ML Part-3

Shiva Reddy

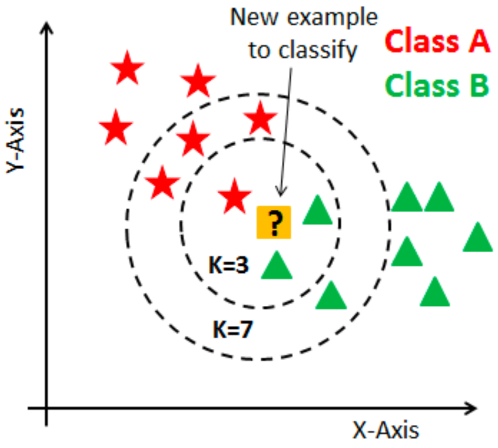
Imp criteria while choosing best ML Model ?

Scalability

Performance

Interpretability

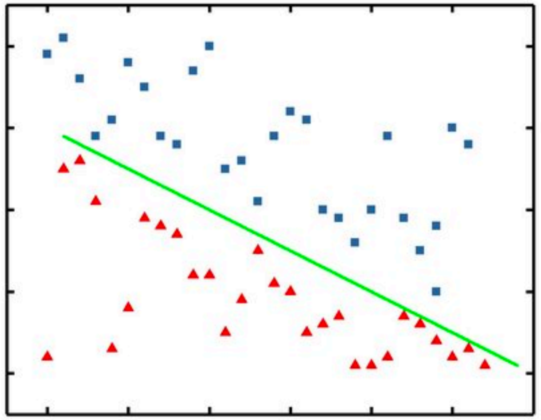
K-Nearest Neighbor



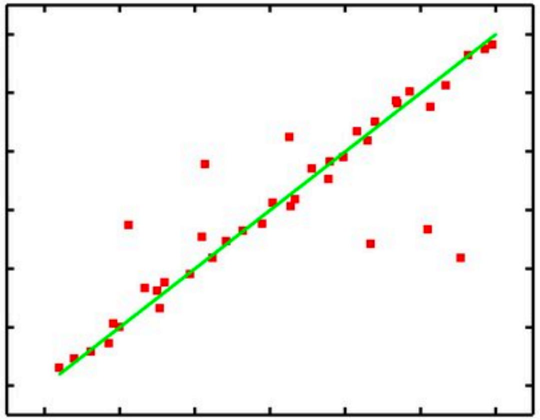
Imp Pointers on KNN :

Not Scalable

Linear Models



(a) Logistic Regression



(b) Linear Regression

Imp Pointers on Linear Models :

Decision boundary : Linear

Highly Interpretable models

E.g. Sales = W1 * Digital Media + W2 * TV + W3 * Offline Adv

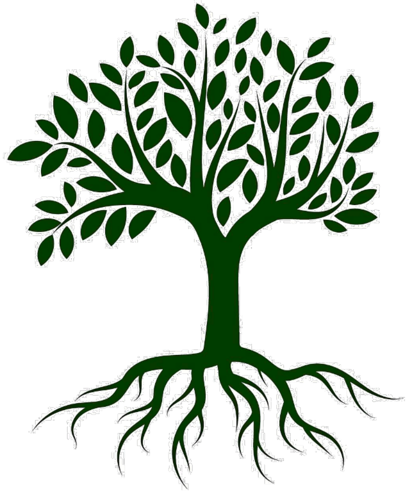
Decision Tree

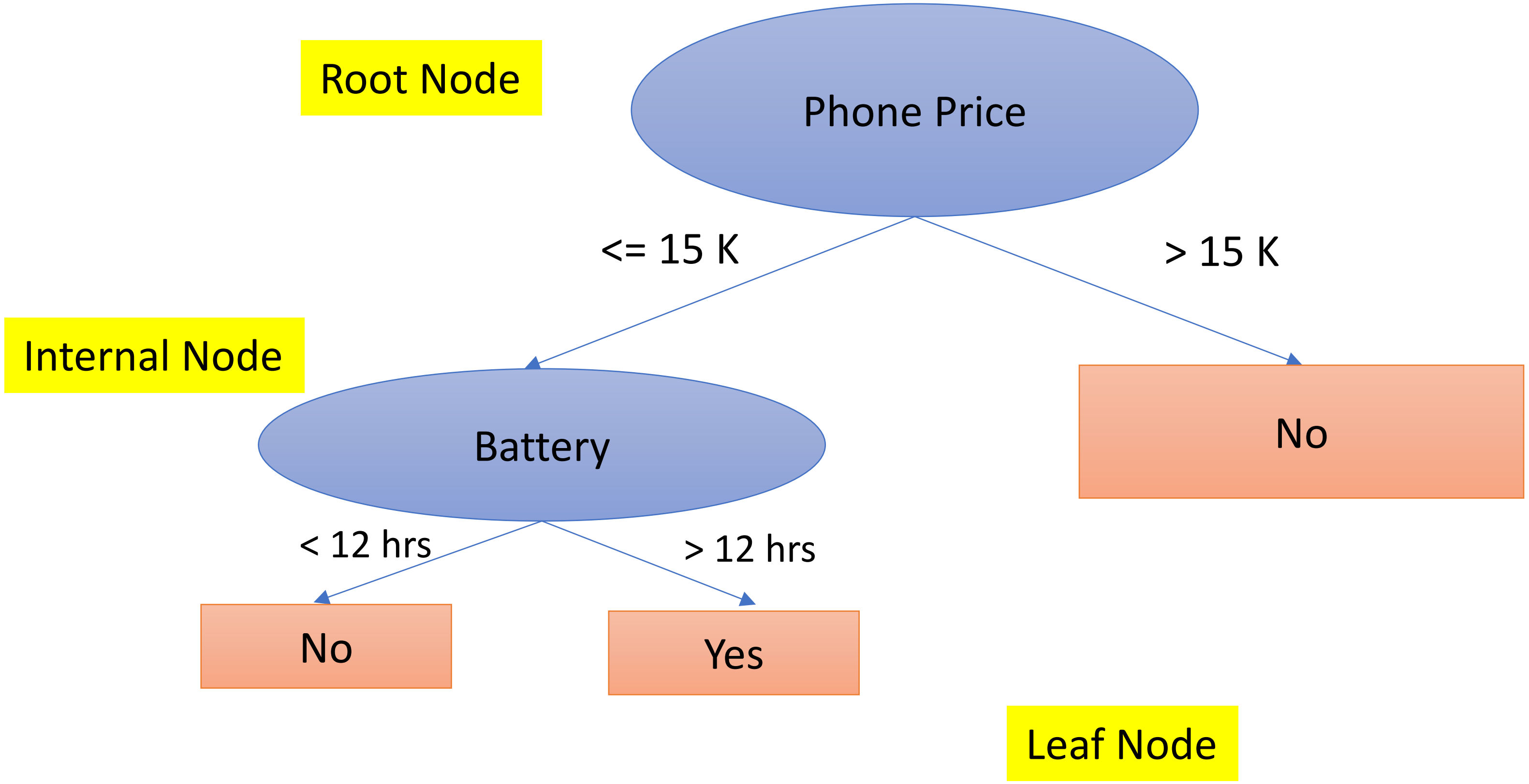
- How to take a decision of which phone to buy ?

