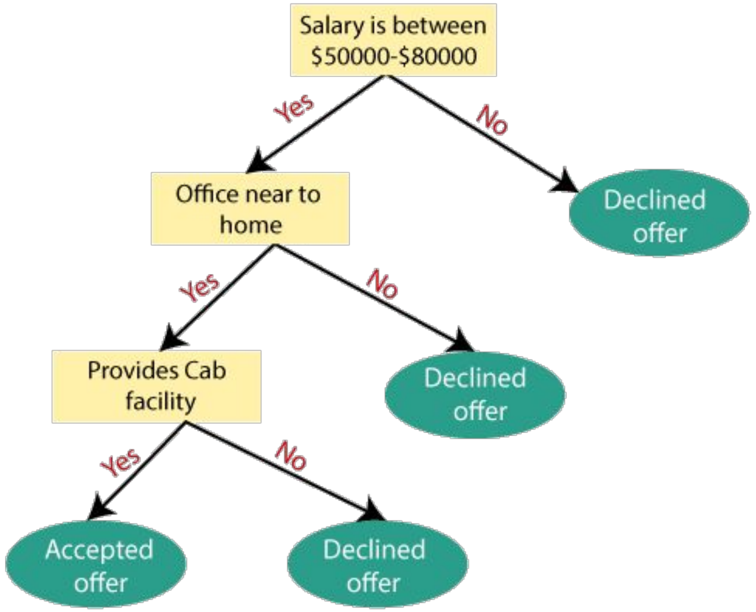
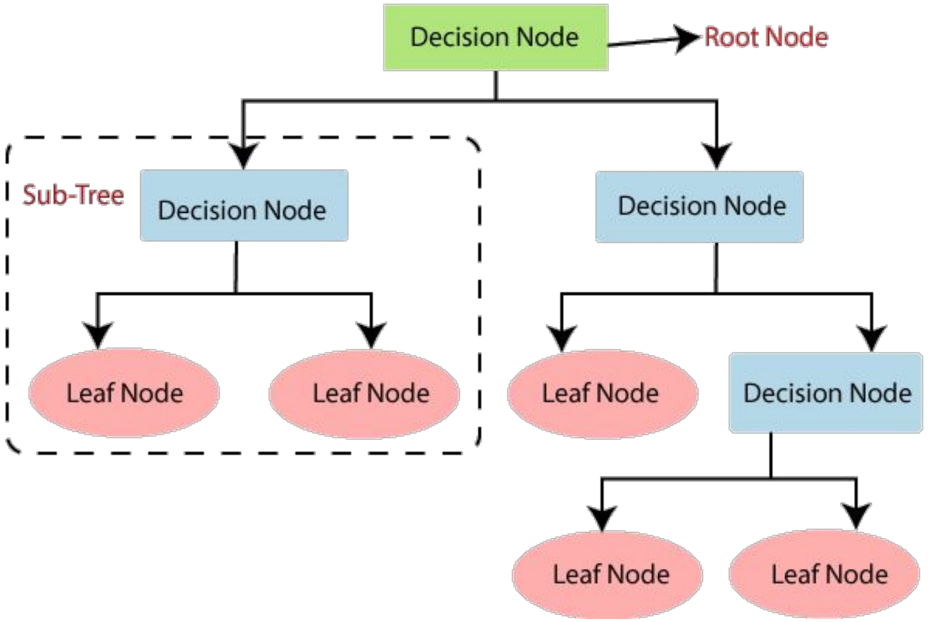
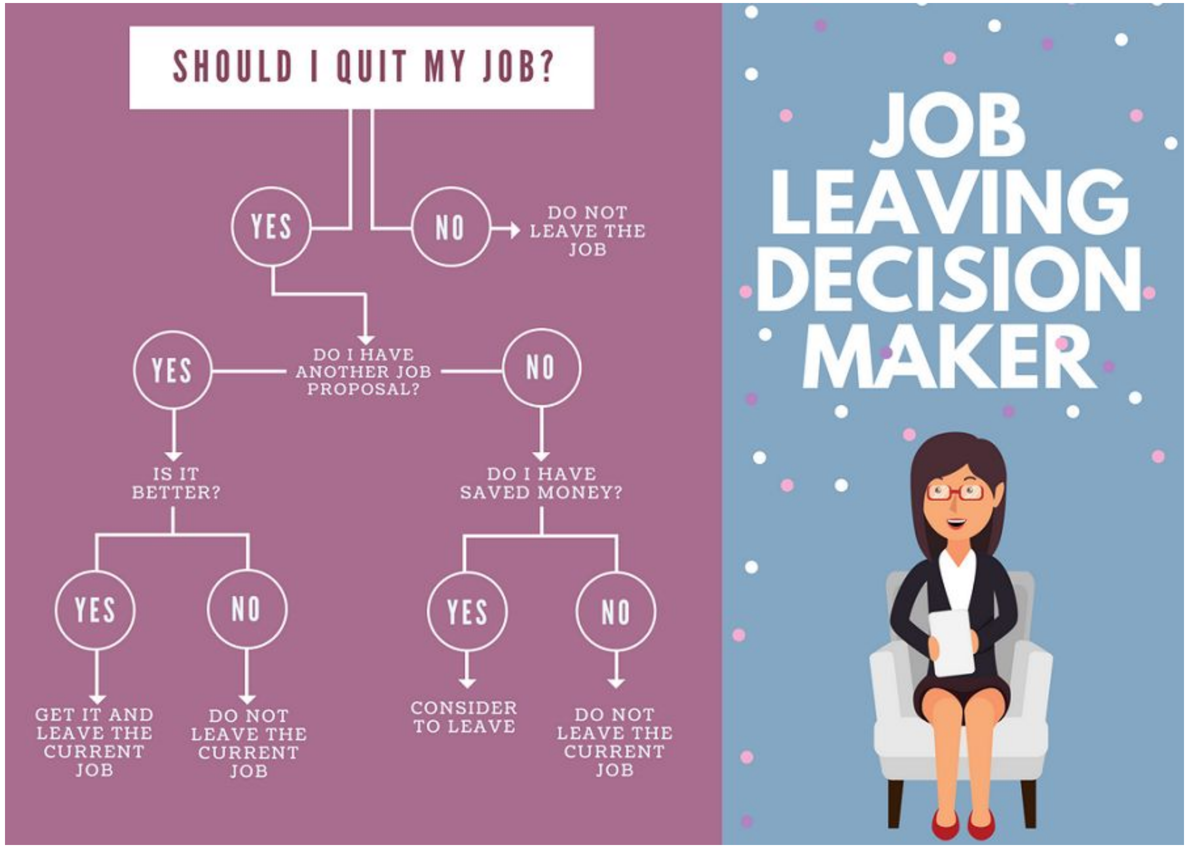
Decision Tree

