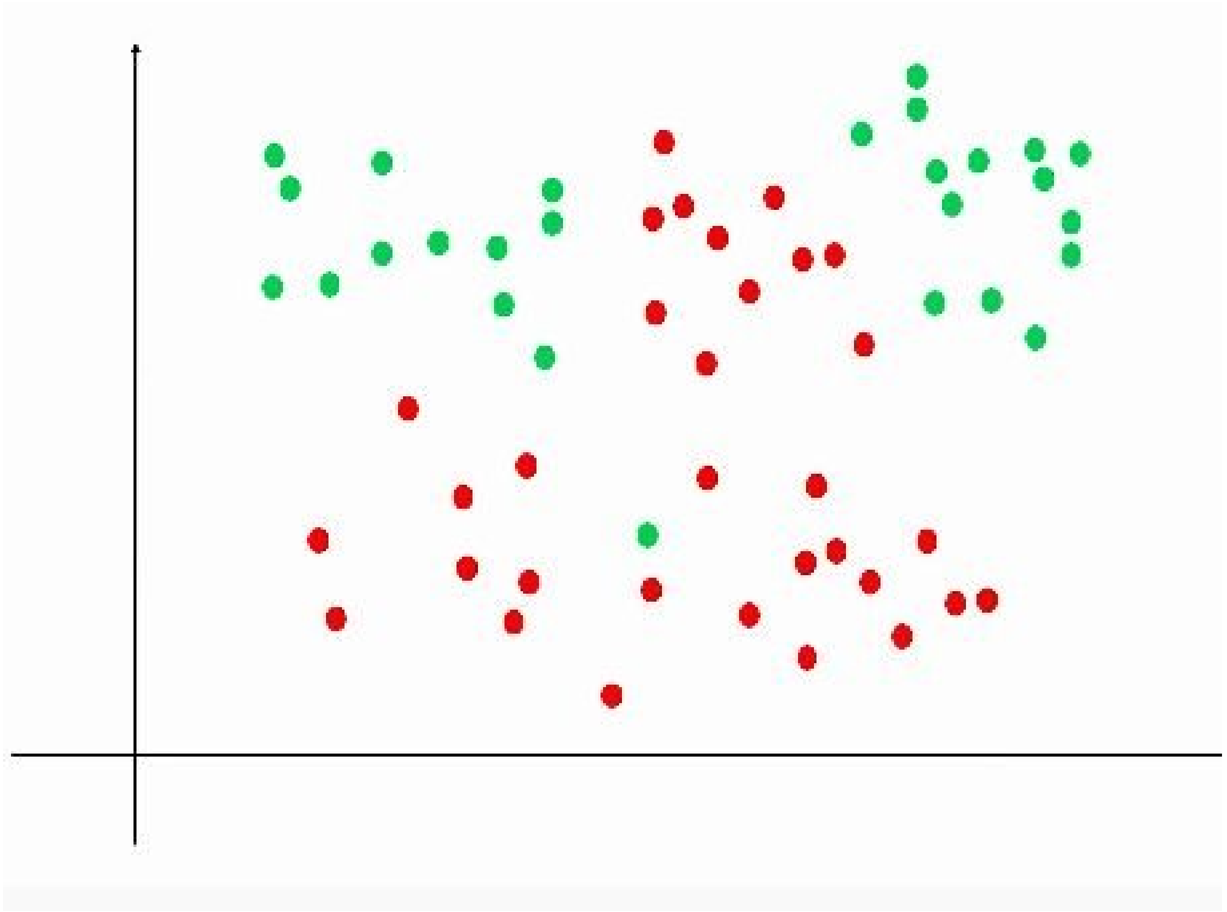
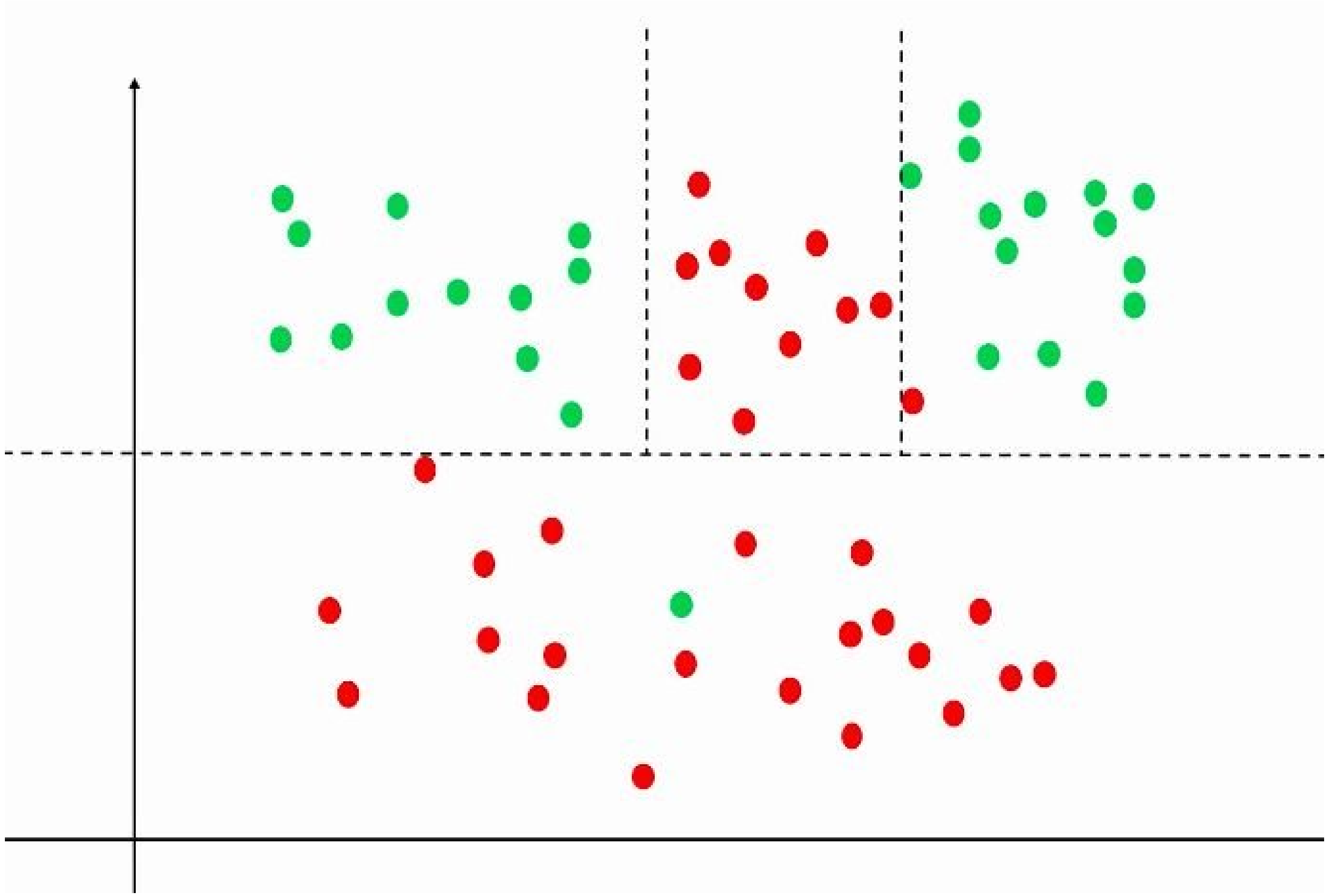
Unit IV Supervised Models

- Classification Decision trees- Overview, general algorithm, decision tree algorithm, evaluating a decision tree using Gini Index and Entropy ,Naïve Bayes – Bayes Theorem and Algorithm, Naïve Bayes Classifier, smoothing, diagnostics. Diagnostics of classifiers, additional classification methods.