Each path from the root of the DT to a leaf can be interpreted as a decision rule

Imp criteria:

1. Budget Phone : 15 K
2. Battery : atleast 12 hrs





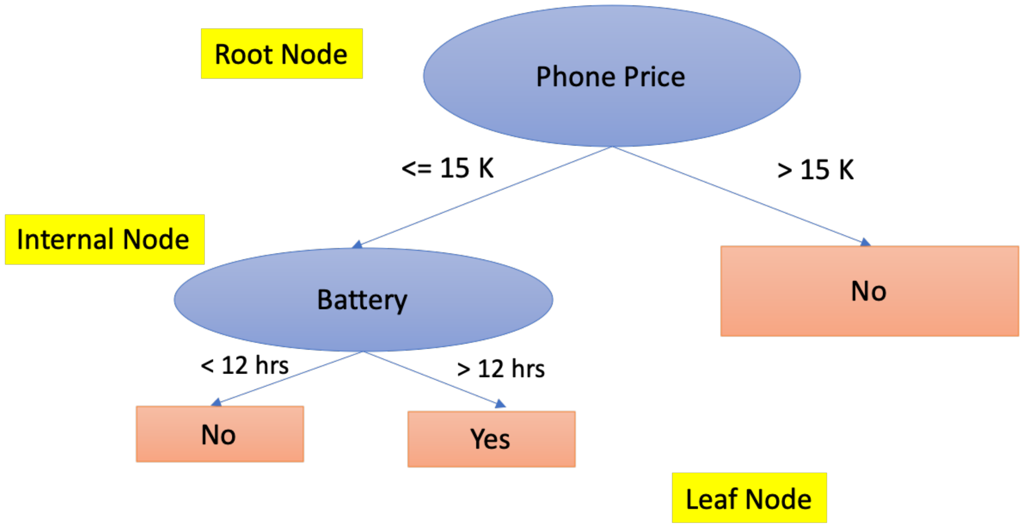
DT Terminology

- **Root Node :** This is first node while building a DT ,where all data is considered.
- **Internal Node** : All nodes after root node & before leaf node are internal Nodes

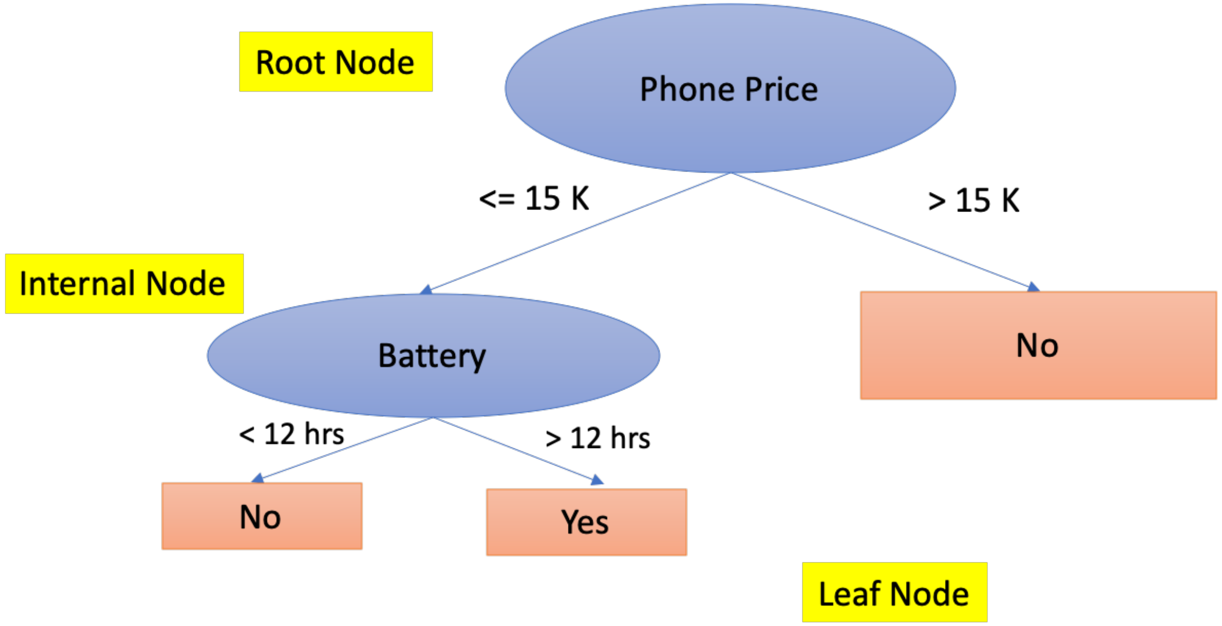
  Root Node & Internal nodes are known as '**Decision Node'.**

- **Leaf Node** : Last node is Leaf node/Terminal node.

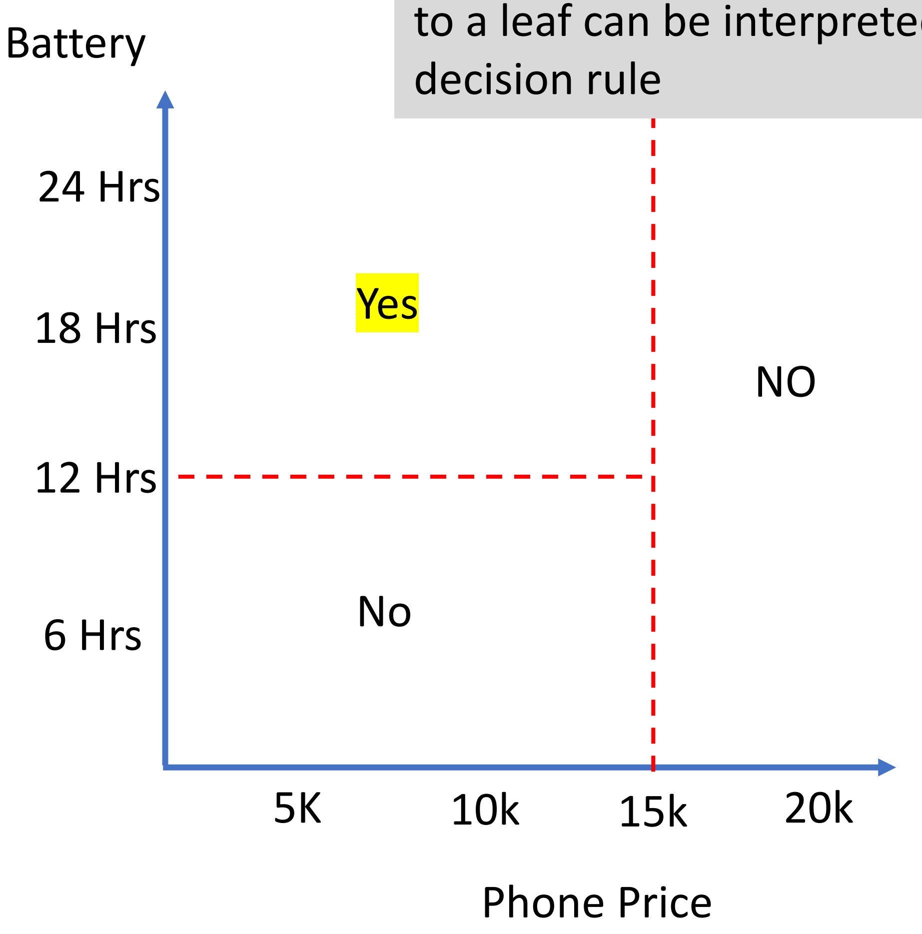
  At leaf node ,we take final decision /labelling.