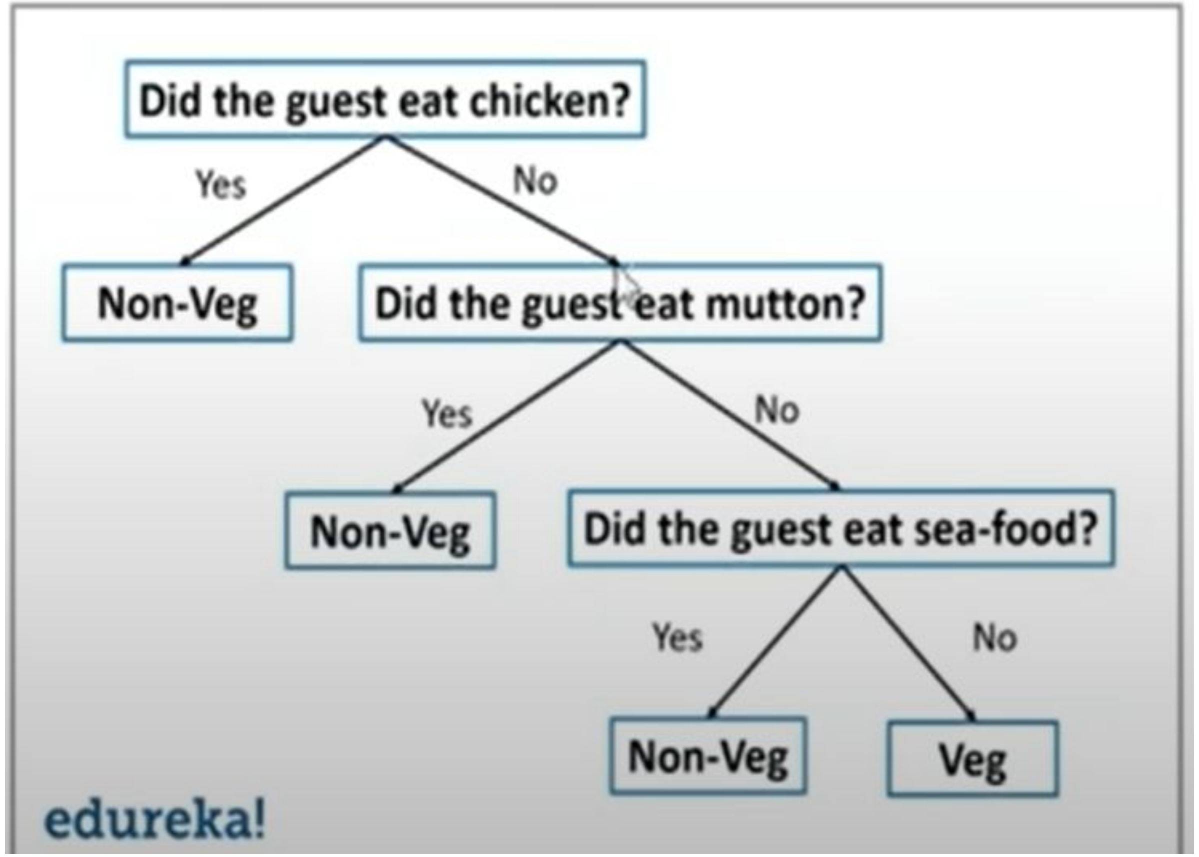




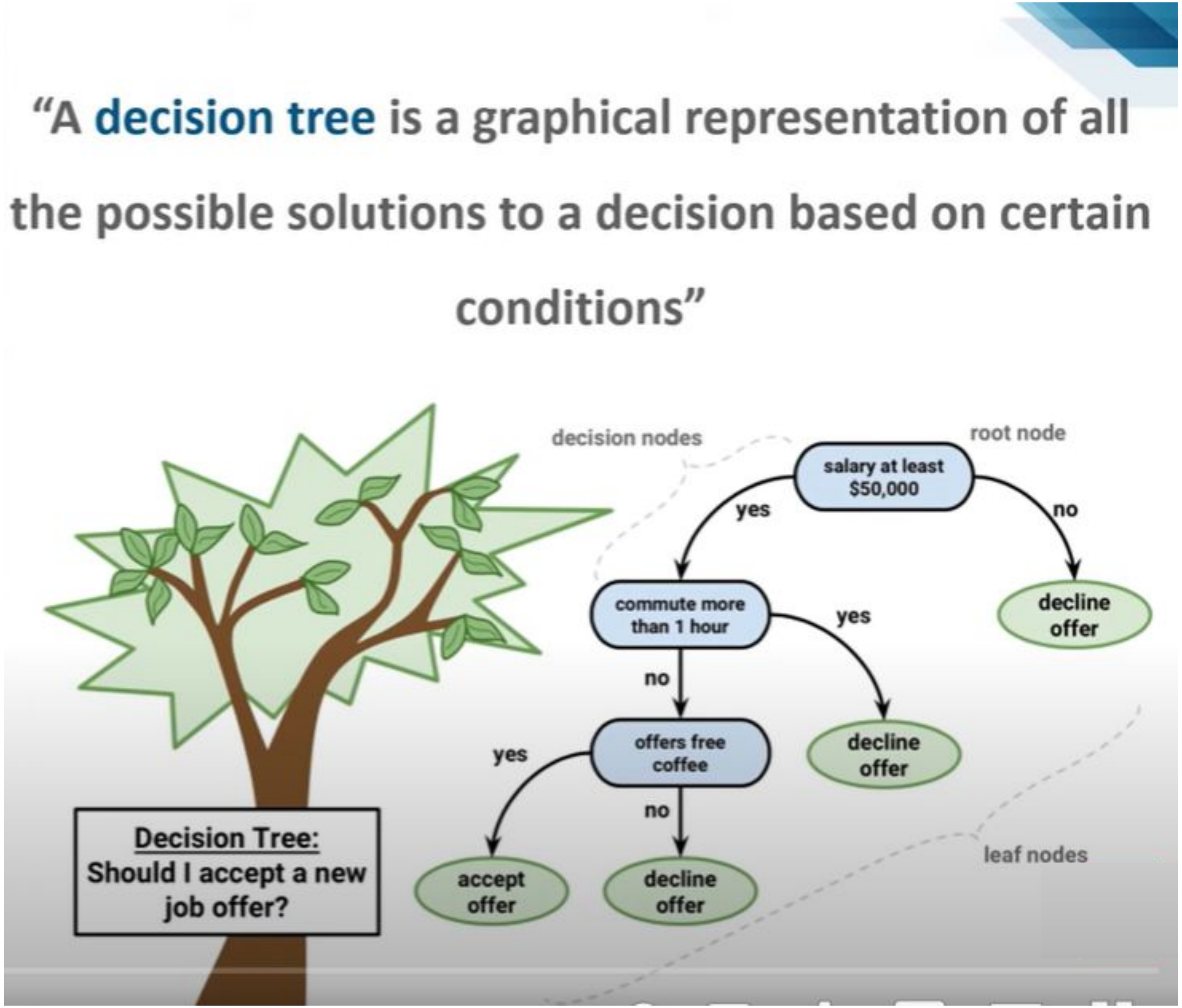
Use cases



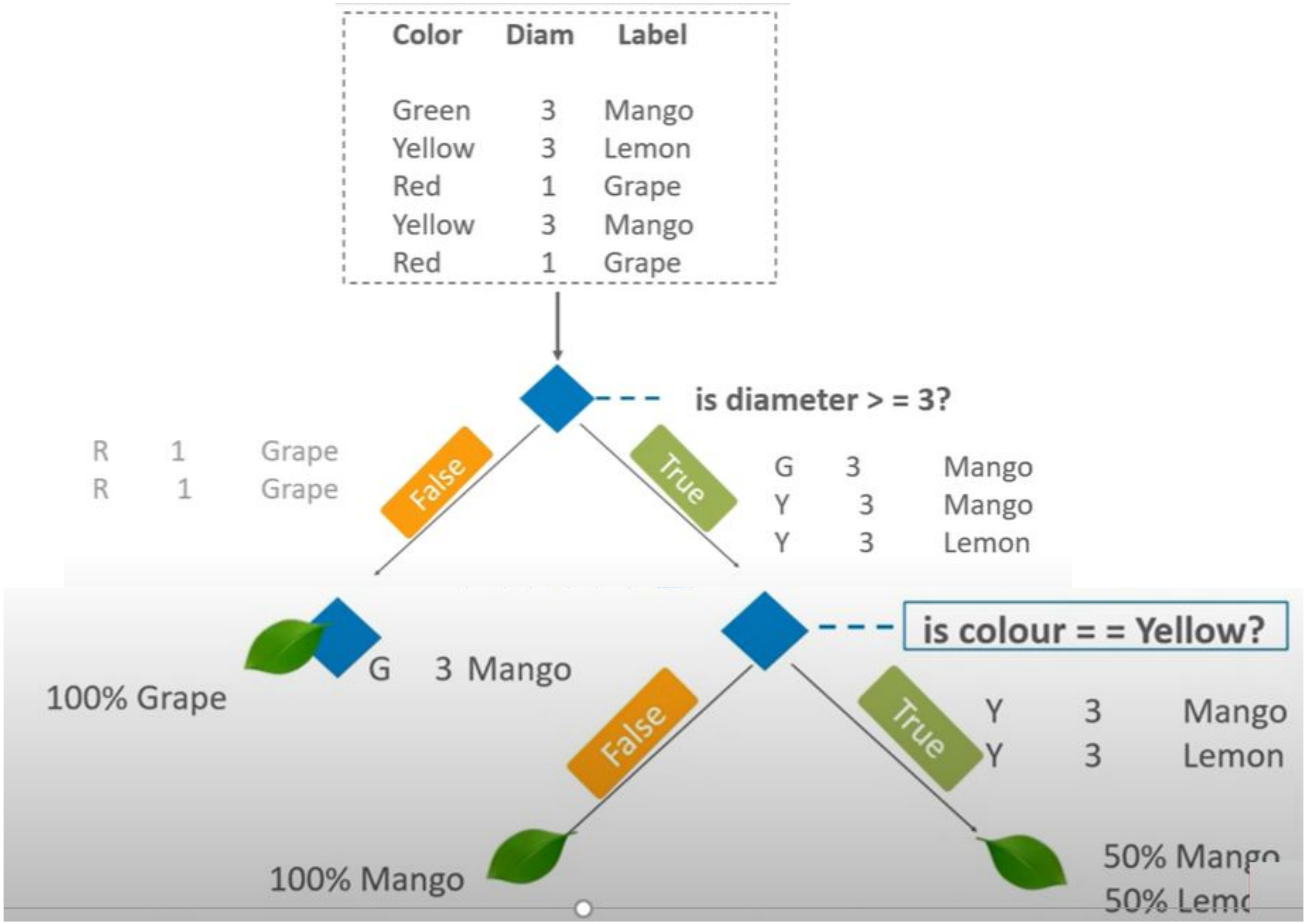
SHOULD I QUIT MY JOB?
YES
NO
DO NOT LEAVE THE JOB
DO I HAVE ANOTHER JOB PROPOSAL?
YES
NO
IS IT BETTER?
DO I HAVE SAVED MONEY?
YES
NO
YES
NO
GET IT AND LEAVE THE CURRENT JOB
DO NOT LEAVE THE CURRENT JOB
CONSIDER TO LEAVE
DO NOT LEAVE THE CURRENT JOB
JOB LEAVING DECISION MAKER



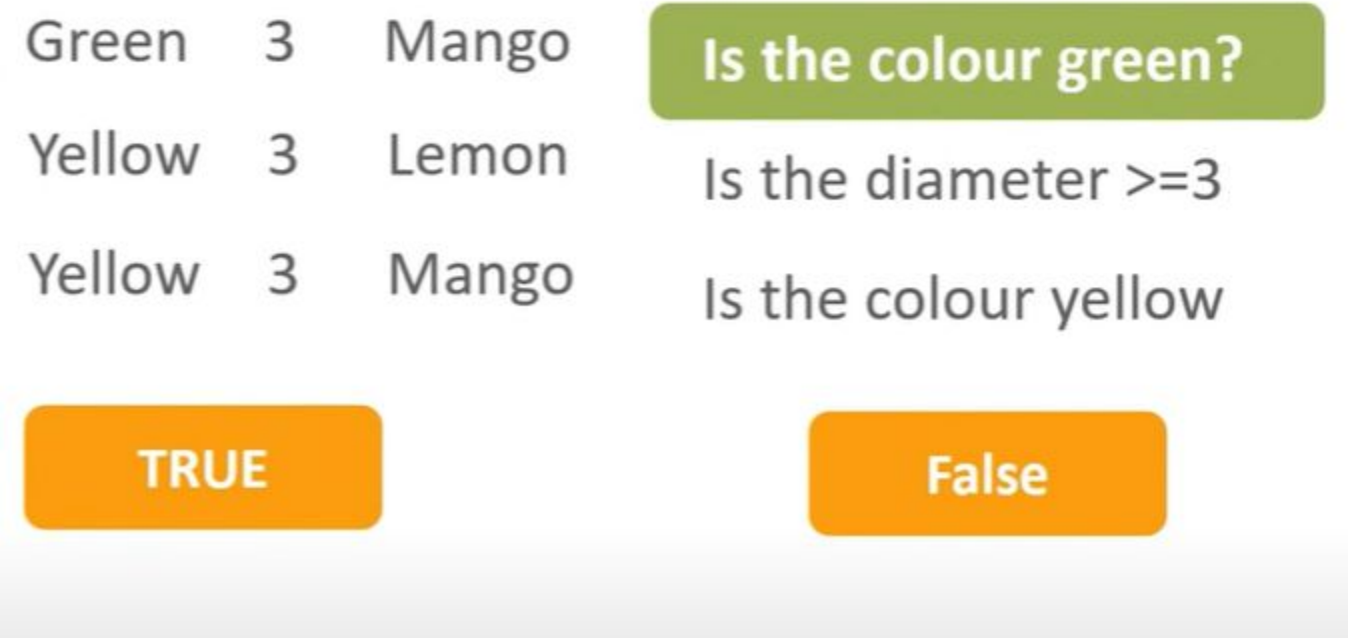
Did the guest eat chicken?
Yes
No
Non-Veg