Decision Tree



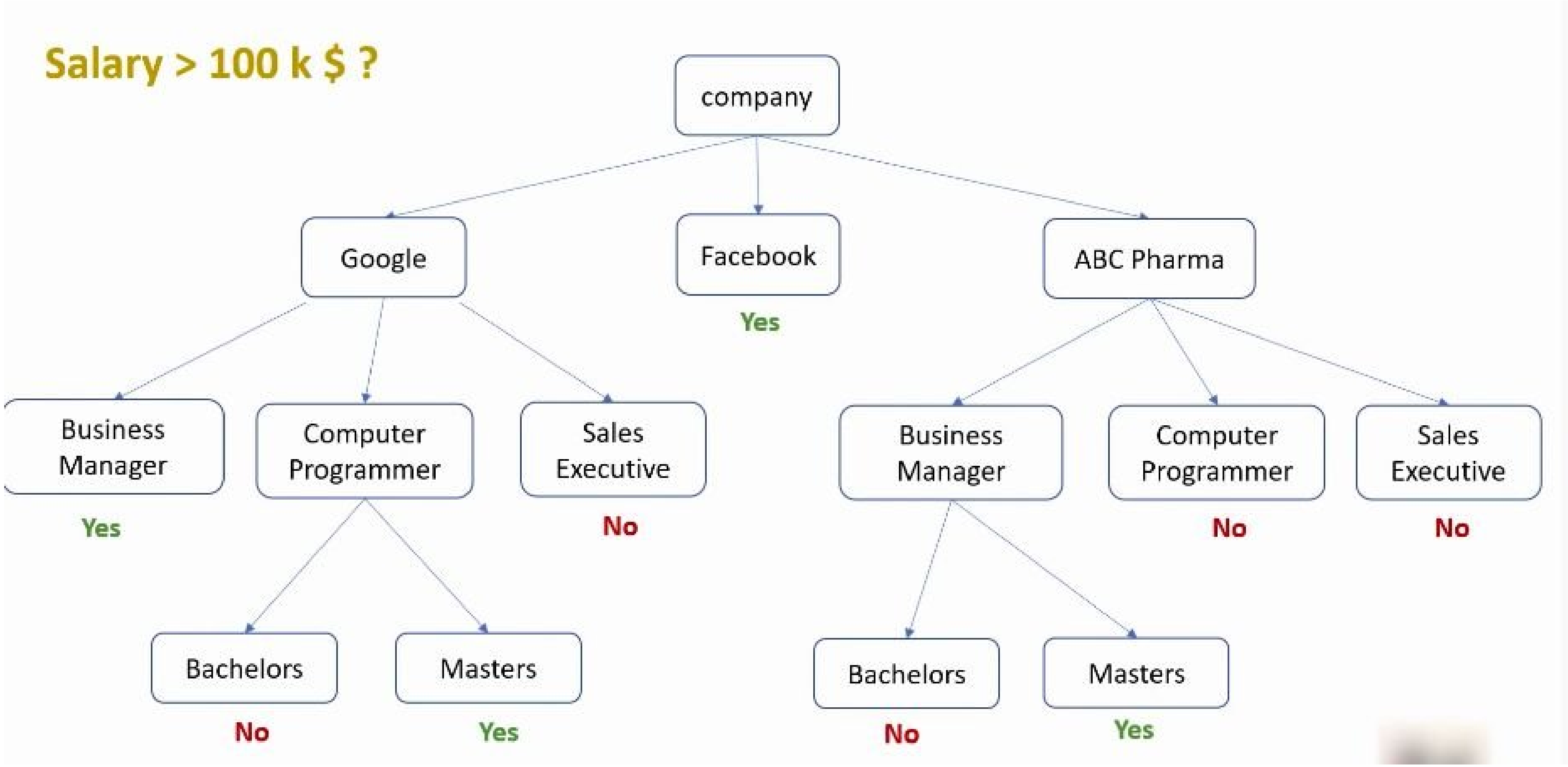
Decision Tree



Decision Tree

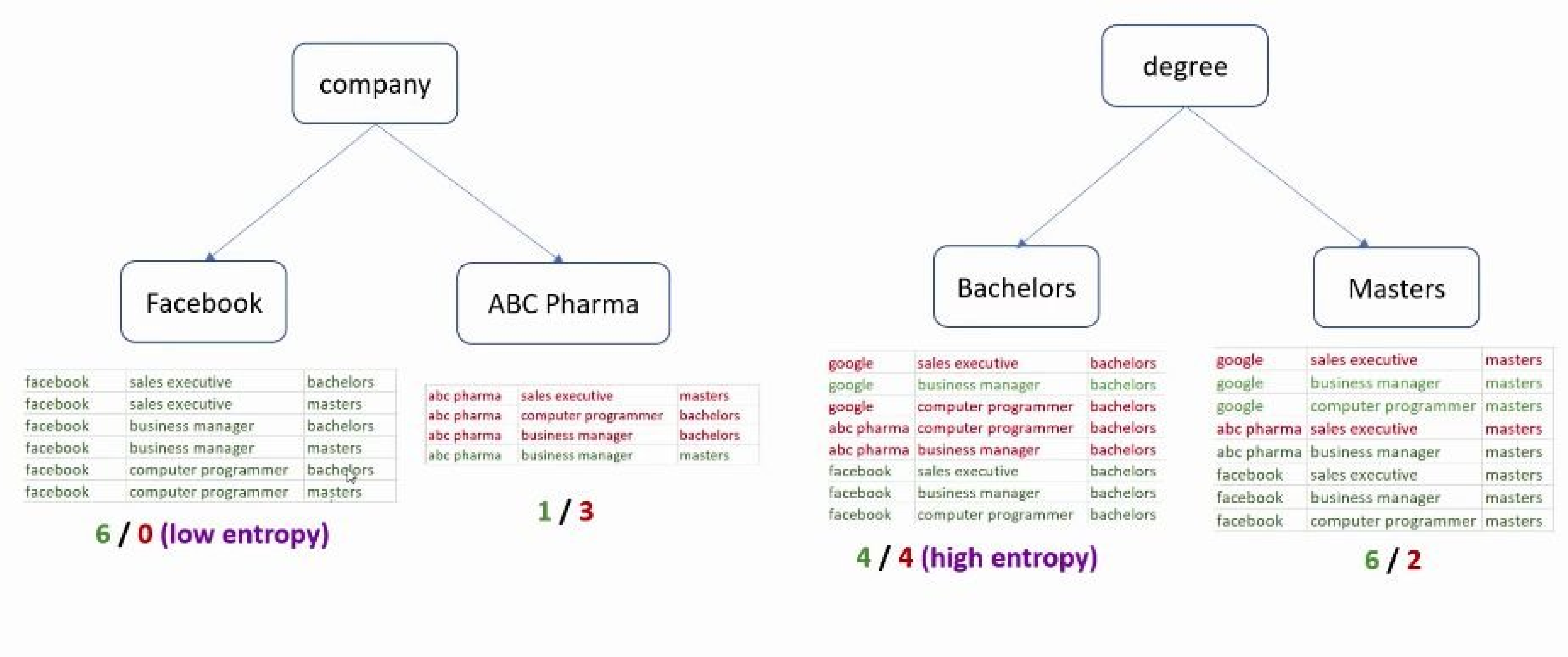
- It is a tree-structured classifier
- Where internal nodes represent the features of a dataset
- branches represent the decision rules
- Each leaf node represents the outcome