Graphical representation of DT



Each path from the root of the DT to a leaf can be interpreted as a decision rule



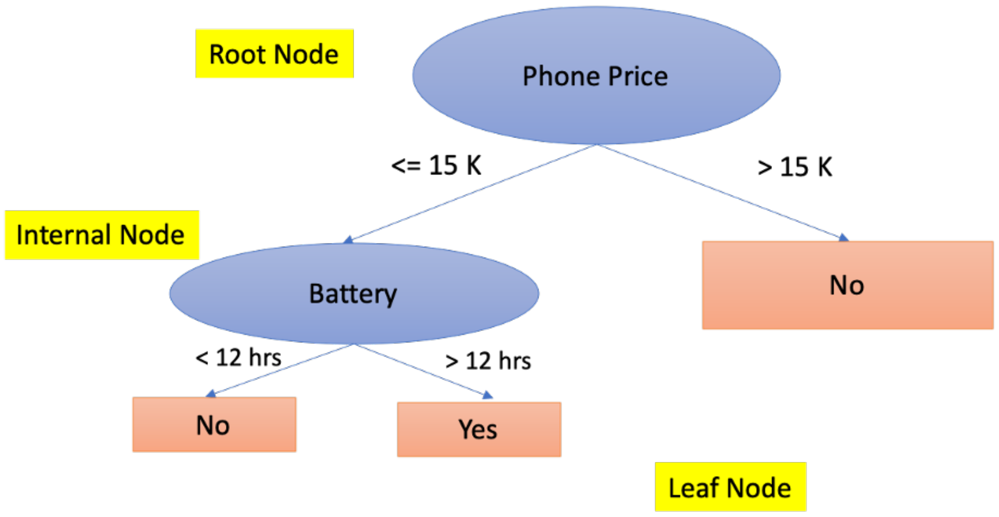
Decision Boundary :

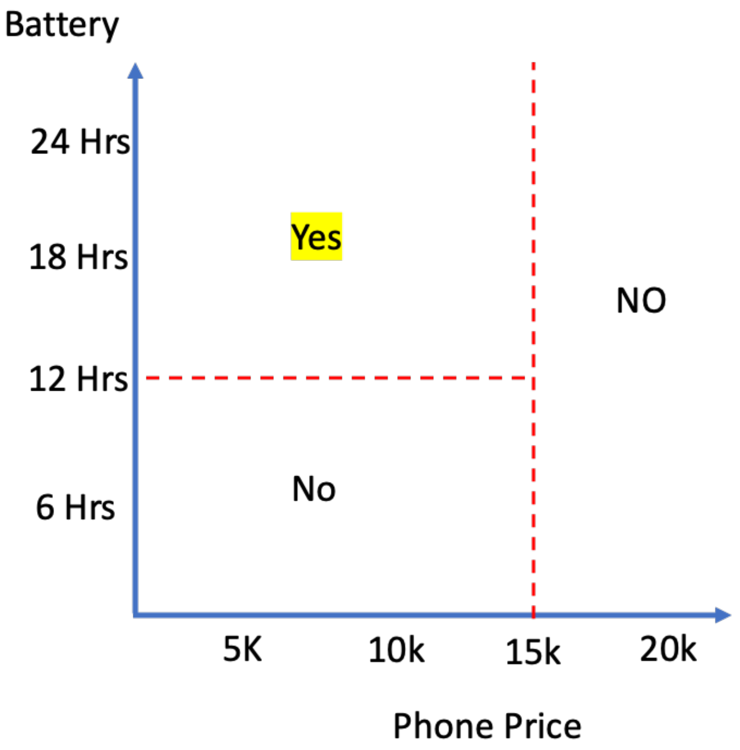
Axis Parallel (Non-linear)

Geometric: set of axis parallel hyperplane that divides your whole region into cubes ,cuboid, hypercube.

Decision Tree

- Decision tree is a tree based method that partition the feature space into a set of rectangles & then assign a constant value (mean/mode) to every region.





How to built a Decision tree ?

- To built a decision tree, we start at the tree root and split the data on the feature that results in the largest information gain (IG).

  i.e Select the feature at root node which gives Max IG

- Information Gain : is a way to measure expected reduction in Entropy

Entropy

- Entropy is a way to measure impurity/uncertainty in data.
- Low Entropy --- > Good
  - The higher the entropy, the harder it is to draw any conclusions from that information.

Win Win Win Win Win Lose Lose Lose Lose Lose

WIN: 5
Prob(win) = 5/10
= 0.5

Lose: 5
Prob(lose) = 5/10
= 0.5

Scenario	# Win	#Lost	Predict
1	5	5	Uncertain
2	10	0	More Certain
3	0	10	More certain

How to calculate Entropy

$$I_H = -\sum_{j=1}^{c} p_j log_2(p_j)$$

C : Number of Classes in the label
Pj: Count of Class j / Total count

Win Win Win Win Win Loss Loss Loss Loss Loss

WIN: 5
Prob 1 = 5/10
= 0.5

Lost: 5
Prob 2 = 5/10
= 0.5

= - ((prob_win * log 2(prob win) + (prob_lose) *log2(prob_lose))

= -((0.5) * log2 (0.5) + (0.5) * log2(0.5))

= -(-0.5 + (-0.5))

= - (-1) = 1

Entropy Max/Min

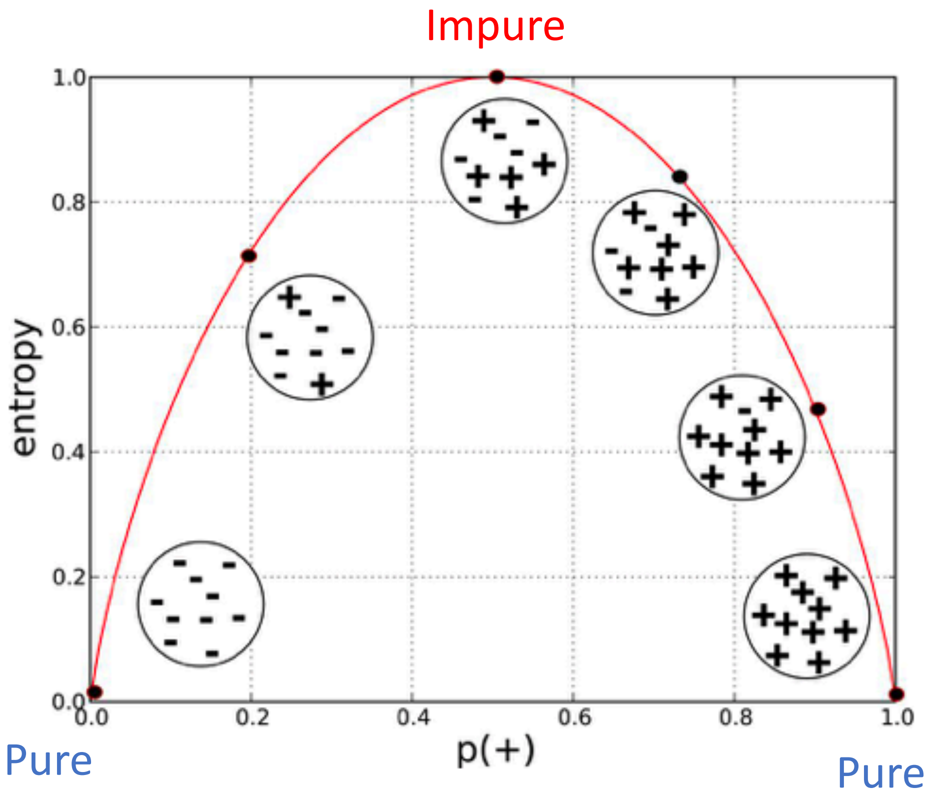
$$I_H = -\sum_{j=1}^{c} p_j log_2(p_j)$$