Did the guest eat mutton?
Yes
No
Non-Veg
Did the guest eat sea-food?
Yes
No
Non-Veg
Veg
edureka!

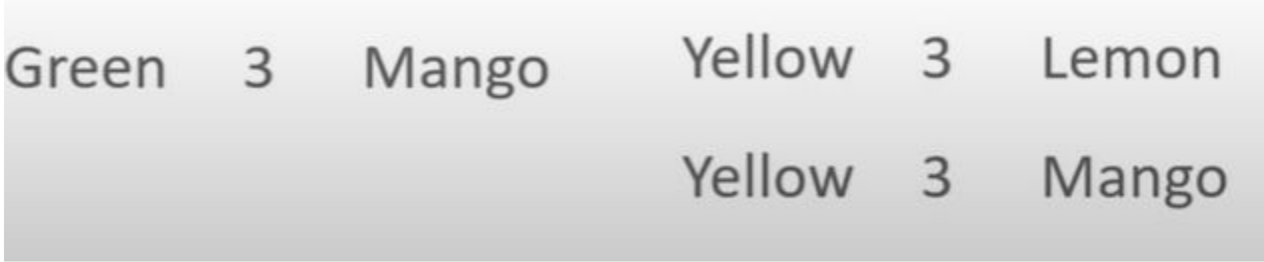


- Starts with root and branches to number of decisions based on the conditions.



Color Diam Label
Green 3 Mango
Yellow 3 Lemon
Red 1 Grape
Yellow 3 Mango
Red 1 Grape
is diameter > = 3?
False
True
R 1 Grape
R 1 Grape
G 3 Mango
Y 3 Mango
Y 3 Lemon
100% Grape
G 3 Mango
is colour = = Yellow?
False
True
Y 3 Mango
Y 3 Lemon
100% Mango
50% Mango
50% Lemon