Decision Tree

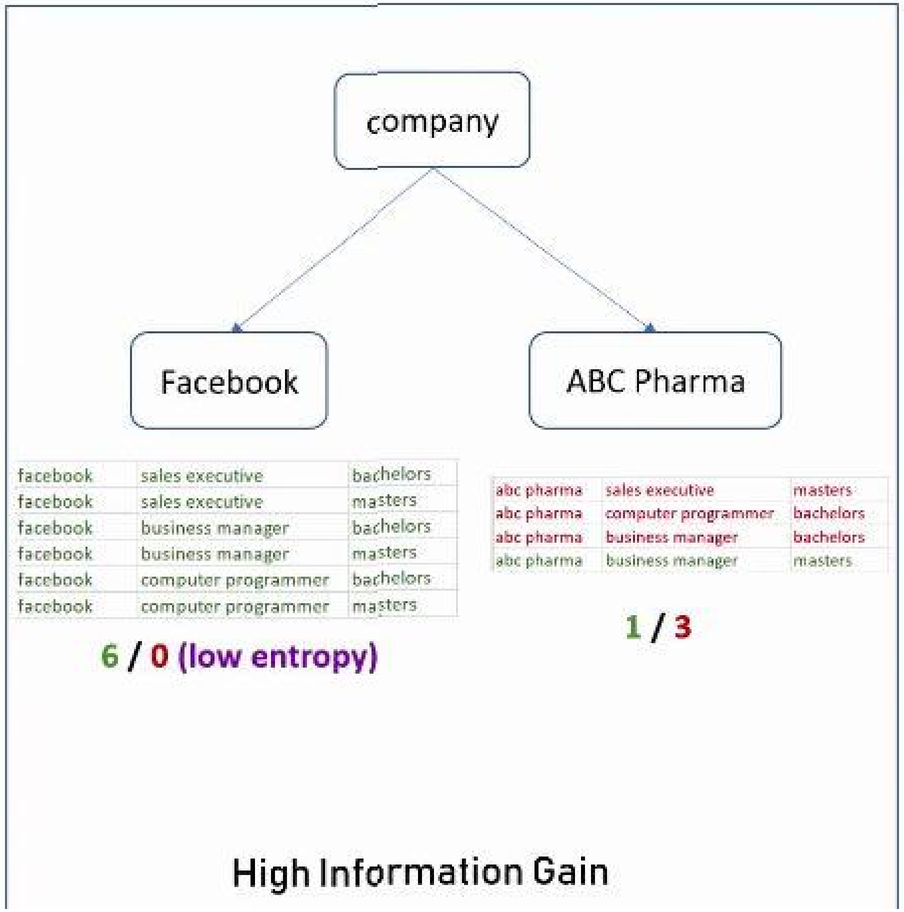


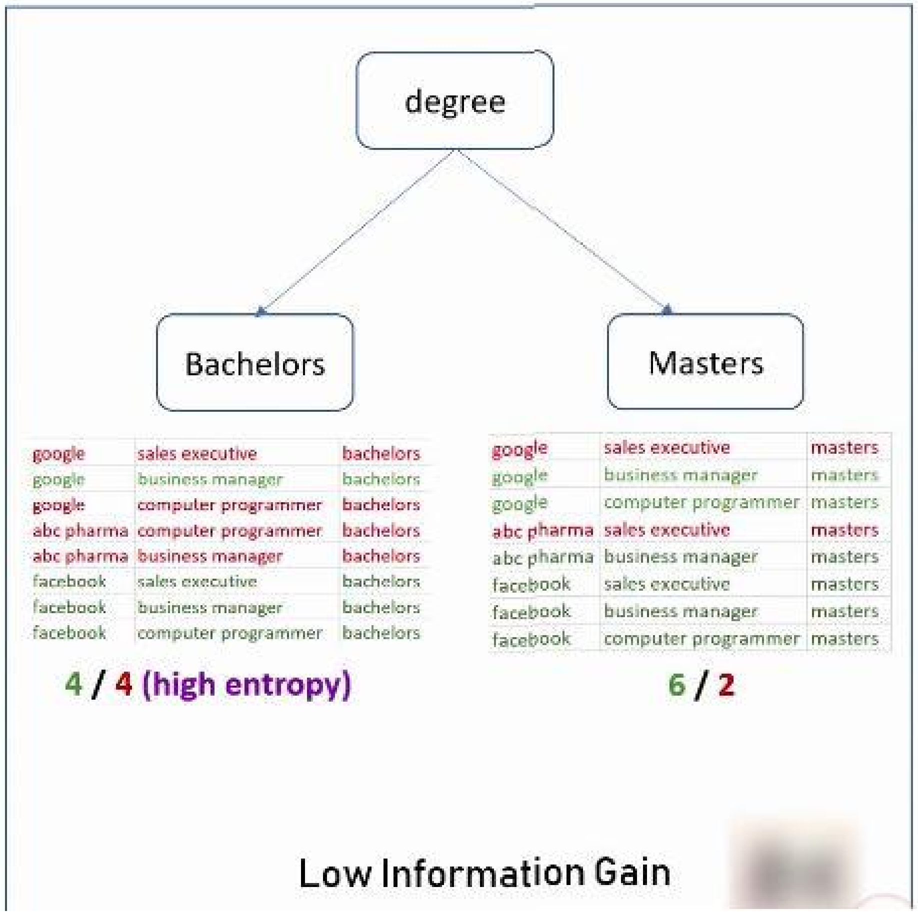
How to select ordering of features ?

Decision Tree

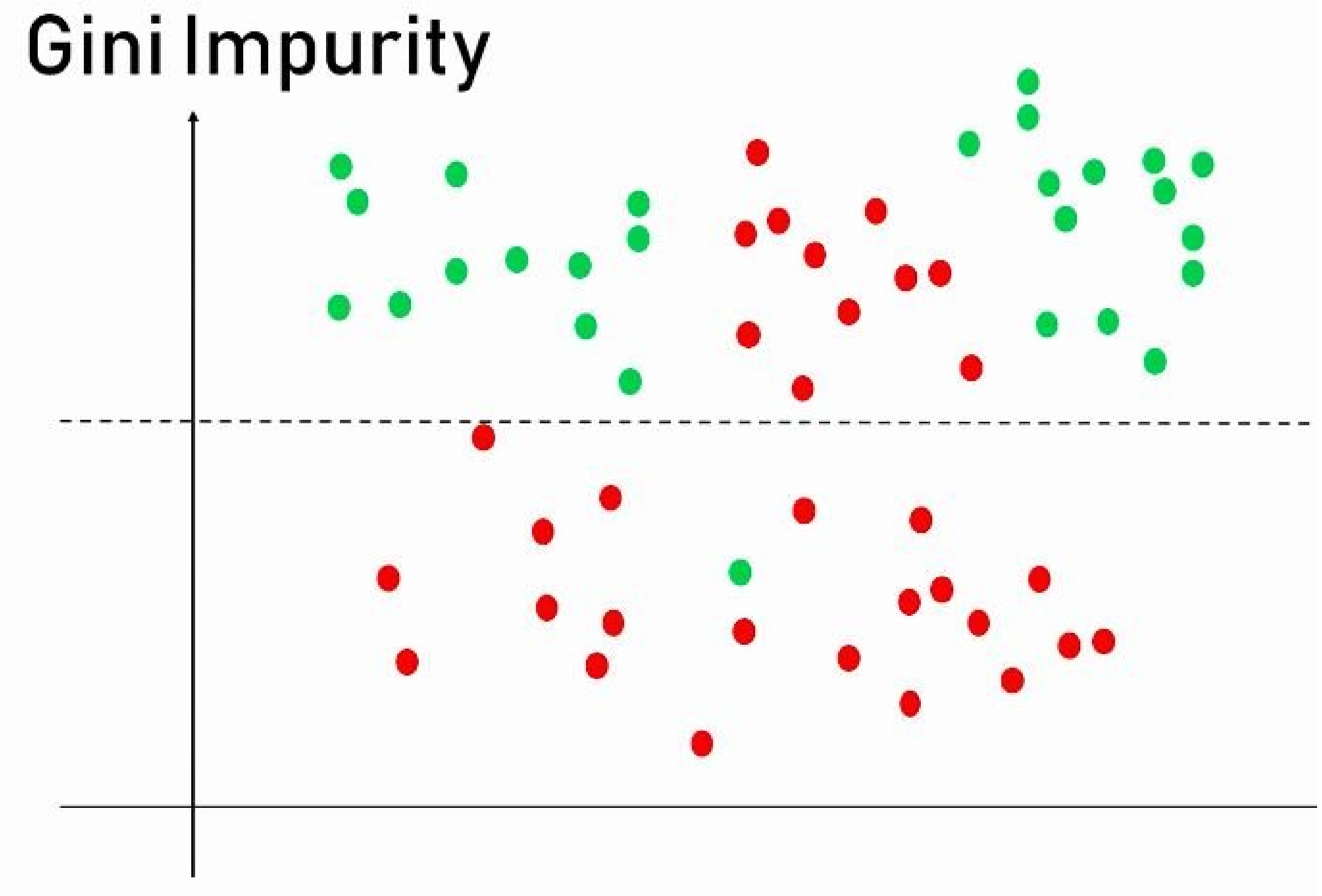


Decision Tree





Decision Tree

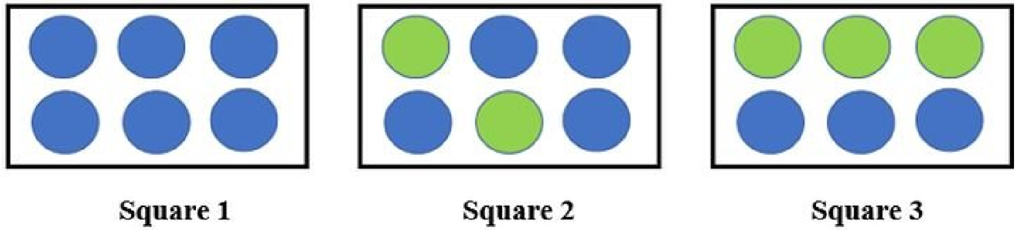


Decision Tree

- **Step-1:** Begin the tree with the root node, says S, which contains the complete dataset.
- **Step-2:** Find the best attribute in the dataset
- **Step-3:** Divide the S into subsets that contains possible values for the best attributes.
- **Step-4:** Generate the decision tree node, which contains the best attribute.
- **Step-5:** Recursively make new decision trees using the subsets of the dataset created in step -3. Continue this process until a stage is reached where you cannot further classify the nodes and called the final node as a leaf node.

What is impurity?

- For understanding decision tree algorithm better, let us first understand "impurity" and types of measures of impurity.
- Let us understand impurity from the below image
- So, there are two round objects of blue and green color inside a square and there are 3 squares. So now how much information I need in each square to accurately identify the color of round objects. So, square 1 needs less information as all objects are blue, square 2 needs little more information than square 1 to tell accurately the color of the object and square 3 requires maximum information as both blue and green objects are equal in number.
- As information is a measure of purity, square 1 is a pure node, square 2 is less impure and square 3 is more impure.
- So how to measure impurity in data?
- .



So how to measure impurity in data?

- Entropy
- Gini index/ Gini impurity
- Standard deviation

Entropy

- Entropy is the amount of information needed to accurately describe data. So, if data is homogenous that is all elements are similar then entropy is 0 (that is pure), else if elements are equally divided then entropy move towards 1 (that is impure).
- So, square 1 has the lowest entropy, square 2 has more entropy and square 3 has the highest entropy.
- Mathematically it is written as:

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

Gini index/Gini impurity

- It measures impurity in the node. It has a value between 0 and 1. So the Gini index of value 0 means sample are perfectly homogeneous and all elements are similar, whereas, Gini index of value 1 means maximal inequality among elements. It is sum of the square of the probabilities of each class. It is illustrated as,

$$Gini\,index = 1 - \sum_{i=1}^{n} p_i^2$$

- So, (Im)purity measures homogeneity in data and if data is homogenous then it belongs to the same class and decision tree splits on homogeneity.