Scenarios	#Win	Prob_win	#Lose	Prob_lose	Entropy
1	0	0	10	1	0
2	1	0.1	9	0.9	0.46
3	2	0.2	8	0.8	0.72
4	3	0.3	7	0.7	0.88
5	4	0.4	6	0.6	0.97
6	5	0.5	5	0.5	1
7	6	0.6	4	0.4	0.97
8	7	0.7	3	0.3	0.88
9	8	0.8	2	0.2	0.72
10	9	0.9	1	0.1	0.46
11	10	1	0	0	0

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

Entropy Max/Min

Scenarios	#Win	Prob_win	#Lose	Prob_lose	Entropy
1	0	0	10	1	0
2	1	0.1	9	0.9	0.46
3	2	0.2	8	0.8	0.72
4	3	0.3	7	0.7	0.88
5	4	0.4	6	0.6	0.97
6	5	0.5	5	0.5	1
7	6	0.6	4	0.4	0.97
8	7	0.7	3	0.3	0.88
9	8	0.8	2	0.2	0.72
10	9	0.9	1	0.1	0.46
11	10	1	0	0	0



Entropy is Min

when all elements belong to one class

Entropy is Max

when all elements are equally probable

- Calculate entropy of below variable ‘Play’ :

  [Yes , Yes , Yes , Yes , Yes, No, No, Yes, No, Yes, No, Yes , Yes, No]

Count of yes	9	9/14
Count of No	5	5/14
Total	14	

Entropy. = -1 * (9/14 * log2(9/14) + 5/14* log2(5/14))

Information Gain

- Information Gain : is a way to measure expected reduction in Entropy

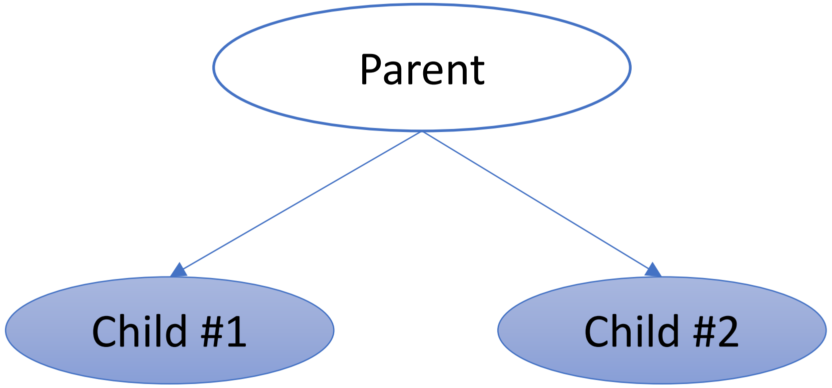
Information Gain = Parent Entropy - Weighted Average of Child Entropy

Entropy Before Split - Entropy After Split

= 0.88 – 0.68 = 0.2

#Win	70
#Lost	30
Total	**100**

Entropy before split :0.88



#Win	60
#Lost	10
Total	**70**

#Win	10
#Lost	20
Total	**30**

Entropy @ left node : 0.59

Entropy @ right node : 0.91

$$IG(D_p, x_i) = I(D_p) - \frac{N_{left}}{N_p} I(D_{left}) - \frac{N_{right}}{N_p} I(D_{right})$$

- IG: Information Gain
- x_i : feature to perform the split
- N_p: number of samples in the parent node
- N_{left} : number of samples in the left child node
- N_{right} : number of samples in the right child node
- I: impurity
- D_p: training subset of the parent node
- D_{left} : training subset of the left child node
- $D_{r}ight$: training subset of the right child node

Entropy after split = 70/100 * Entropy @ left node + 30/100 * Entropy @ right node

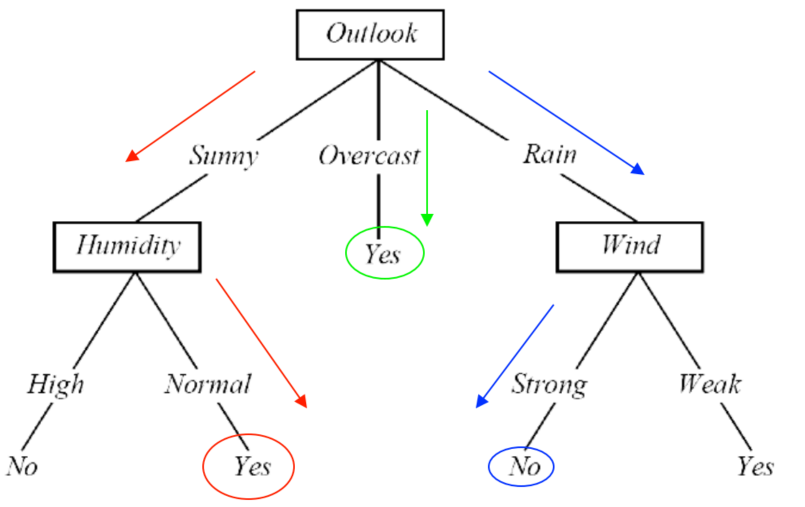
= 0.7 * 0.59 + 0.3 *0.91 = 0.68

Play Tennis Dataset

Day	Outlook	Temp.	Humidity	Wind	Play?
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

Each path from the root of the DT to a leaf can be interpreted as a decision rule.

Decision Tree



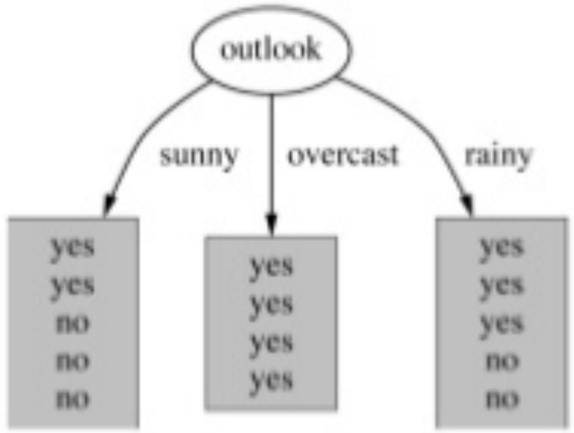
IF *Outlook* = *Sunny* AND *Humidity* = *Normal* THEN *Playtennis* = *Yes*

IF *Outlook* = *Overcast* THEN *Playtennis* =*Yes*

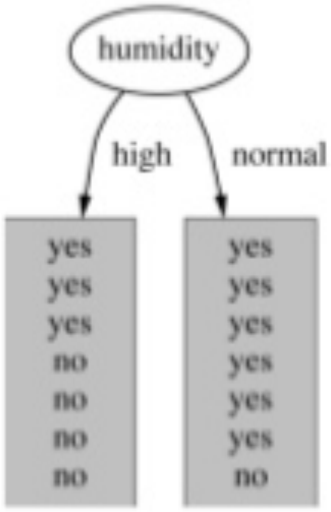
IF *Outlook* = *Rain* AND *Wind* = *Strong* THEN *Playtennis* = *No*

Which attribute to select?