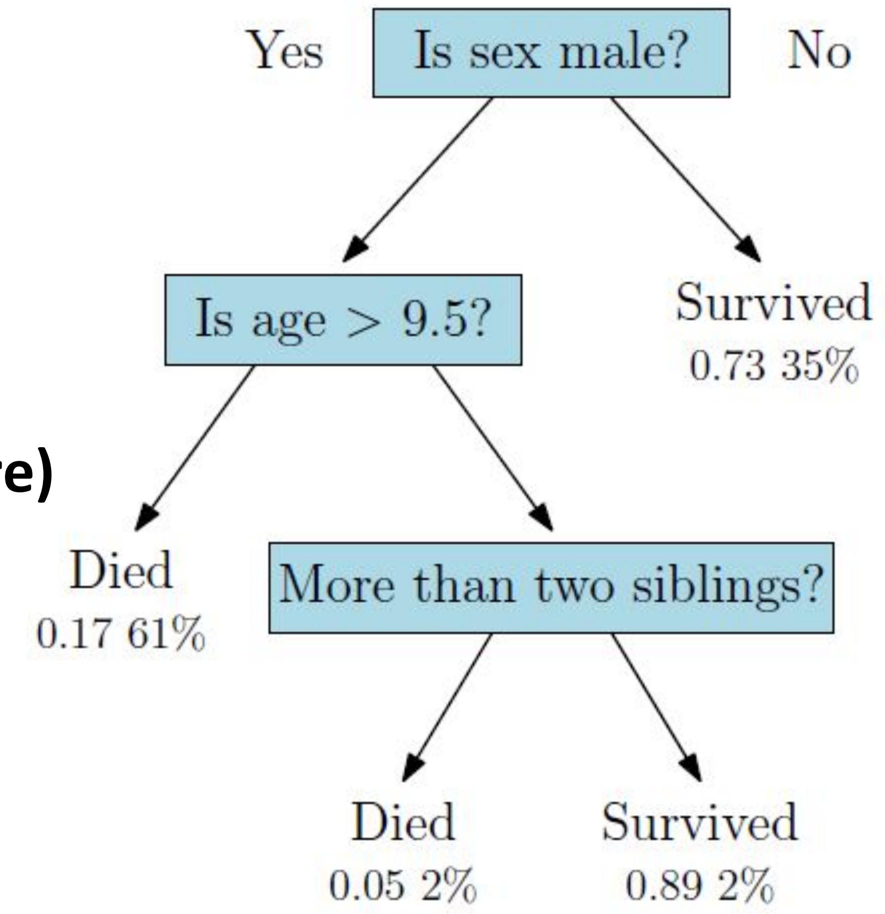




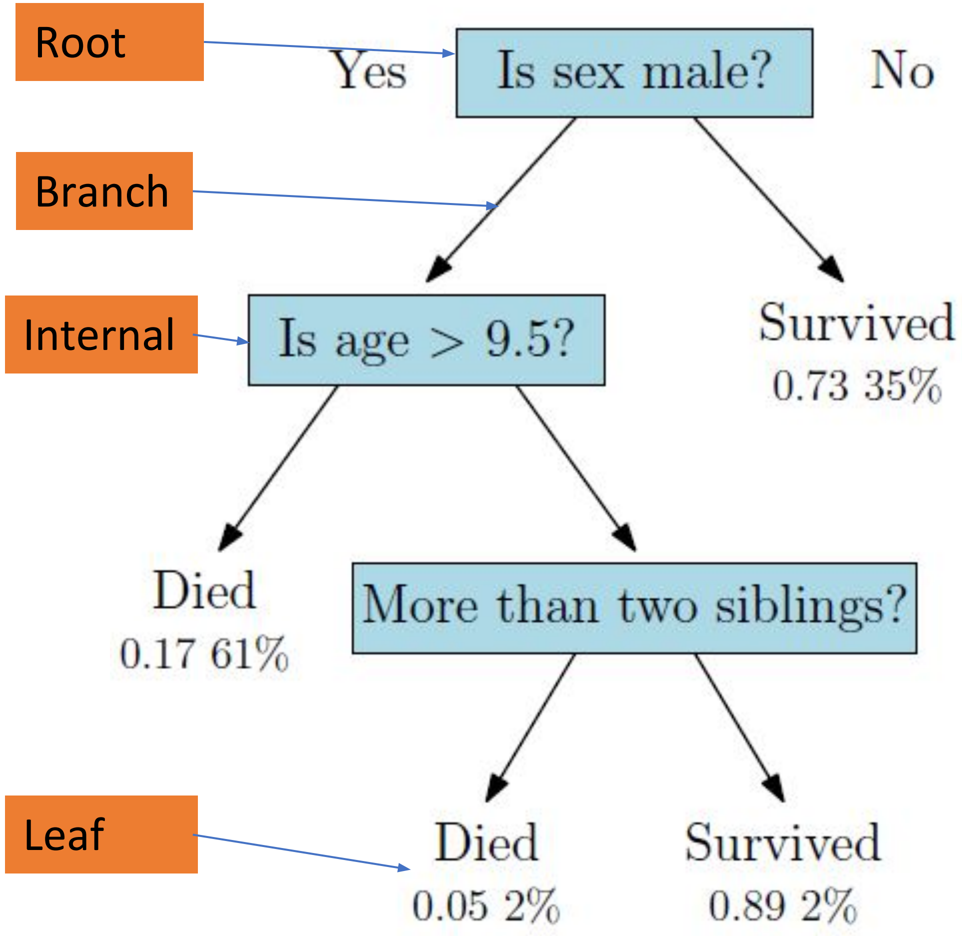
- **Note:** Questions resembles the data set

Decision Tree

- Decision tree is a supervised learning algorithm.
- It uses a tree-like model of decisions.
- It is used for both **classification and regression**.
- Decision Tree algorithms are referred to as CART or **Classification and Regression Trees**.
- Each node represents the **predictor variable (feature)**
- Link between the nodes represents **a decision**
- Each leaf node represents **an outcome (response variable)**



Structure of Decision Tree



Root Node: Starting point of the tree. This is a decision node at the topmost level where the first split performed. It represents the most significant predictor variable.

Internal Node: Represents a decision point (predictor variable) that leads to the prediction of outcome.

Leaf Nodes: Lowest nodes which represents final class of the decision outcome.

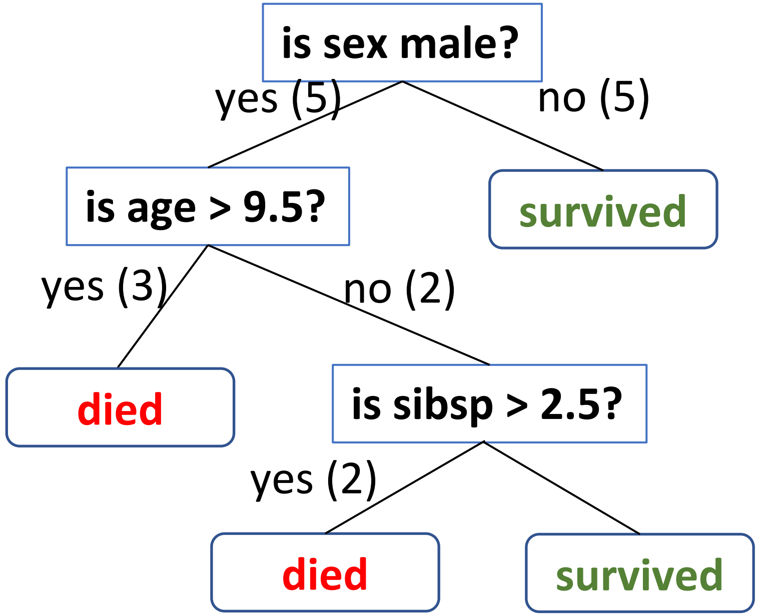
Branches: Represents connection between nodes. Each branch represents a response as yes or no.

Splitting: *D*ividing the root/internal node into different parts on the basis of some conditions.

Problem statement:

Uses titanic data set for predicting whether a passenger will survive or not.

	survived	pclass	sex	age	sibsp	parch	fare	embarked	class	who	adult_male	deck	embark_town
0	0	3	male	22.0	1	0	7.2500	S	Third	man	True	NaN	Southampton
1	1	1	female	38.0	1	0	71.2833	C	First	woman	False	C	Cherbourg
2	1	3	female	26.0	0	0	7.9250	S	Third	woman	False	NaN	Southampton
3	1	1	female	35.0	1	0	53.1000	S	First	woman	False	C	Southampton
4	0	3	male	35.0	0	0	8.0500	S	Third	man	True	NaN	Southampton
5	0	3	male	NaN	0	0	8.4583	Q	Third	man	True	NaN	Queenstown
6	0	1	male	54.0	0	0	51.8625	S	First	man	True	E	Southampton
7	0	3	male	2.0	3	1	21.0750	S	Third	child	False	NaN	Southampton
8	1	3	female	27.0	0	2	11.1333	S	Third	woman	False	NaN	Southampton
9	1	2	female	14.0	1	0	30.0708	C	Second	child	False	NaN	Cherbourg



- Construct decision tree model uses 3 features/attributes/columns from the data set, namely sex, age and sibsp (number of spouses or children along).