Decision tree types

- There are various algorithm that used to generate decision tree, some are as following
- ID3 (Iterative Dichotomiser 3)
- ID 4.5 (successor of ID3)
- CART (Classification and Regression Tree)
- CHAID (Chi-squared Automatic Interaction Detector)

CART

- It is used for both classification and regression. Let us understand classification (CA) and regression tree (RT)
- Classification tree
  - A decision tree where target variable is categorical
  - The algorithm classifies the class within which the target variable would most likely to fall
  - It uses Gini index as metric/cost function to evaluate split in feature selection
  - Example like predicting who will or who will not order food, whether the weather will be rainy, sunny or cool
- 2. Regression tree
  - A decision tree where the target variable is continuous/discrete
  - Algorithm predicts value
  - It uses least square / standard deviation reduction as a metric to select features in case of the Regression tree.
  - Example like predicting the price of a house, predicting the sell of crops

- Let’s start with a dataset that is hypothetical, where the target variable is whether the customer liked the food or not.

Days	Meal Type	Spicy	Cuisine	Packed	Price	Liked/Dislike
1	Breakfast	Low	Gujarati	Hot	25	No
2	Breakfast	Low	Gujarati	cold	30	No
3	Lunch	Low	Gujarati	Hot	46	Yes
4	Dinner	normal	Gujarati	Hot	45	Yes
5	Dinner	High	South Indian	Hot	52	Yes
6	Dinner	High	South Indian	cold	23	No
7	Lunch	High	South Indian	cold	43	Yes
8	Breakfast	normal	Gujarati	Hot	35	No
9	Breakfast	High	South Indian	Hot	38	Yes
10	Dinner	normal	South Indian	Hot	46	Yes
11	Breakfast	normal	South Indian	cold	48	Yes
12	Lunch	normal	Gujarati	cold	52	Yes
13	Lunch	Low	South Indian	Hot	44	Yes
14	Dinner	normal	Gujarati	cold	30	No

- From the above data, Meal type, Spicy, Cuisine and packed are the inputs/features of data and liked/dislike is the target variable.

- Now let's start building tree having Gini index as im(purity) measure.
- **Meal Type**
- Meal Type is a nominal data that has 3 values Breakfast, Lunch and Dinner. Let's classify the instances on basis of liked/dislike.

Meal Type	# Yes	# No	# Total
Breakfast	2	3	5
Lunch	4	0	4
Dinner	3	2	5

- Gini index (Meal Type = Breakfast) = $1-(2/5)^2+(3/5)^2 = 1-(0.16+0.36) = 0.48$
- Gini index (Meal Type = Lunch) = $1-(4/4)^2+(0/4)^2 = 1-(1+0) = 0$
- Gini index (Meal Type = Dinner) = $1-(3/5)^2+(2/5)^2 = 1-(0.36+0.16) = 0.48$
- Now, we will calculate the weighted sum of Gini index for Meal Type features,
- Gini (Meal Type) = $(5/14) * 0.48 + (4/14) * 0 + (5/14) * 0.48 = 0.342$

- **Spicy**
- Spicy is a nominal data that has 3 values Low, Normal and High. Let's classify the instances on basis of liked/dislike.

Spicy	# Yes	# No	# Total
Low	2	2	4
High	3	1	4
Normal	4	2	6

- Gini (Spicy = Low) = $1-(2/4)^2+(2/4)^2 = 0.5$
- Gini (Spicy = High) = $1-(3/4)^2+(1/4)^2 = 0.375$
- Gini (Spicy = Normal) = $1-(4/6)^2+(2/6)^2 = 0.445$
- Now, the weighted sum of Gini index for Spicy features can be calculated as,
- Gini (Spicy)= $(4/14) *0.5 + (4/14) *0.375 + (6/14) *0.445 =0.439$

- **Cuisine**
- The cuisine is a binary data that has 2 values Gujarati and south Indian. Let's classify the instances on the basis of liked/dislike.

Cuisine	# Yes	# No	# Total
Gujarati	3	4	7
south Indian	6	1	7

- Gini (Cuisine = Gujarati) = $1-(3/7)^2+(4/7)^2 = 0.489$
- Gini (Cuisine =south Indian) = $1-(6/7)^2+(1/7)^2 = 0.244$
- Now, the weighted sum of the Gini index for Cuisine features can be calculated as,
- Gini (Cuisine) = (7/14) *0.489 + (7/14) *0.244=0.367

- **Packed**
- Packed is a binary data that has 2 values Hot and cold. Let's classify the instances on the basis of liked/dislike.

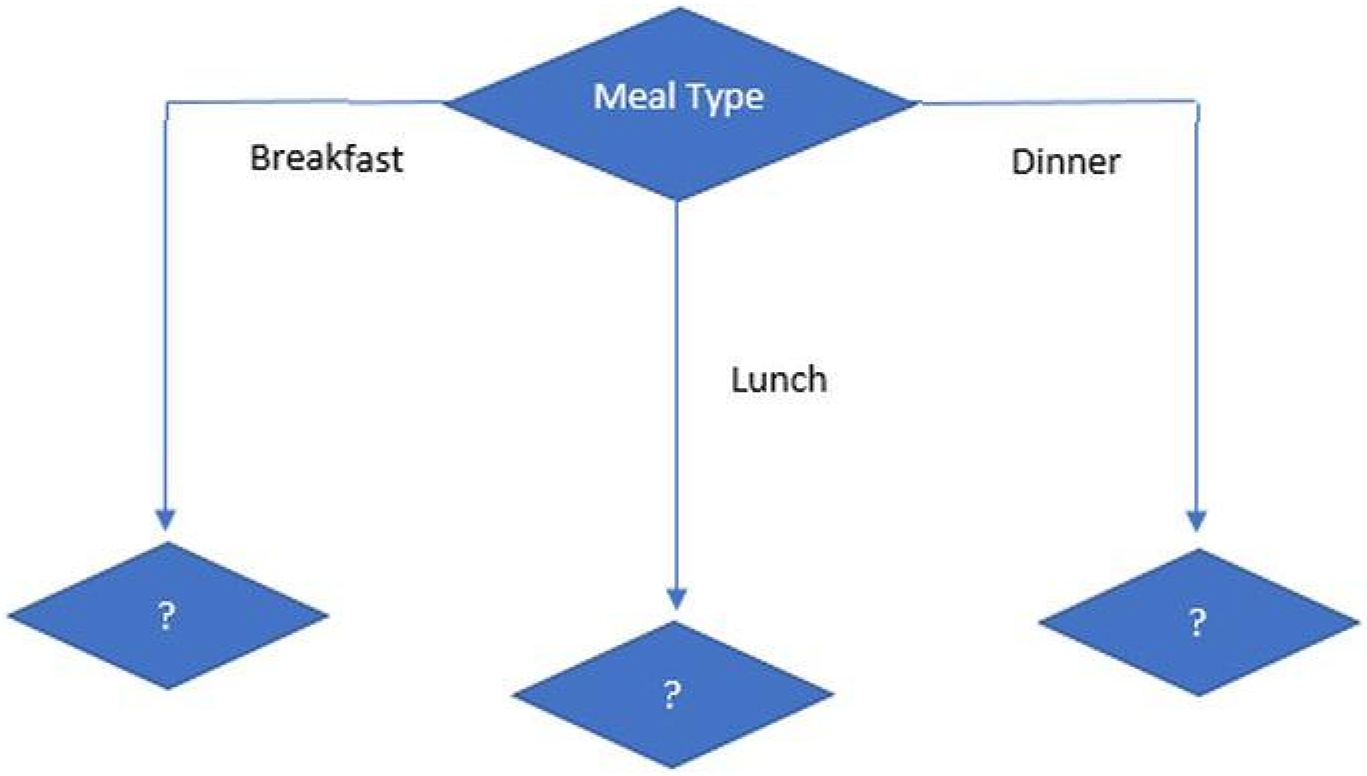
Packed	# Yes	# No	# Total
Hot	6	2	8
Cold	3	3	6

- Gini (Packed = Hot) = $1-(6/8)^2+(2/8)^2 = 0.375$
- Gini (Packed = Cold) = $1-(3/6)^2+(3/6)^2= 0.5$
- Now, the weighted sum of the Gini index for Packed features can be calculated as,
- Gini (Packed) = $(8/14) *0.375 + (6/14) *0.5=0.428$

- So, the Gini index for all the feature is:

Features	Gini Index
Meal type	0.342
Spicy	0.439
Cuisine	0.367
Packed	0.428

- So, we can conclude that the lowest Gini index is of “Meal Type” and a lower Gini index means the highest purity and more homogeneity. So, our root node is “Meal type”. So, our tree looks like.
-


Meal Type
Breakfast
Dinner
Lunch
?
?
?