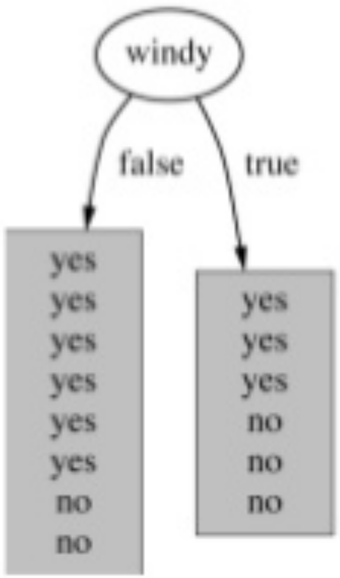
IG = 0.25



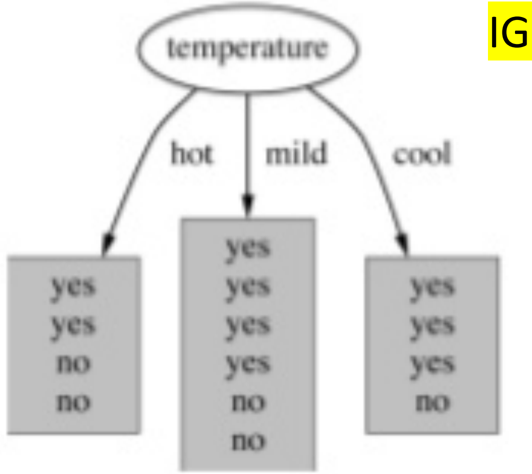
IG = 0.16



IG = 0.04



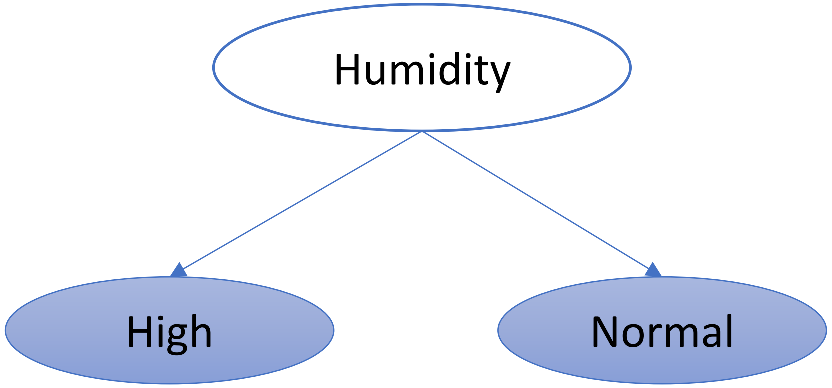
IG = 0.051



Humidity

#Yes	9
#No	5
Total	**14**

Entropy before split : 0.94



#Yes	3
#No	4
Total	**7**

#Yes	6
#No	1
Total	**7**

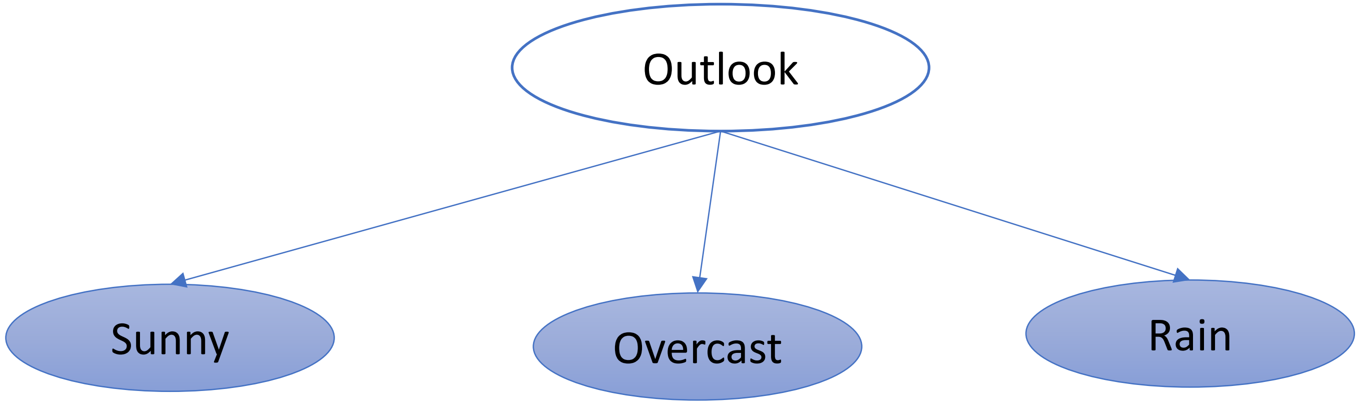
Entropy @ after split = 7/14 * Entropy @ left node : 0.98 + 7/14 * Entropy @ right node : 0.59