Which Question to ask and When?

Day	Weather	Temperature	Humidity	Wind	Play?
1	Sunny	Hot	High	Weak	No
2	Cloudy	Hot	High	Weak	Yes
3	Sunny	Mild	Normal	Strong	Yes
4	Cloudy	Mild	High	Strong	Yes
5	Rainy	Mild	High	Strong	No
6	Rainy	Cool	Normal	Strong	No
7	Rainy	Mild	High	Weak	Yes
8	Sunny	Hot	High	Strong	No
9	Cloudy	Hot	Normal	Weak	Yes
10	Rainy	Mild	High	Strong	No

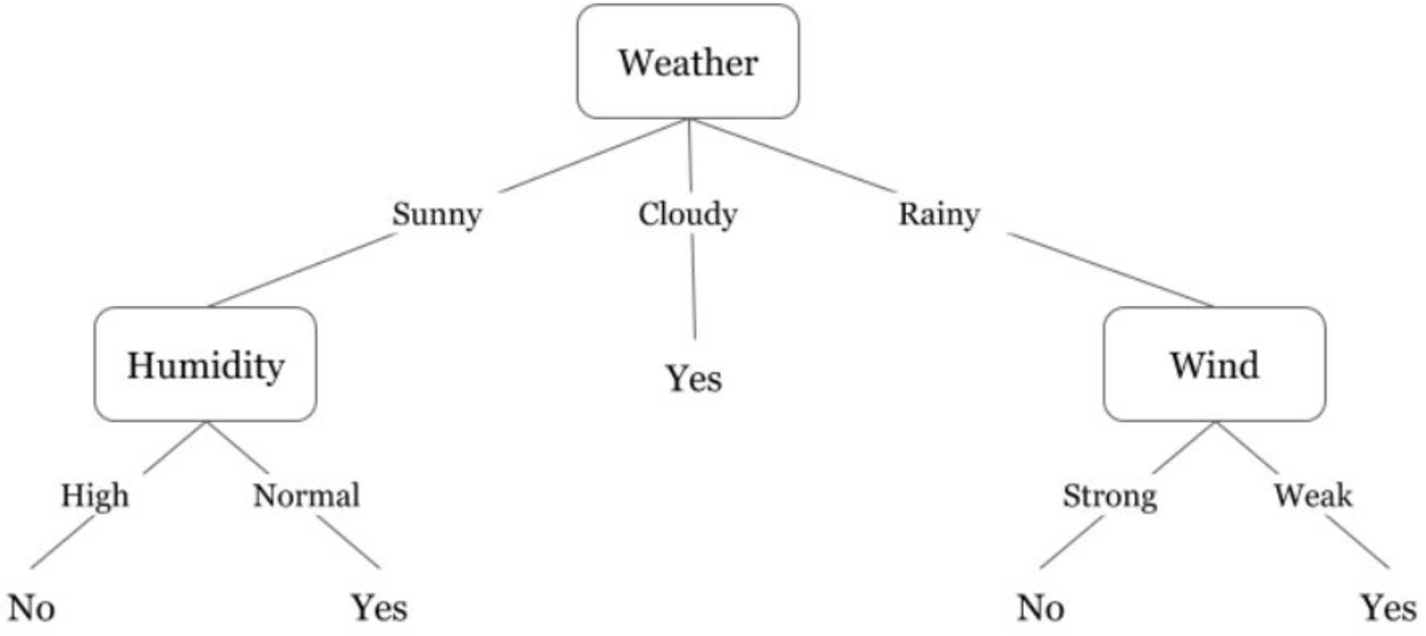
Which one among them should you pick first?

But How do we choose the best attribute?

Or

How does a tree decide where to split?

Day	Weather	Temperature	Humidity	Wind	Play?
1	Sunny	Hot	High	Weak	No
2	Cloudy	Hot	High	Weak	Yes
3	Sunny	Mild	Normal	Strong	Yes
4	Cloudy	Mild	High	Strong	Yes
5	Rainy	Mild	High	Strong	No
6	Rainy	Cool	Normal	Strong	No
7	Rainy	Mild	High	Weak	Yes
8	Sunny	Hot	High	Strong	No
9	Cloudy	Hot	Normal	Weak	Yes
10	Rainy	Mild	High	Strong	No



Step by Step procedure for building decision tree

- Step 1: Select Best Attribute (A)
  - best predictor variable that separates the data into different classes most effectively or feature that best splits the data
- Step 2: Assign A as a decision variable for the root node
- Step 3: For each value of A, build a descend of the node
- Step 4: Assign classification labels to the leaf node
- Step 5: If data is correctly classified: Stop
- Step 6: Else iterate over the tree
  - Keep changing the position of predictor variables in the tree or change the root node also, to get the correct output

Reference: https://youtu.be/JMUxmLyrhSk

How Does A Tree Decide Where To Split?

Gini Index

The measure of impurity (or purity) used in building decision tree in CART is Gini Index

Information Gain

The information gain is the decrease in entropy after a dataset is split on the basis of an attribute. Constructing a decision tree is all about finding attribute that returns the highest information gain

Chi Square

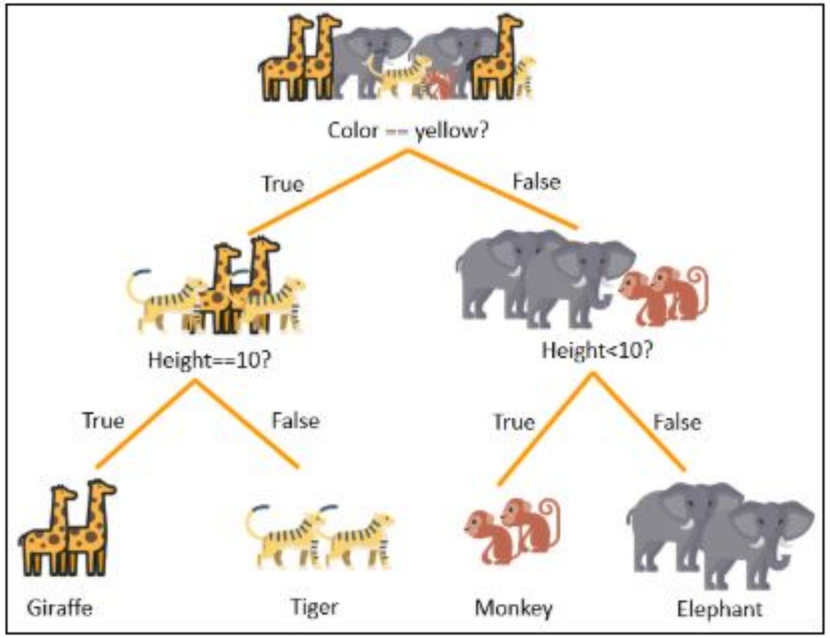
It is an algorithm to find out the statistical significance between the differences between sub-nodes and parent node