- Let’s calculate the next split with the Gini index on the sub data set for the Meal Type feature, we will use the same method as above to find the next split.
- **Let’s find the Gini index of spicy, cuisine and packed on sub-data of Meal type = Breakfast.**

Days	Meal Type	Spicy	Cuisine	Packed	Price	Liked/Dislike
1	Breakfast	Low	Gujarati	Hot	25	No
2	Breakfast	Low	Gujarati	cold	30	No
8	Breakfast	normal	Gujarati	Hot	35	No
9	Breakfast	High	South Indian	Hot	38	Yes
11	Breakfast	normal	South Indian	cold	48	Yes

- Gini index for Spicy on breakfast meal type

Spicy	# Yes	# No	# Total
Low	0	2	2
Normal	1	1	2
High	1	0	1

- Gini (Meal type = Breakfast & Spicy = Low) = $1-(0/2)^2+(2/2)^2 = 0$
- Gini (Meal type = Breakfast & Spicy = High) = $1-(1/1)^2+(0/1)^2 = 0$
- Gini (Meal type = Breakfast & Spicy = Normal) = $1-(1/2)^2+(1/2)^2 = 0.5$
- Now, the weighted sum of Gini index for temperature on sunny outlook features can be calculated as,
- Gini (Meal type = Breakfast & Spicy) = (2/5) *0 + (1/5) *0+ (2/5) *0.5 =0.2

- Gini index for cuisine on breakfast meal type

Cuisine	# Yes	# No	# Total
Gujarati	0	3	3
South Indian	2	0	2

- Gini (Meal type = Breakfast & Cuisine = Gujarati) = $1-(0/3)^2+(3/3)^2 = 0$
- Gini (Meal type = Breakfast & Cuisine = South Indian) = $1-(2/2)^2+(0/2)^2 = 0$
- Now, the weighted sum of Gini index for humidity on sunny outlook features can be calculated as,
- Gini (Meal type = Breakfast & Cuisine) = $(3/5) *0 + (2/5) *0=0$

- Gini index for packed on breakfast meal type

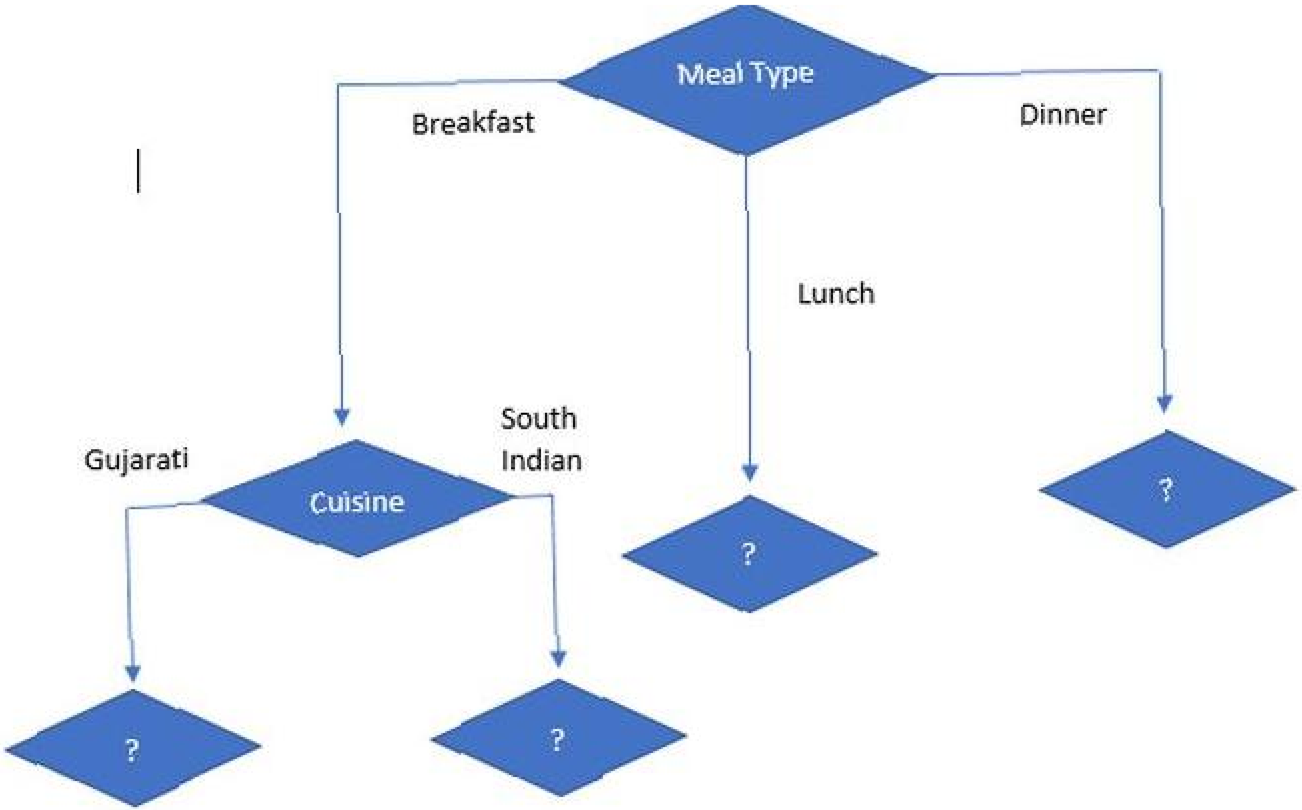
Packed	# Yes	# No	# Total
Hot	1	2	3
Cold	1	1	2

- Gini (Meal type = Breakfast & Packed = hot) = $1-(1/3)^2+(2/3)^2 = 0.44$
- Gini (Meal type = Breakfast & Packed = cold) = $1-(1/2)^2+(1/2)^2 = 0.5$
- Now, the weighted sum of Gini index for wind on sunny outlook features can be calculated as,
- Gini (Meal type = Breakfast and Packed) = $(3/5) *0.44 + (2/5) *0.5=0.266+0.2= 0.466$

- According to the Gini index, Decision on Breakfast Meal Type is:

Features	Gini index
Spicy	0.2
Cuisine	0
Packed	0.466

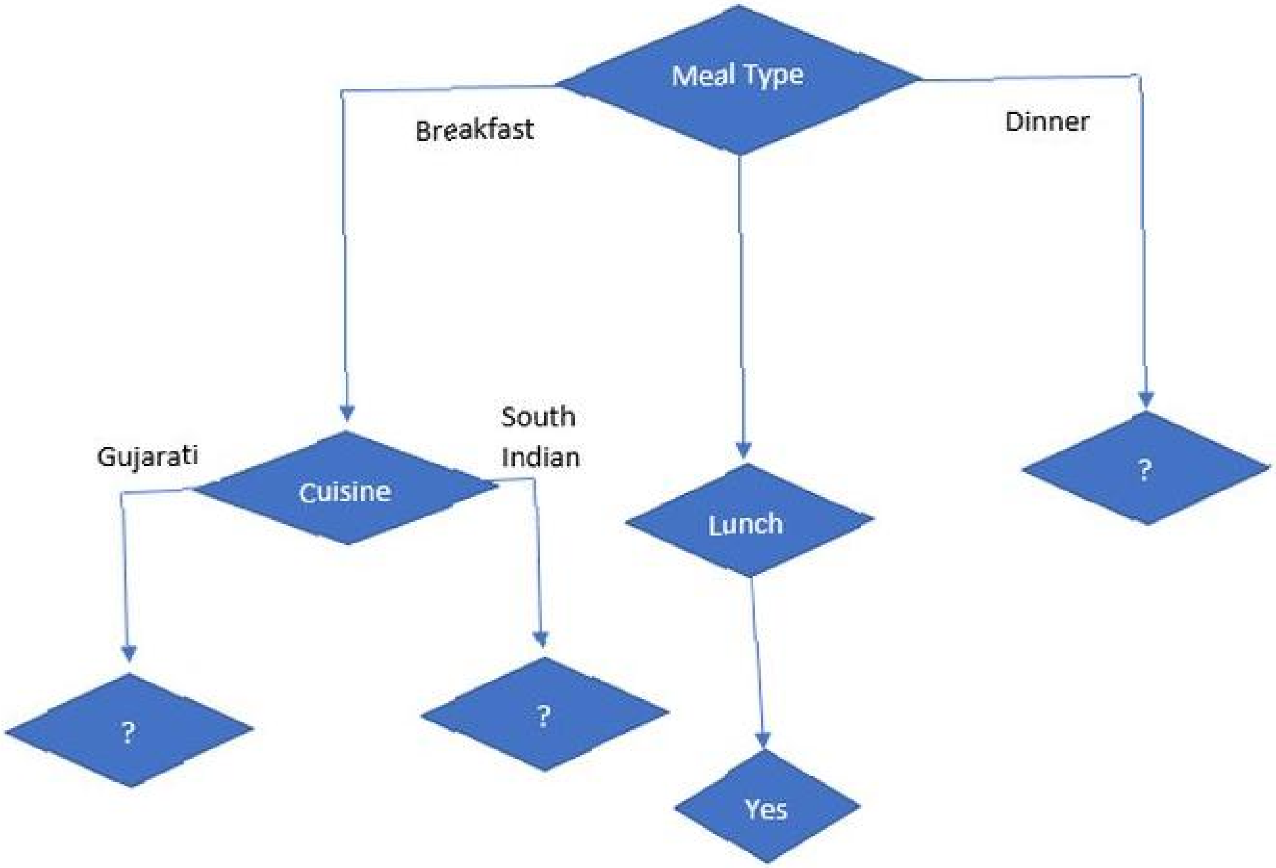
- As we can see for the breakfast meal type, the cuisine has the lowest Gini value that is highly homogenous and highest pure amongst other features, so we can conclude that the next node will be cuisine. So, the tree will be like:


Meal Type
Breakfast
Lunch
Dinner
Cuisine
Gujarati
South Indian
?
?
?
?