Entropy before spilt	Entropy after split	Information Gain
0.94	0.78	0.16

Outlook

#Yes	9
#No	5
Total	**14**

Entropy before split : 0.94



#Yes	2
#No	3
Total	**5**

#Yes	4
#No	0
Total	**4**

#Yes	3
#No	2
Total	**5**

5/14 * Entropy @ left node : 0.97

+

4/14 * Entropy @ middle node : 0

+

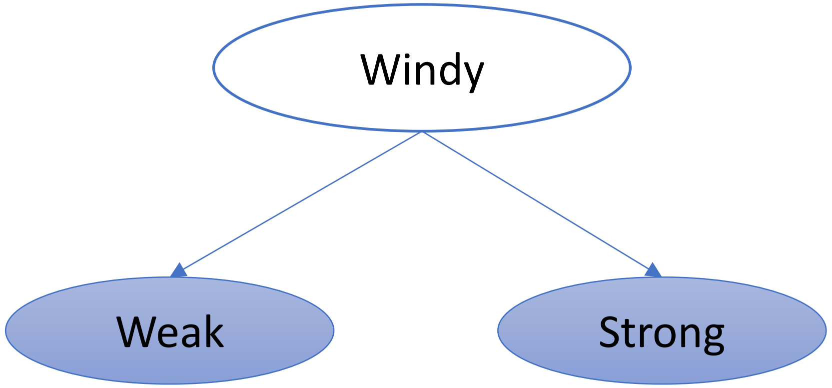
5/14 * Entropy @ right node : 0.97

Entropy before spilt	Entropy after split	Information Gain
0.94	0.69	0.25

Windy

#Yes	9
#No	5
Total	**14**

Entropy before split : 0.94



#Yes	
#No	
Total	

#Yes	
#No	
Total	

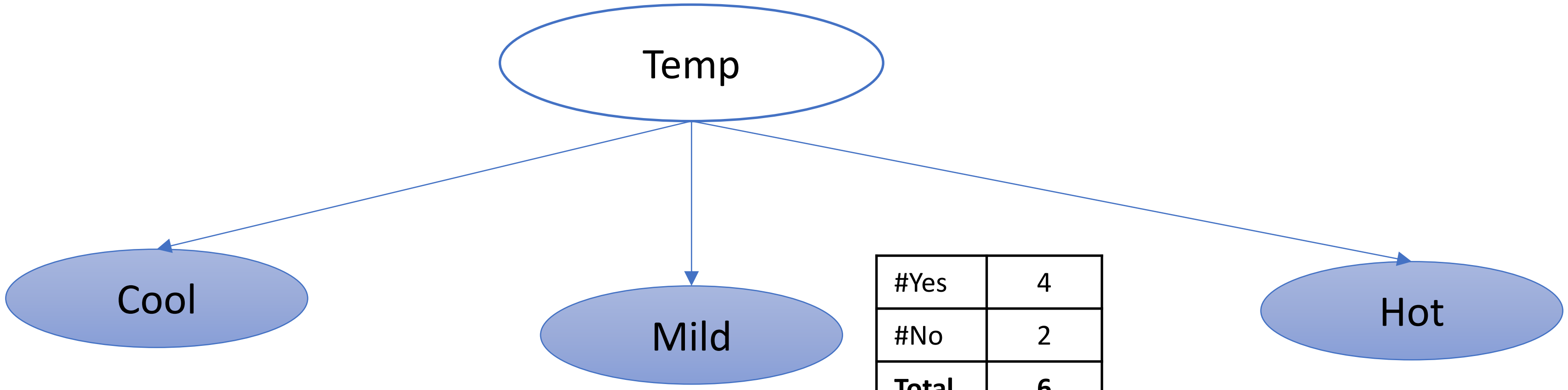
	Weight avg @ left node	Entropy @ left node	Weight avg @ right node	Entropy @ right node
Entropy after split				

Entropy before spilt	Entropy after split	Information Gain
0.94		0.04

Temp

#Yes	9
#No	5
Total	**14**

Entropy before split : 0.94



Cool:

#Yes	3
#No	1
Total	**4**

Mild:

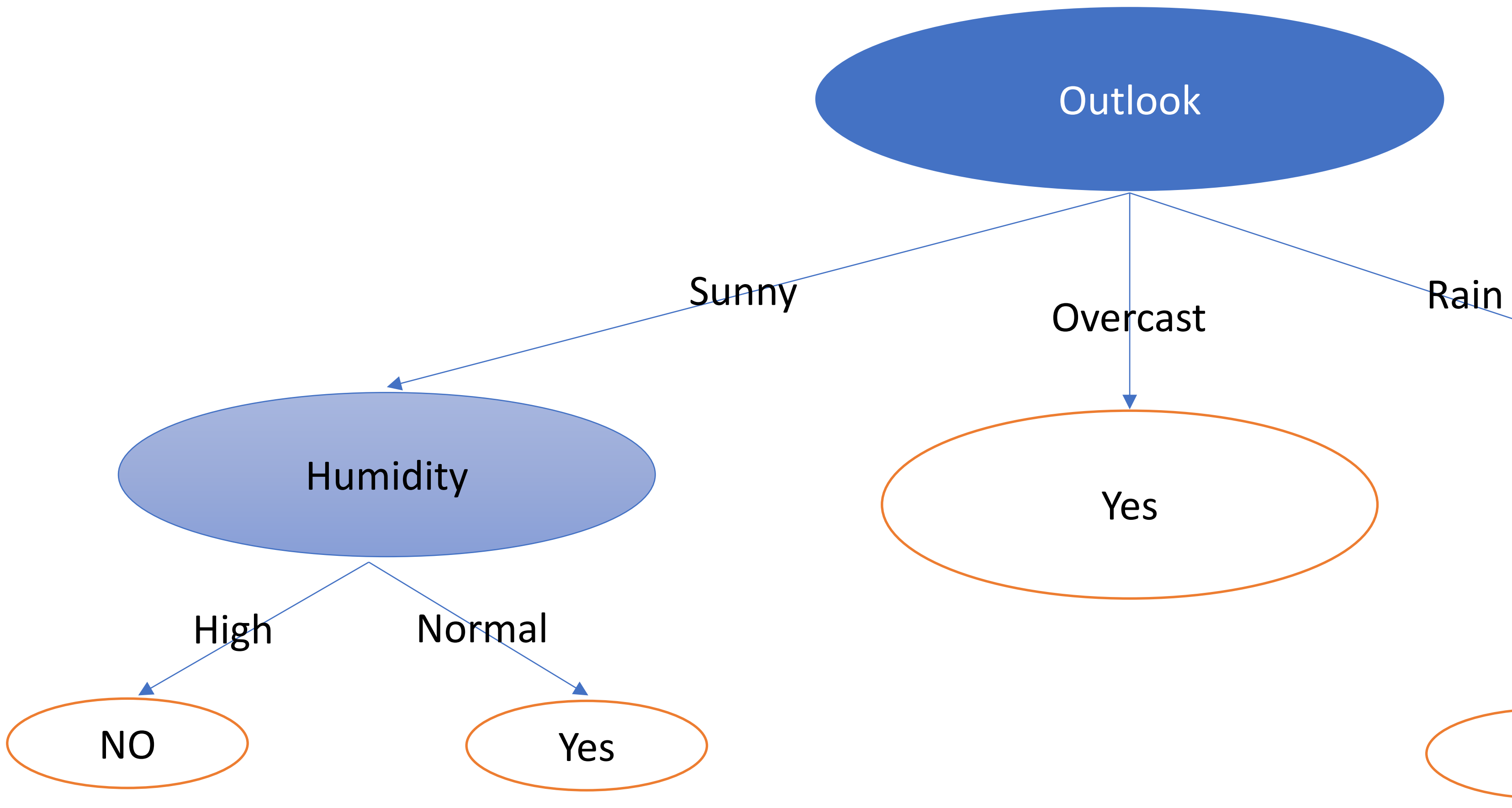
#Yes	4
#No	2
Total	**6**

Hot:

#Yes	2
#No	2
Total	**4**

	Weight avg @ left node	Entropy @ left node	Weight avg @ middle node	Entropy @ middle node	Weight avg @ right node	Entropy @ right node
Entropy after split	0.28	0.81	0.42	0.91	0.28	1

Entropy before spilt	Entropy after split	Information Gain
0.94	0.889	0.051



Outlook	Temp	Humidity	Wind	Play Tennis
Rain	Cool	Normal	Weak	Yes
Rain	Mild	Normal	Weak	Yes
Rain	Cool	Normal	Strong	No
Rain	Mild	High	Weak	Yes
Rain	Mild	High	Strong	No

Outlook	Temp	Humidity	Wind	Play Tennis
Sunny	Cool	Normal	Strong	Yes
Rain	Mild	High	Strong	No