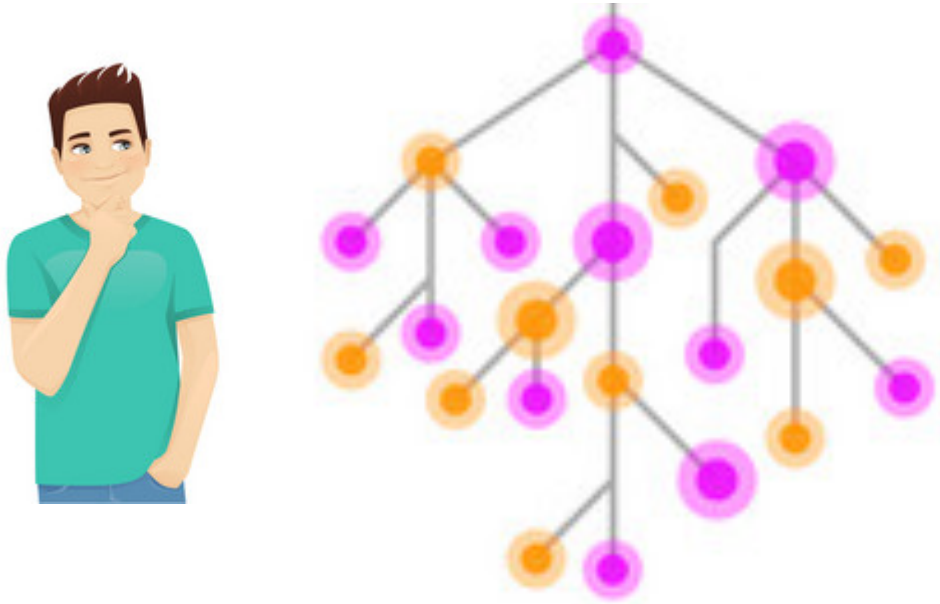
Reduction in Variance

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

ID3 Algorithm

- There are many ways to construct the decision tree.
- One of the effective way to construct decision tree is ID3 (Iterative Dichotomizer 3) algorithm, uses the concept of **entropy and information gain**.



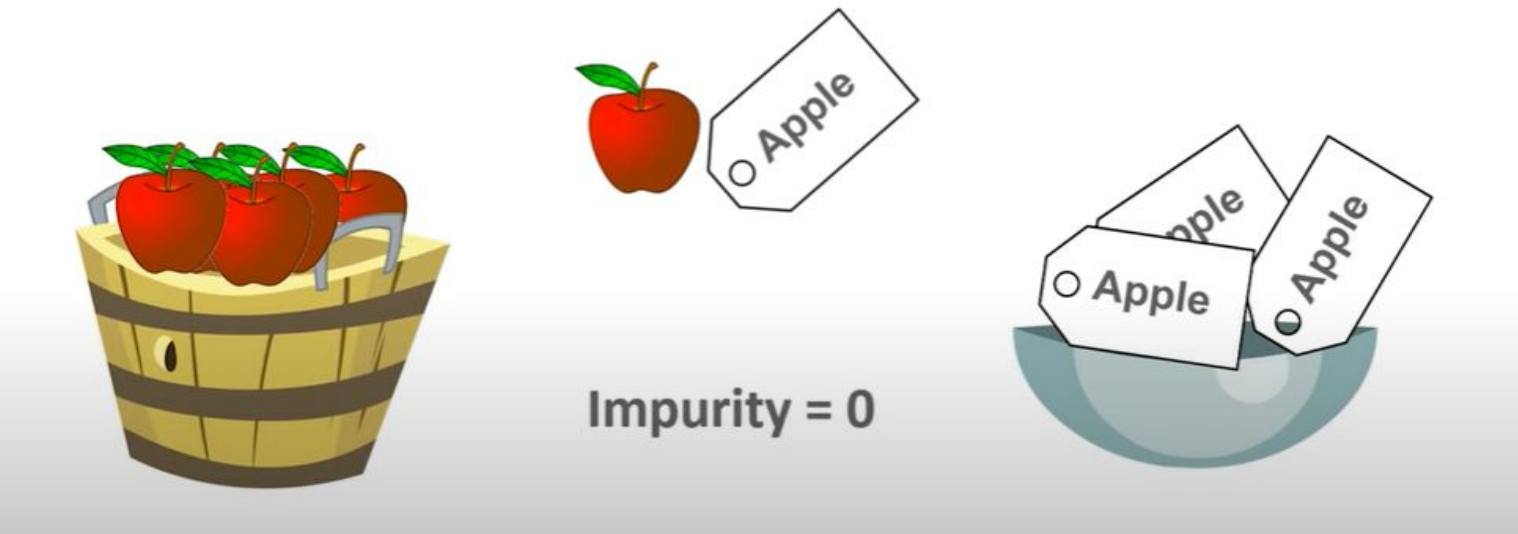


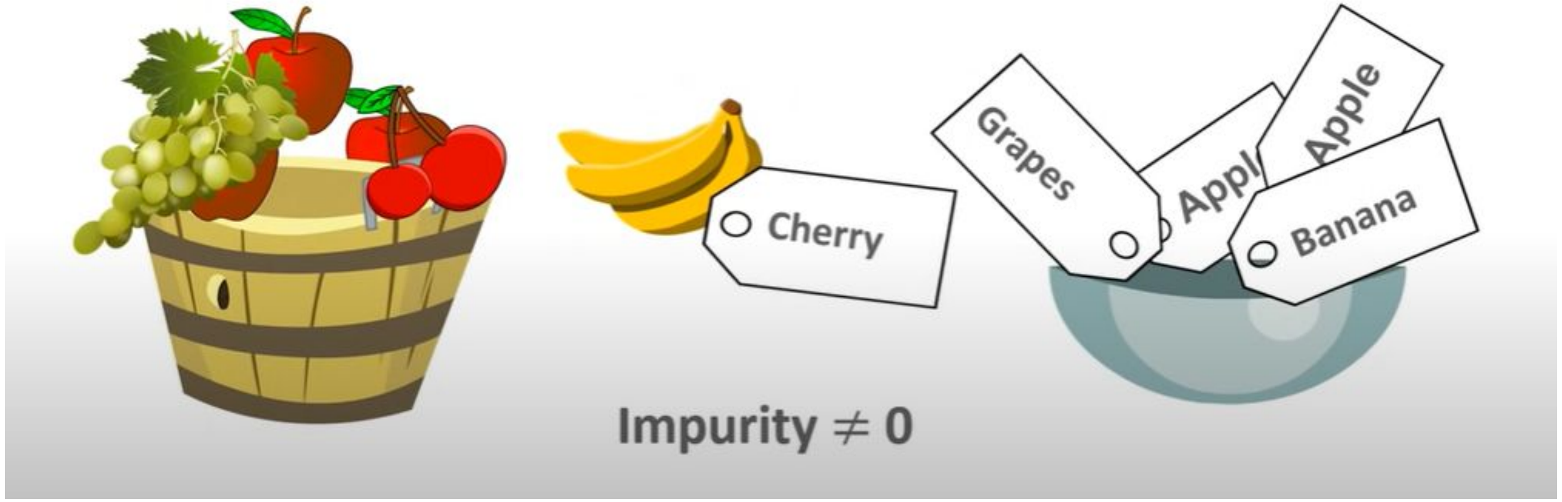
Decide the best attribute (predictor variable)

- Best attribute separates data into different classes most effectively.
  - A predictor variable that best splits the data set
- How will you decide the best predictor variable?

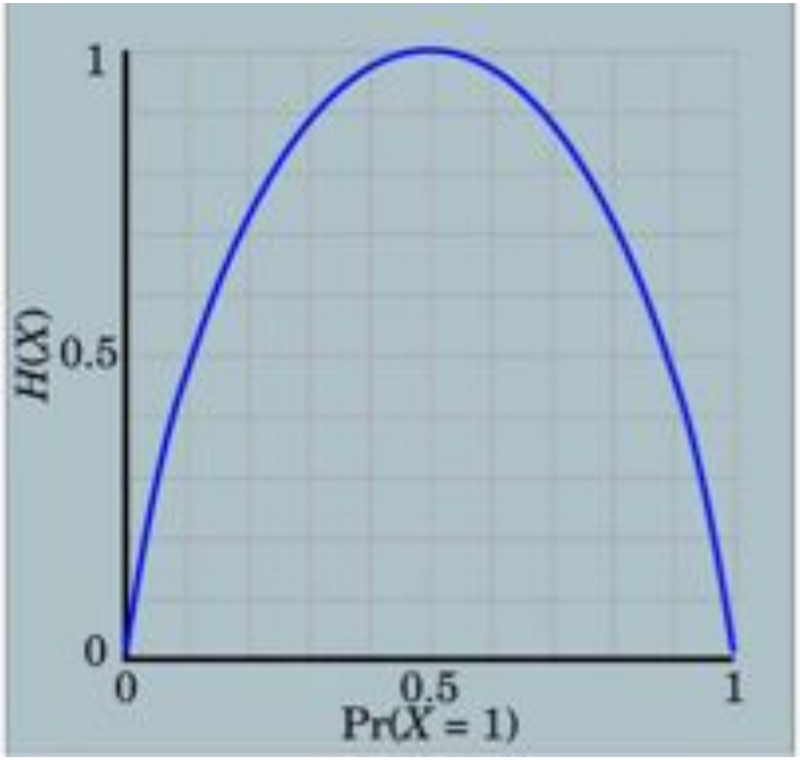
  Two measures: information gain and entropy
  - **Entropy:** measures the impurity or uncertainty of the data (How the decision tree split the data?)
  - **Information gain:** specifies how much information a particular predictor variable gives about the final outcome (used to choose the predictor variable that best splits the data)
  - The predictor variable with high information gain is considered as best attribute (root node) that divides the data into desired output classes