- Now let’s focus on sub-data of Meal Type = Lunch

Days	Meal Type	Spicy	Cuisine	Packed	Price	Liked/Dislike
3	Lunch	Low	Gujarati	Hot	46	Yes
7	Lunch	High	South Indian	cold	43	Yes
12	Lunch	normal	Gujarati	cold	52	Yes
13	Lunch	Low	South Indian	Hot	44	Yes

- As we can see for Meal Type = Lunch, the target variable is “Yes” for all so the Gini index is 0 that is there is no impurity and it is highly homogenous. So, it’s a leaf node.


Meal Type
Breakfast
Dinner
Gujarati
Cuisine
South
Indian
Lunch
?
?
?
Yes

- Now, let’s focus on Meal Type = Dinner and find the Gini index for spicy, cuisine and packed.

Days	Meal Type	Spicy	Cuisine	Packed	Price	Liked/Dislike
4	Dinner	normal	Gujarati	Hot	45	Yes
5	Dinner	High	South Indian	Hot	52	Yes
6	Dinner	High	South Indian	cold	23	No
10	Dinner	normal	South Indian	Hot	46	Yes
14	Dinner	normal	Gujarati	cold	30	No

- Gini index for spicy on meal type = Dinner

Spicy	# Yes	# No	# Total
Normal	2	1	3
High	1	1	2

- Gini (meal type = Dinner and Spicy= High) = 1 – (1/2)2 + (1/2)2 = 0.5
- Gini (meal type = Dinner and Spicy = Normal) = 1 – (2/3)2 + (1/3)2 = 0.444
- Gini (meal type = Dinner and Spicy) = (2/5) *0.5 + (3/5) *0.444 = 0.466

- Gini index for cuisine on meal type = Dinner

Cuisine	# Yes	# No	# Total
South Indian	2	1	3
Gujarati	1	1	2

- Gini (meal type = Dinner and Cuisine = Gujarati) = 1 – (1/2)2 + (1/2)2 = 0.5
- Gini (meal type = Dinner and Cuisine = South Indian) = 1 – (2/3)2 + (1/3)2 = 0.444
- Gini (meal type = Dinner and Cuisine) = (2/5) *0.5 + (3/5) *0.444 = 0.466

- Gini index for Packed on meal type = Dinner

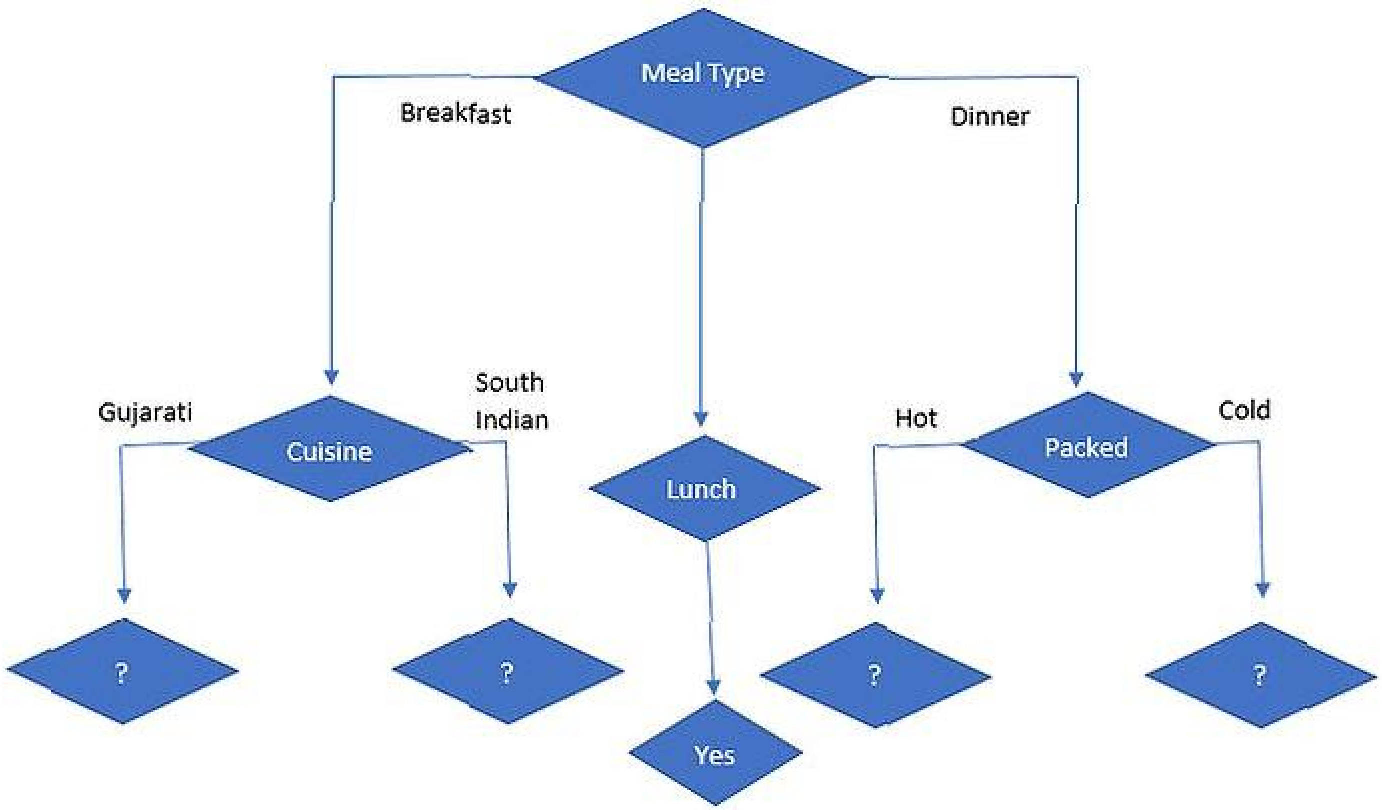
Packed	# Yes	# No	# Total
Hot	3	0	3
Cold	0	2	2

- Gini (meal type = Dinner and Packed = Hot) = 1 – (3/3)2 + (0/3)2 = 0
- Gini (meal type = Dinner and Packed = Cold) = 1 – (0/2)2 + (2/2)2 = 0
- Gini (meal type = Dinner and Packed) = (3/5) *0 + (2/5) *0 = 0

- The decision on Meal Type = Dinner

Features	Gini Index
Spicy	0.466
Cuisine	0.466
Packed	0

- So, packed has the lowest Gini value, so the next node will be packed and the following is a decision tree.


Meal Type
Breakfast
Dinner
Gujarati
Cuisine
South Indian
Lunch
Hot
Packed
Cold
?
?
?
?
Yes

- **Now, let’s focus on sub-data of:**
- 1. Cuisine
    - Gujarati
    - South Indian
- 2. Packed
    - Hot
    - Cold

- First, we will focus on Meal Type= Breakfast and Gujarati and south Indian cuisine

Days	Meal Type	Spicy	Cuisine	Packed	Price	Liked/Dislike
1	Breakfast	Low	Gujarati	Hot	25	No
2	Breakfast	Low	Gujarati	cold	30	No
8	Breakfast	normal	Gujarati	Hot	35	No
9	Breakfast	High	South Indian	Hot	38	Yes
11	Breakfast	normal	South Indian	cold	48	Yes