Pure node : All elements belongs to one class

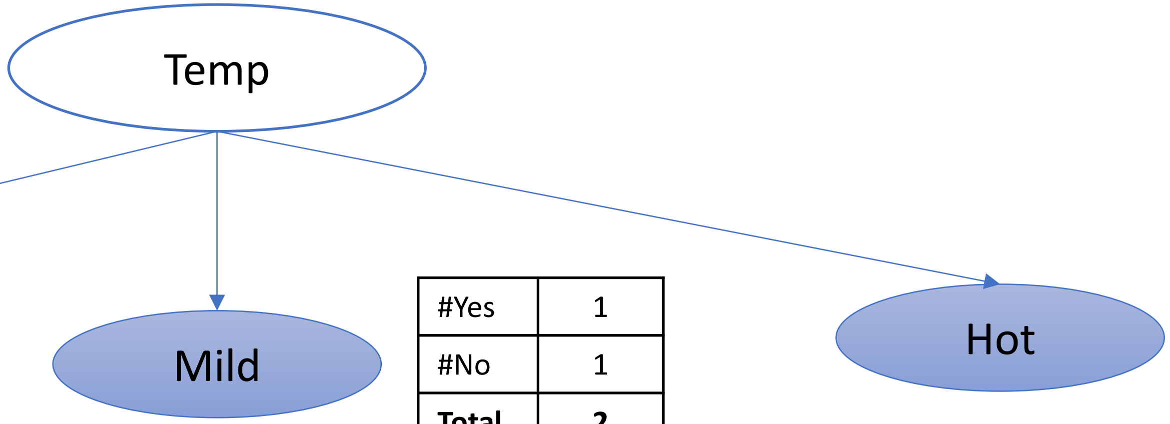
Entropy = 0

Temp : 2^{nd} Level

Temp	Play Tennis
Cool	Yes
Mild	Yes
Hot	No
Hot	No
Mild	No

#Yes	2
#No	3
Total	**5**

Entropy before split : 0.97



Cool:

#Yes	1
#No	0
Total	**1**

Mild:

#Yes	1
#No	1
Total	**2**

Hot:

#Yes	0
#No	2
Total	**2**

	Weight avg @ left node	Entropy @ left node	Weight avg @ middle node	Entropy @ middle node	Weight avg @ right node	Entropy @ right node
Entropy after split	0.2	0	0.4	1	0.4	0

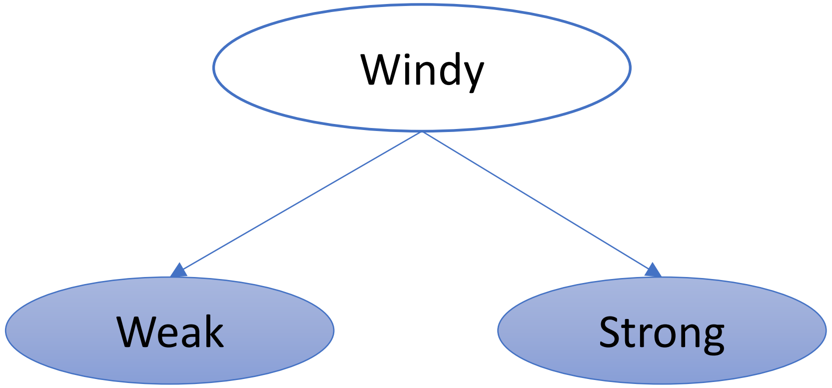
Entropy before spilt	Entropy after split	Information Gain
0.97	0.4	0.57

Windy : 2^{nd} Level

Wind	Play Tennis
Weak	Yes
Strong	Yes
Weak	No
Strong	No
Weak	No

#Yes	2
#No	3
Total	**5**

Entropy before split : 0.97



#Yes	1
#No	2
Total	**3**

#Yes	1
#No	1
Total	**2**

	Weight avg @ left node	Entropy @ left node	Weight avg @ right node	Entropy @ right node
Entropy after split	0.6	0.918	0.4	1

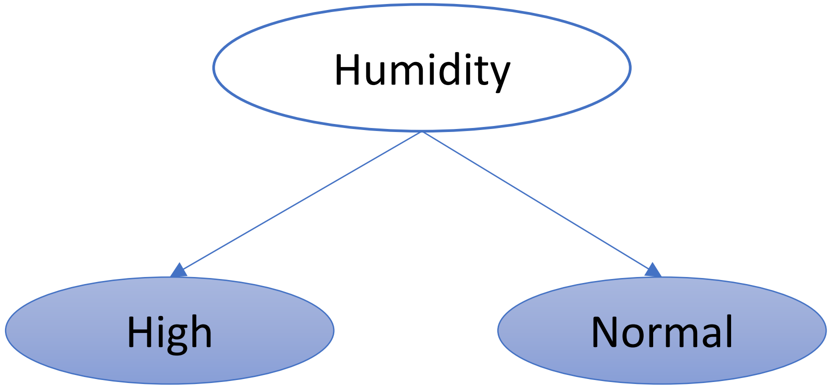
Entropy before spilt	Entropy after split	Information Gain
0.97	0.9508	0.019

Humidity: 2^{nd} Level

Outlook	Temp	Humidity	Wind	Play Tennis
Sunny	Cool	Normal	Weak	Yes
Sunny	Mild	Normal	Strong	Yes
Sunny	Hot	High	Weak	No
Sunny	Hot	High	Strong	No
Sunny	Mild	High	Weak	No

#Yes	2
#No	3
Total	**5**

Entropy before split : 0.97



#Yes	0
#No	3
Total	**3**

#Yes	2
#No	0
Total	**2**

Entropy @ after split =

Entropy @ left node :

+

Entropy @ right node :

Entropy before spilt	Entropy after split	Information Gain
0.97	0	0.97

DT Algorithm

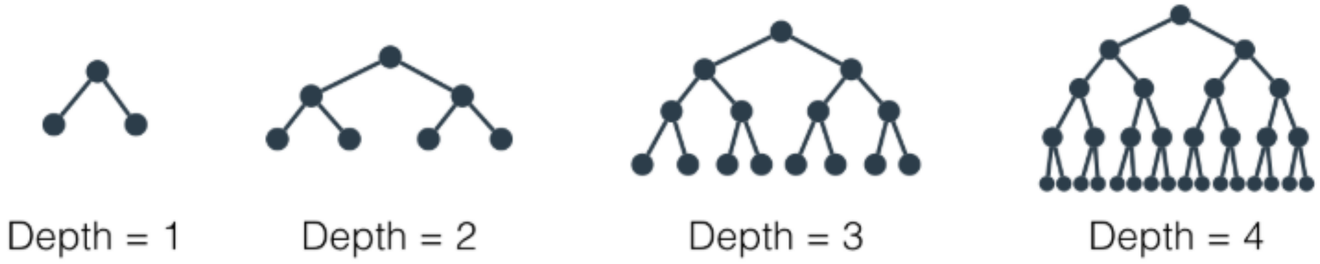
1. Start at the root node : Select Feature which gives Max IG
2. Split the data at root node into different nodes using the feature x_i which has maximize information gain.
3. Assign the subset data to all new child nodes after spliting.
4. Repeat steps 1 and 2 for each new child node
   - **Till : Stop if leaf nodes are pure or early stopping criteria is satisfied**

Stopping criteria

❖ **Leaf nodes are Pure** : All elements belong to one class

❖ **Depth of Tree** : A maximal node depth is reached

- Depth of Tree is small → Model Underfit
- Depth of Tree is large → Model Overfits
- if last node is not a pure node then majority is taken as decision



Maximum depth of a decision tree

❖ **min_samples_split :**

The minimum number of samples required to split an internal node

❖ **min_samples_leaf :**

The minimum number of samples required to be at a leaf node.

Gini Index

Gini Impurity tells us what is the probability of misclassifying an observation.
Gini impurity can be considered as an alternative for the entropy method.