Reference: https://youtu.be/JMUxmLyrhSk


Apple
Apple
Apple
Impurity = 0


Cherry
Grapes
Apple
Banana
Impurity ≠ 0

- Entropy is the measure of impurity



If number of *yes* = number of *no ie P(S)* = 0.5

$\Rightarrow$ *Entropy(s) = 1*

If it contains all yes or all no ie P(S) = 1 or 0

$\Rightarrow$ *Entropy(s) = 0*

E(S) = -P(Yes) $\log_2 P(Yes)$

When P(Yes) =P(No) = 0.5 ie YES + NO = Total Sample(S)

E(S) = 0.5 $\log_2 0.5 -$ *0.5* $\log_2 0.5$

E(S) = 0.5($\log_2 0.5$ - $\log_2 0.5$*)*

E(S) = 1

$$E =- \sum_{i}^{C} p_i log_2 p_i$$

E(S) = -P(Yes) $\log_2 P(Yes)$

When P(Yes) = 1 ie YES = Total Sample(S)

E(S) = 1 $\log_2 1$

E(S) = 0

E(S) = -P(No) $\log_2 P(No)$

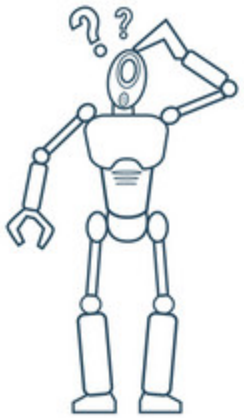
When P(No) = 1 ie No = Total Sample(S)

E(S) = 1 $\log_2 1$

E(S) = 0

Problem statement: For a given data set create a decision tree and classify the speed of the vehicle as Slow or Fast.

Road type	Obstruction	Speed limit	Speed
Steep	Yes	Yes	Slow
Steep	No	Yes	Slow
Flat	Yes	No	Fast
Steep	No	No	Fast



Predictor variables: road type, obstruction, speed limit
Target variable: speed

Reference: https://youtu.be/JMUxmLyrhSk

Step 1: Decide the **best variable** that splits the data set. i.e root node of decision tree

Calculate entropy (E) and information gain (IG)

$$E = -\sum_{i}^{C} p_i log_2 p_i$$

$IG =$ E(parent) – weighted average of E(children)

a. Calculate entropy of parent node

Target variable is Speed

It is the Parent node

Two classes: Slow and Fast

Speed
Slow
Slow
Fast
Fast

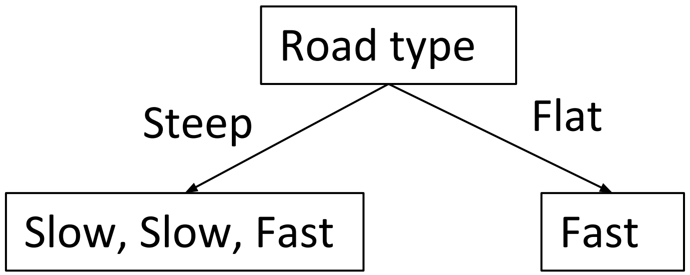
P(Slow) = 2/4 = 0.5

P(Fast) = 2/4 = 0.5

E(Parent) = - (P(Slow)*log2(P(Slow)) + P(Fast)*log2(P(Fast)))

= - (0.5*log2(0.5) + 0.5*log2(0.5)) = 1

b. Information gain of child node (Road type)



Road type	Obstruction	Speed limit	Speed
Steep	Yes	Yes	Slow
Steep	No	Yes	Slow
Flat	Yes	No	Fast
Steep	No	No	Fast

$$E = -\sum_{i}^{C} p_i log_2 p_i$$

$IG =$ E(parent) – weighted average of E(children)

b.1. Entropy of child

Entropy of right child node is

P(Fast) = 1/1 = 1

E(Road type=Flat) = - (P(Fast)*log2(P(Fast)))

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

Entropy of left child node is

P(Slow) = 2/3 = 0.667

P(Fast) = 1/3 = 0.334

E(Road type=Steep) = - (P(Slow)*log2(P(Slow)) + P(Fast)*log2(P(Fast)))

= - (0.667*log2(0.667)) + 0.334*log2(0.334)) = 0.9

b.2. Calculate the weighted average of E(child) for Road type

weighted average of E(child) = (no. of left_child) / (no. of parent) * E(Road type=Steep) + (no. of right_child) / (no. of parent) * E(Road type=Flat)

= (3/4) * 0.9 + (1/4) *0

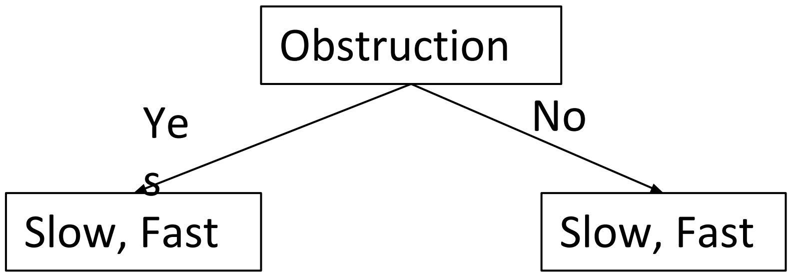
= 0.675

Parent node: 4

Right child: 1

Left child: 3

b. Information gain of child node (Obstruction)



Road type	Obstruction	Speed limit	Speed
Steep	Yes	Yes	Slow
Steep	No	Yes	Slow
Flat	Yes	No	Fast
Steep	No	No	Fast

$$E = -\sum_{i}^{C} p_i log_2 p_i$$

$IG =$ E(parent) – weighted average of E(children)

b.1. Entropy of child

Entropy of right child node, E(Obstruction=No) = 1

P(Slow) = 1/2 = 0.5

P(Fast) = 1/2 = 0.5

E(Obstruction=No) = - (P(Slow)*log2(P(Slow)) + P(Fast)*log2(P(Fast)))

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

= 1

Entropy of left child node, E(Obstruction=Yes) = 1

b.2. Calculate the weighted average of E(child) for Obstruction

Parent node: 4

Right child: 2

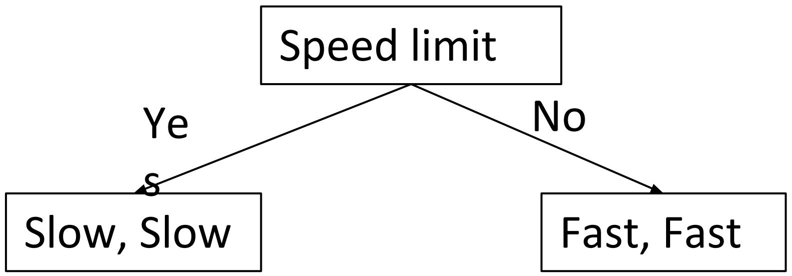
Left child: 2

weighted average of E(child) = (no. of left_child) / (no. of parent) * E(Obstruction=Yes) +
(no. of right_child) / (no. of parent) * E(Obstruction=No)

= (2/4) * 1 + (2/4) *1

= 1

b. Information gain of child node (Speed limit)



Road type	Obstruction	Speed limit	Speed
Steep	Yes	Yes	Slow
Steep	No	Yes	Slow
Flat	Yes	No	Fast
Steep	No	No	Fast

$$E = -\sum_{i}^{C} p_i log_2 p_i$$

$IG =$ E(parent) – weighted average of E(children)

b.1. Entropy of child

Entropy of right child node, E(Speed limit=No) = 0

Entropy of left child node, E(Speed limit=Yes) = 0

b.2. Calculate the weighted average of E(child) for Speed limit

Parent node: 4
Right child: 2
Left child: 2

weighted average of E(child) = (no. of left_child) / (no. of parent) * E(Speed limit=Yes) +
(no. of right_child) / (no. of parent) * E(Speed limit=No)
= (2/4) * 0 + (2/4) *0
= 0

$$IG = \text{E(parent)} - \text{weighted average of E(children)}$$

Which input variable can be used as root node?