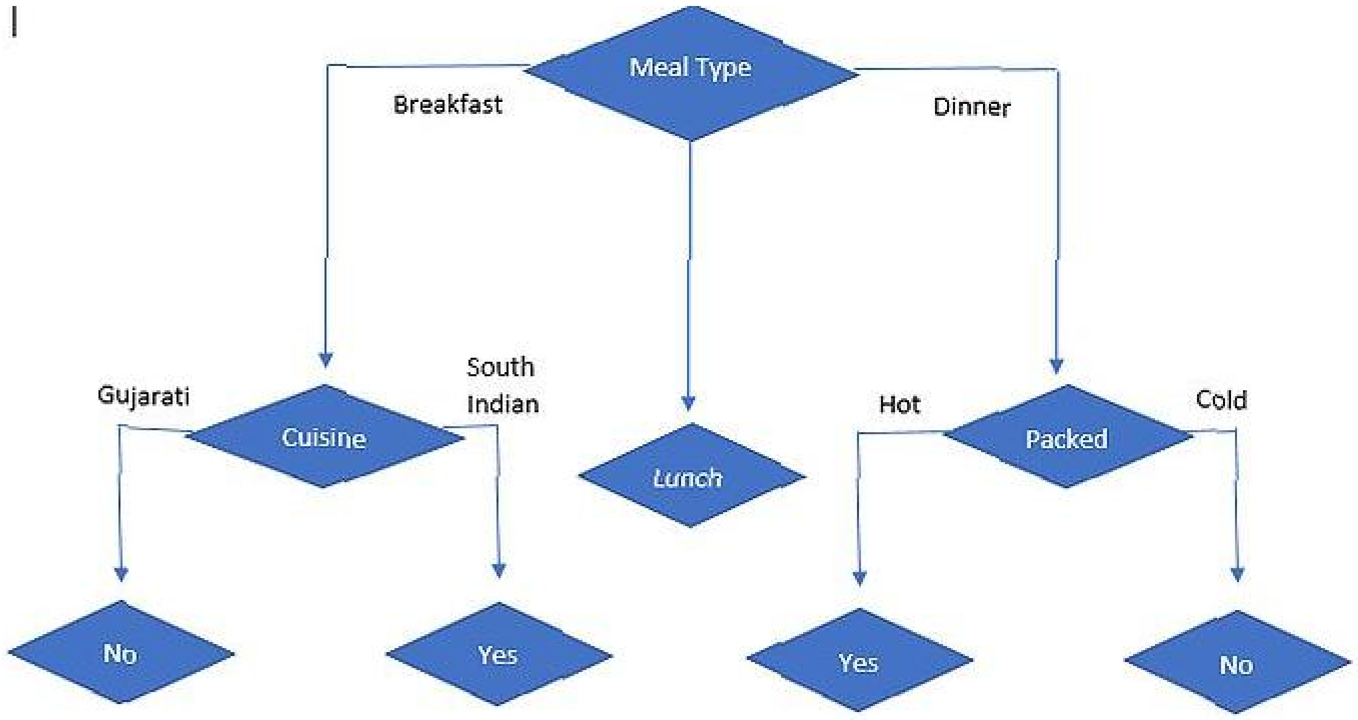
- As we can see that when Meal Type = Breakfast and Cuisine = Gujarati then the decision is always No
- And when Meal Type = Breakfast and Cuisine = South Indian then the decision is always Yes.
- So we got the leaf nodes.

- Now we will focus on Meal Type= Dinner and hot and cold packed

Days	Meal Type	Spicy	Cuisine	Packed	Price	Liked/Dislike
4	Dinner	normal	Gujarati	Hot	45	Yes
5	Dinner	High	South Indian	Hot	52	Yes
6	Dinner	High	South Indian	cold	23	No
10	Dinner	normal	South Indian	Hot	46	Yes
14	Dinner	normal	Gujarati	cold	30	No

- As we can see that when Meal Type = Dinner and Packed = Hot then the decision is always Yes
- And when Meal Type = Dinner and Packed = Cold then the decision is always No.
- So, we got the leaf nodes.

The following is our final classification decision tree:



Entropy & Information Gain

Entropy

Day	Outlook	Temperature	Humidity	Wind	Play Tennis
1	Sunny	Hot	High	Weak	No
2	Sunny	Hot	High	Strong	No
3	Overcast	Hot	High	Weak	Yes
4	Rain	Mild	High	Weak	Yes
5	Rain	Cool	Normal	Weak	Yes
6	Rain	Cool	Normal	Strong	No
7	Overcast	Cool	Normal	Strong	Yes
8	Sunny	Mild	High	Weak	No
9	Sunny	Cool	Normal	Weak	Yes
10	Rain	Mild	Normal	Weak	Yes
11	Sunny	Mild	Normal	Strong	Yes
12	Overcast	Mild	High	Strong	Yes
13	Overcast	Hot	Normal	Weak	Yes
14	Rain	Mild	High	Strong	No

Entropy

- To Define Information Gain precisely, we begin by defining a measure which is commonly used in information theory called Entropy. Entropy basically tells us how impure a collection of data is. The term impure here defines non-homogeneity. In other word we can say, “Entropy is the measurement of homogeneity. It returns us the information about an arbitrary dataset that how impure/non-homogeneous the data set is.”
  Given a collection of examples/dataset S, containing positive and negative examples of some target concept, the entropy of S relative to this boolean classification is-

$$Entropy(S) = -(P_{\oplus} log_2 P_{\oplus} + P_{\ominus} log_2 P_{\ominus}) \tag{1.1}$$

Where $P_{\oplus}$ is the portion of positive examples and $P_{\ominus}$ is the portion of negative examples in S.

- The dataset has 9 positive instances and 5 negative instances, therefore-

$$Entropy([9+,5-]) = -(\frac{9}{14}log_2\frac{9}{14} + \frac{5}{14}log_2\frac{5}{14}) = 0.940 \qquad (1.2)$$

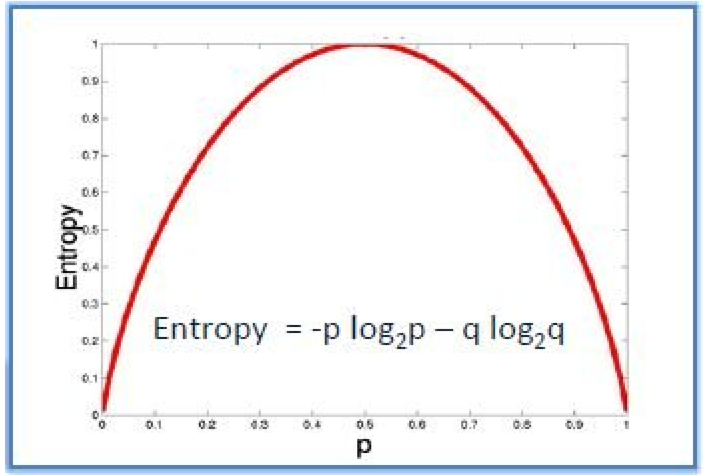
Which concludes, the dataset is 94% impure or 94% non-homogeneous.

Let's do some more calculations and try to understand the nature of *Entropy*. What could be the Entropy of [7+,7-] & [14+,0-]?

$$Entropy[7+,7-] = -(\frac{7}{14}log_2\frac{7}{14} + \frac{7}{14}log_2\frac{7}{14}) = 1 \qquad (1.3)$$

And,

$$Entropy[14+,0-] = -(\frac{14}{14}log_2\frac{14}{14} + \frac{0}{14}log_2\frac{0}{14}) = 0 \qquad (1.4)$$



- Now, if we try to plot the ***Entropy*** in a graph, it will look like Figure 2. It clearly shows that the Entropy is lowest when the data set is homogeneous and highest when the data set is completely non-homogeneous.

Information Gain

- Given Entropy is the measure of impurity in a collection of a dataset, now we can measure the effectiveness of an attribute in classifying the training set. The measure we will use called ***information gain***, is simply the expected reduction in ***entropy*** caused by partitioning the data set according to this attribute. The information gain ***(Gain(S,A)*** of an attribute A relative to a collection of data set S, is defined as-

$$Gain(S, A) = Entropy(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} Entropy(S_v) \qquad (1.5)$$

Where, $Values(A)$ is the all possible values for attribute A, and S_v is the subset of S for which attribute A has value v.

- The dataset has 14 instances, so the sample space is 14 where the sample has 9 positive and 5 negative instances. The Attribute **Wind** can have the values **Weak** or **Strong**. Therefore,

$$S = [9+, 5-]$$
$$S_{\text{weak}} = [6+, 2-]$$
$$S_{\text{strong}} = [3+, 3-]$$
$$Entropy(S) = 0.940 \text{ from eqtn } 1.2$$

$$Gain(S, A) = Entropy(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} Entropy(S_v)$$

$$Gain(S, Wind) = Entropy(S) - (\frac{8}{14} Entropy(S_{\text{weak}}) + \frac{6}{14} Entropy(S_{\text{strong}})) \quad (1.6)$$

$$Entropy(S_{\text{weak}}) = -(\frac{6}{8} log_2 \frac{6}{8} + \frac{2}{8} log_2 \frac{2}{8}) = 0.811 \quad (1.7)$$

$$Entropy(S_{\text{strong}}) = -(\frac{3}{3} log_2 \frac{3}{3} + \frac{3}{3} log_2 \frac{3}{3}) = 1.00 \quad (1.8)$$

Put the values of $Entropy(S_{\text{weak}})$ and $Entropy(S_{\text{strong}})$ in eqtn 1.6

$$\begin{aligned} Gain(S, Wind) &= Entropy(S) - (\frac{8}{14} 0.811 + \frac{6}{14} 1.00) \\ &= 0.940 - (0.463 + 0.429) \\ &= 0.048 \end{aligned} \quad (1.9)$$

- So, the ***information gain*** by the **Wind** attribute is 0.048.