Note that **the lower the Gini the better the split**. In other words the lower the likelihood of misclassification.

$$Gini = 1 - \sum_{i=1}^{C} (p_i)^2$$

C : Number of Classes in the label
Pi: Count of Class j / Total count

Win Win Win Win Win Loss Loss Loss Loss Loss

WIN: 5
Prob 1 = 5/10
= 0.5

Lost: 5
Prob 2 = 5/10
= 0.5

Gini Index = 1 – (prob(win)^2 +prob(loss)^2)
= 1- (0.25 + 0.25)
= 1- 0.5 =0.5

Which one to use : Entropy vs Gini Index

Entropy

$$I_H = -\sum_{j=1}^{c} p_j log_2(p_j)$$

C : Number of Classes in the label
Pj: Count of Class j / Total count

log computation
Takes more time ,more complex

Gini Index

$$Gini = 1 - \sum_{i=1}^{C} (p_i)^2$$

C : Number of Classes in the label
Pi: Count of Class j / Total count

square computation

- Since the computing square is cheaper than the logarithmic function we prefer Gini impurity over entropy.

Variants of Decision Tree

- CART Classification & Regression Tree) :
  - Uses Gini Index(Classification) as metrics
  - Uses MAE (Regression)

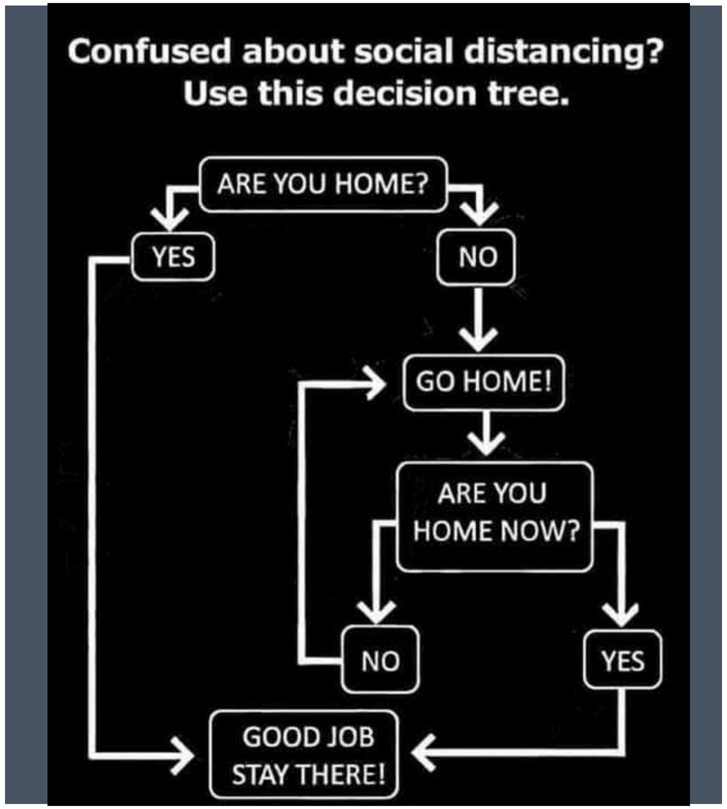
- ID3 (Iterative Dichotomiser 3) :
  - Uses Entropy as metrics

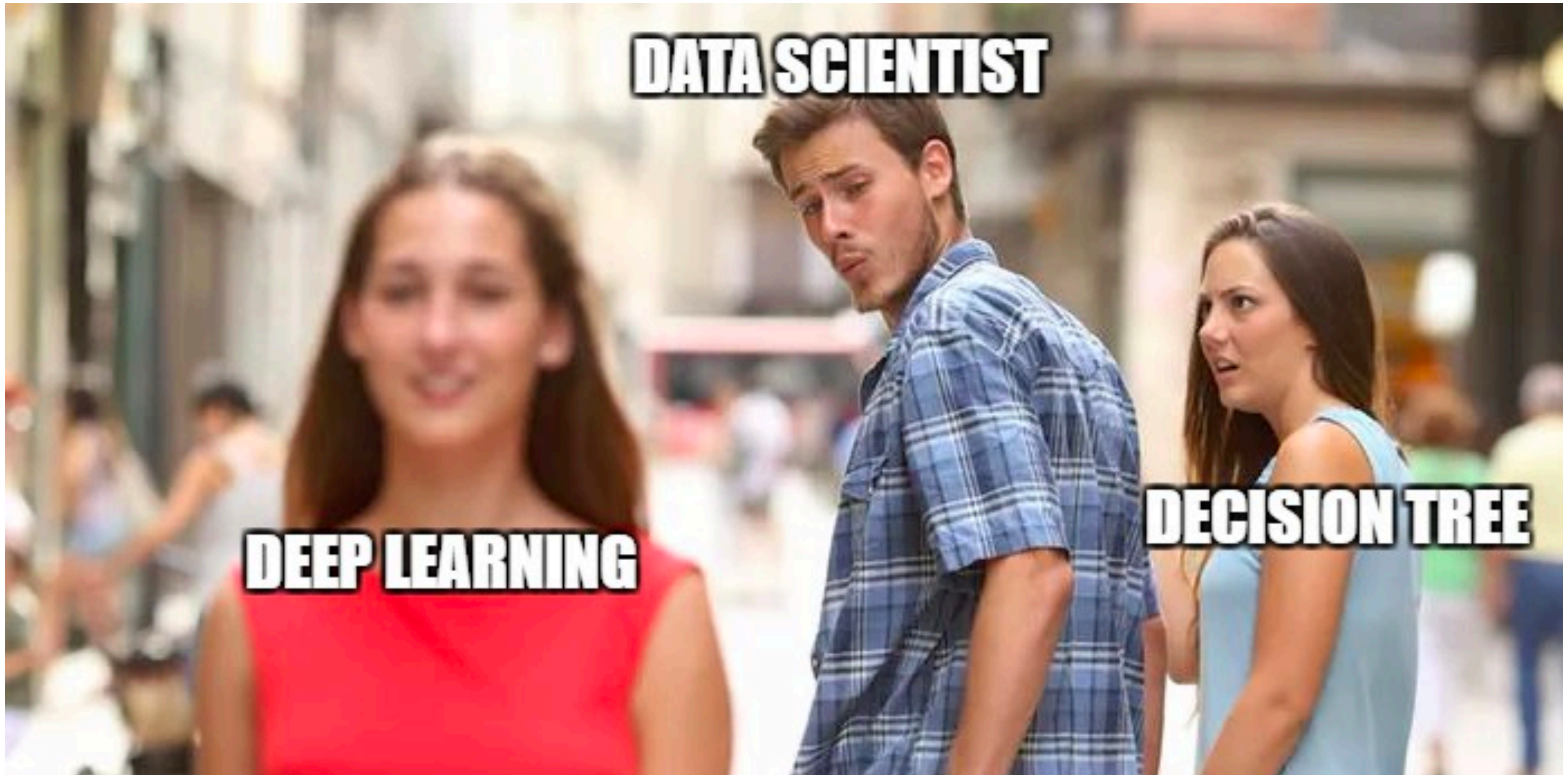
How to deal with Numeric Feature ?

Regression problem

DT Demo

Summary of DT





Thank you

- Calculate entropy of below variable ‘Pass’ :

  [Yes , Yes , Yes , Yes , Yes, No, No, Yes, No, Yes, No, Yes , Yes, No]