- Let's calculate the ***information gain*** by the **Outlook** attribute.

$$Values(Outlook) = Sunny, Overcast, Rain$$
$$S = [9+, 5-]$$
$$S_{\text{sunny}} = [2+, 3-]$$
$$S_{\text{overcast}} = [4+, 0-]$$
$$S_{\text{rain}} = [3+, 2-]$$

$$G(S, Outlook) = Entropy(S) - (\frac{5}{14}Entropy(S_{\text{sunny}}) + \frac{4}{14}Entropy(S_{\text{overcast}}) + \frac{5}{14}Entropy(S_{\text{rain}})) \tag{1.10}$$

$$Entropy(S_{\text{sunny}}) = -(\frac{2}{5}log_2\frac{2}{5} + \frac{3}{5}log_2\frac{3}{5}) = 0.971 \tag{1.11}$$

$$Entropy(S_{\text{overcast}}) = -(\frac{4}{4}log_2\frac{4}{4} + \frac{0}{4}log_2\frac{0}{4}) = 0 \tag{1.12}$$

$$Entropy(S_{\text{rain}}) = -(\frac{3}{5}log_2\frac{3}{5} + \frac{2}{5}log_2\frac{2}{5}) = 0.971 \tag{1.13}$$

Put the values of eqtn 1.11, 1.12, 1.13 in 1.10.

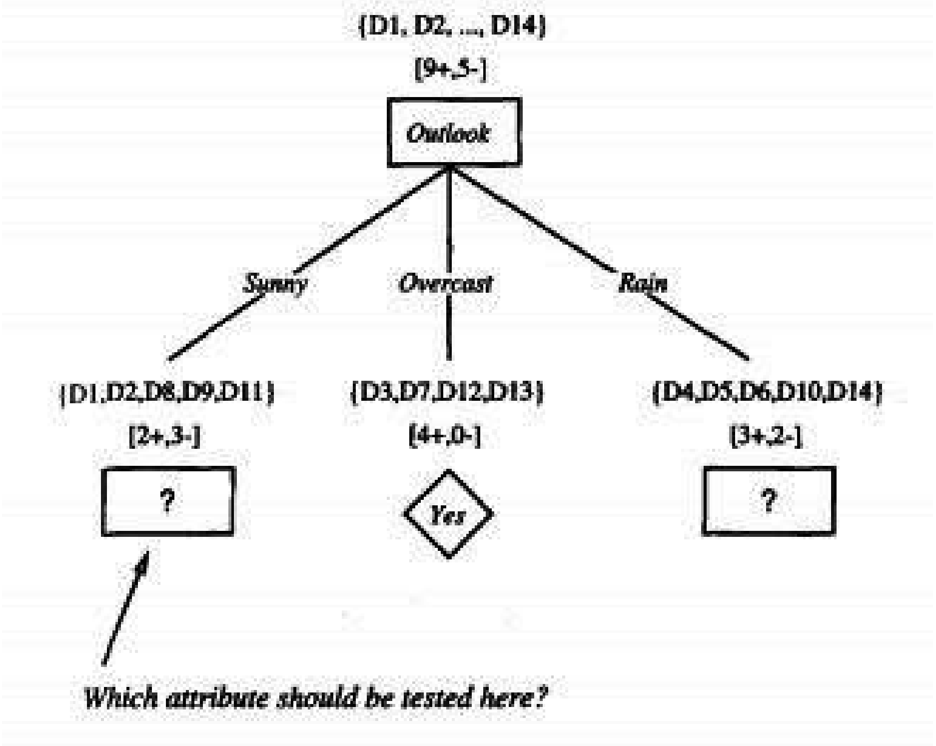
$$G(S, Outlook) = 0.940 - (\frac{5}{14}0.971 + \frac{4}{14}0 + \frac{5}{14}0.971) = 0.246 \tag{1.14}$$

- The information gain of the 4 attributes of Figure 1 dataset are:

$$Gain(S, Outlook) = 0.246$$
$$Gain(S, Humidity) = 0.151$$
$$Gain(S, Wind) = 0.048$$
$$Gain(S, Temperature) = 0.029$$

- the main goal of measuring ***information gain*** is to find the attribute which is most useful to classify training set. Our ID3 algorithm will use the attribute as it's root to build the decision tree. Then it will again calculate information gain to find the next node. As far as we calculated, the most useful attribute is "Outlook" as it is giving us more information than others. So, "Outlook" will be the root of our tree.

- The **Overcast** descendant has only positive instances and therefore becomes a leaf node with classification **Yes**. For other two nodes, the question again arises which attribute should be tested? These two nodes will be further expanded by selecting the attributes with the highest information gain **relative** to the new subset of examples.



- Let's find the attribute that should be tested at the **Sunny** descendant.
- The Dataset in Figure 1 has the value **Sunny** on Day1, Day2, Day8, Day9, Day11. So the Sample Space S=5 here.

$$S_{\text{suuny}} = 5 = S$$
$$Humidity = High, Normal$$
$$Humidity_{\text{high}} = [0+, 3-]$$
$$Humidity_{\text{normal}} = [2+, 0-]$$
$$Gain(S, Humidity) = ?$$

$$Gain(S_{\text{sunny}}, Humidity) = Entropy(S) - (\frac{3}{5}Entropy(Humidity_{\text{high}}) + \frac{2}{5}Humidity_{\text{normal}}) \tag{1.15}$$

$$Entropy(Humidity_{\text{high}}) = -(\frac{0}{3}log_2\frac{0}{3} + \frac{3}{3}log_2\frac{3}{3}) = 0 \tag{1.16}$$

$$Entropy(Humidity_{\text{normal}}) = -(\frac{2}{2}log_2\frac{2}{2} + \frac{0}{2}log_2\frac{0}{2}) = 0 \tag{1.17}$$

Put the values in eqtn 1.15

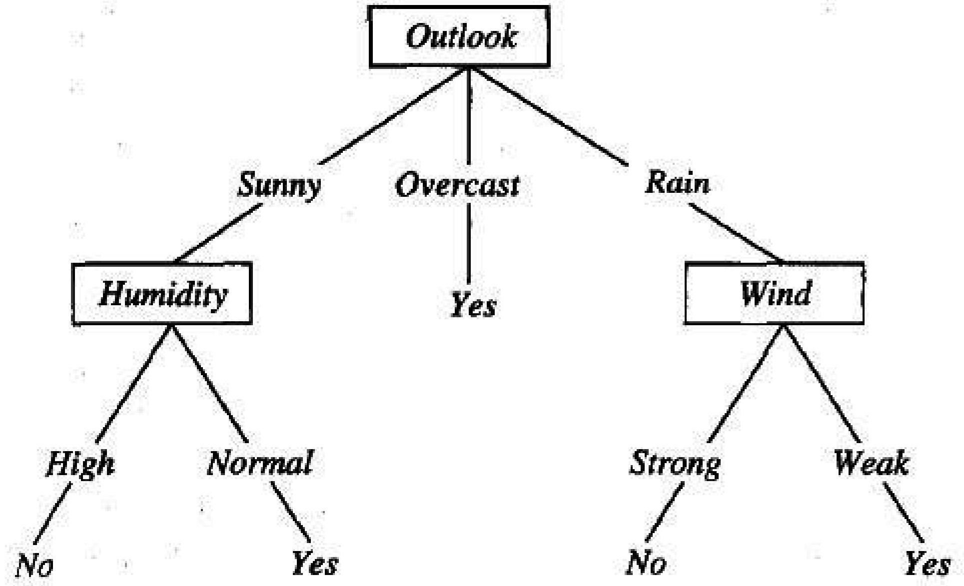
$$\begin{aligned} Gain(S_{\text{sunny}}, Humidity) &= Entropy(S) - (\frac{3}{5}0 + \frac{2}{5}0) \\ &= 0.970 - 0 \\ &= 0.970 \end{aligned} \tag{1.18}$$

- We can now measure the information gain of Temperature and Wind by following the same way we measured **Gain(S, Humidity)**. Finally, we will get:

$$Gain(S, Humidity) = 0.970$$
$$Gain(S, Temperature) \approx 0.570$$
$$Gain(S, Wind) = 0.019$$

- So Humidity gives us the most information at this stage. The node after **“Outlook”** at **Sunny** descendant will be **Humidity**.

- The **High** descendant has only negative examples and the **Normal** descendant has only positive examples. So both of them become the leaf node and can not be furthered expanded. If we expand the **Rain** descendant by the same procedure we will see that the **Wind** attribute is providing most information.



- **Inductive Bias in Decision Tree Learning:**
- The inductive bias (also known as learning bias) of a learning algorithm is the set of assumptions that the learner uses to predict outputs given inputs that it has not encountered {Tom M. Mitchell, Machine Learning}. Given a collection of examples, there could be many decision trees consistent with these examples. Which decision tree does ID3 choose? The ID3 search strategy (a) selects in favor of shorter trees over longer trees and (b) selects trees that place the attributes with the highest information gain closest to the root. Because of the subtle interaction between attribute selection heuristic used by ID3 and the particular training examples it encounters, it is difficult to characterize precisely the inductive bias exhibited by ID3. However, we can approximately characterize it's bias as a preference for ***"shorter trees over longer trees and Trees that place high information gain attributes close to the root are preferred over those that do not."***

Any Doubt ?

Please feel free to write to me:

gr83gita@gmail.com

Mobile: 9403846842

Thank You !!!