Answer

The predictor variable have the higher information gain can be set as root node and then the root node can be further split.

b.3. Calculate the IG(Road type)

IG(Road type) = 1 – 0.675 = 0.325

c. Using the same methodology, calculate IG for other predictor variables

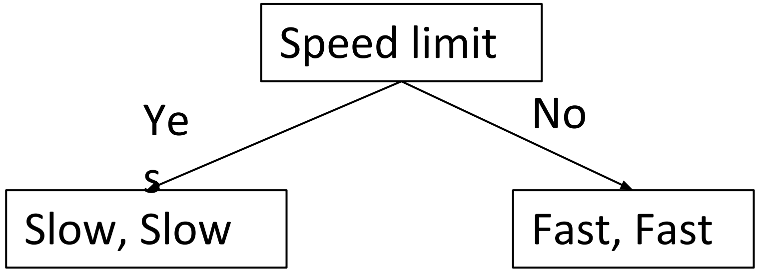
IG(Road type) = 1 – 0.675 = 0.325

IG(Obstruction) = 1 – 1 = 0

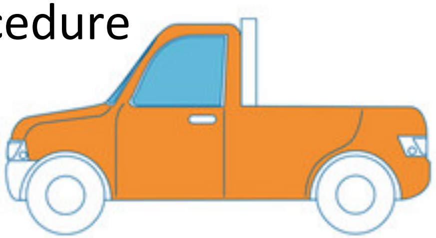
IG(Speed limit) = 1 – 0 = 1

Step 2: Assign best variable as a decision variable for the root node

Road type	Obstruction	Speed limit	Speed
Steep	Yes	Yes	Slow
Steep	No	Yes	Slow
Flat	Yes	No	Fast
Steep	No	No	Fast



Step 3: For each value of root node, build a descend of the node using the above procedure



Continue the procedure to build the complete decision tree until if data is correctly classified.

Problem statement: For a given data set create a decision tree and predict if John would play golf or not.

Attributes

Classes

Outlook	Temperature	Humidity	Windy	Play Golf
Rainy	Hot	High	FALSE	No
Rainy	Hot	High	TRUE	No
Overcast	Hot	High	FALSE	Yes
Sunny	Mild	High	FALSE	Yes
Sunny	Cool	Normal	FALSE	Yes
Sunny	Cool	Normal	TRUE	No
Overcast	Cool	Normal	TRUE	Yes
Rainy	Mild	High	FALSE	No
Rainy	Cool	Normal	FALSE	Yes
Sunny	Mild	Normal	FALSE	Yes
Rainy	Mild	Normal	TRUE	Yes
Overcast	Mild	High	TRUE	Yes
Overcast	Hot	Normal	FALSE	Yes
Sunny	Mild	High	TRUE	No

Predictor variables: Outlook, Temperature, Humidity, Windy
Target variable: Play Golf

Step 1: Decide the **best variable** that splits the data set. i.e root node of decision tree

Calculate entropy (E) and information gain (IG)

$$E = -\sum_{i}^{C} p_i log_2 p_i$$

$$H(S) = \sum_{x \in X} p(x) \log_2 \frac{1}{p(x)}$$

IG = E(parent) – weighted average of E(children)

Play Golf
No
No
Yes
Yes
Yes
No
Yes
No
Yes
Yes
Yes
Yes
Yes
No

a. Calculate entropy of parent node

Target variable is Play Golf

It is the Parent node

Two classes: Yes and No

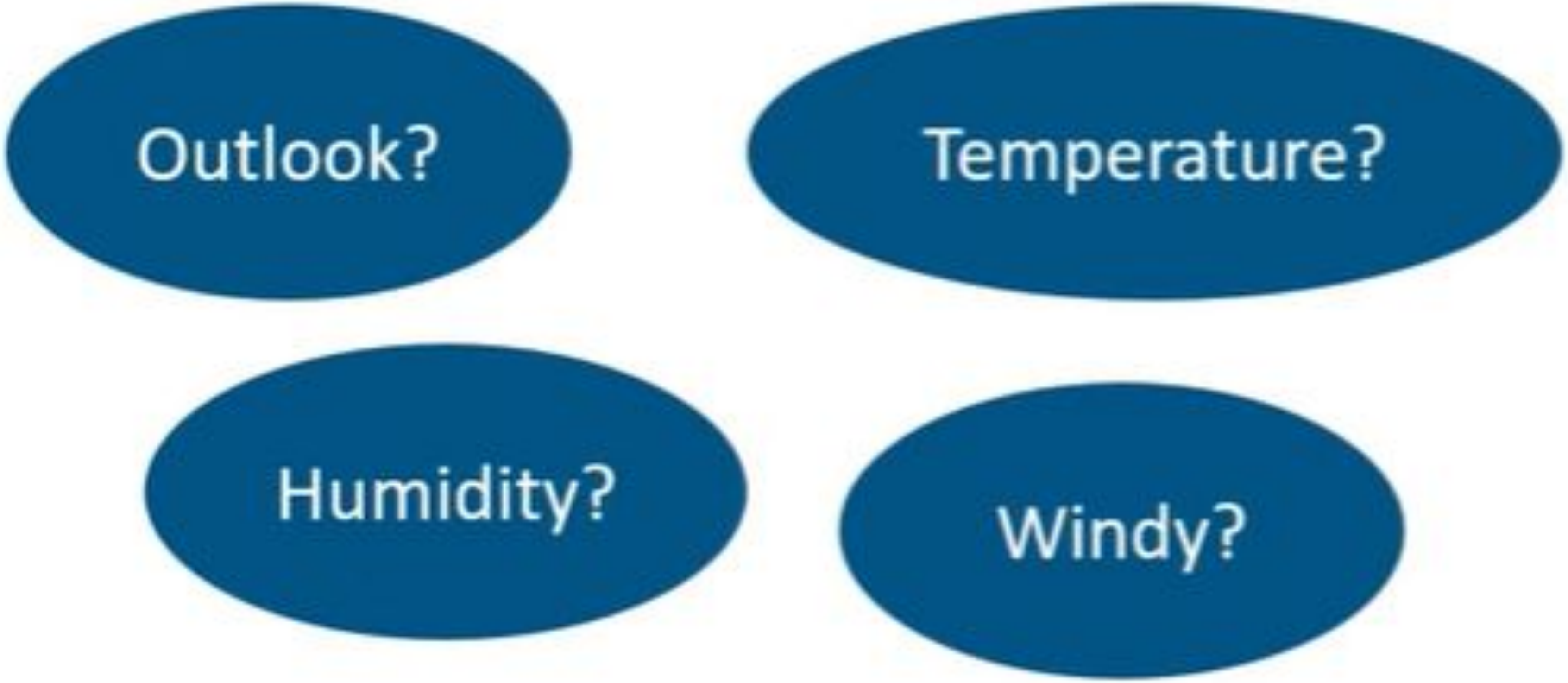
P(Yes) = 9/14

P(No) = 5/14

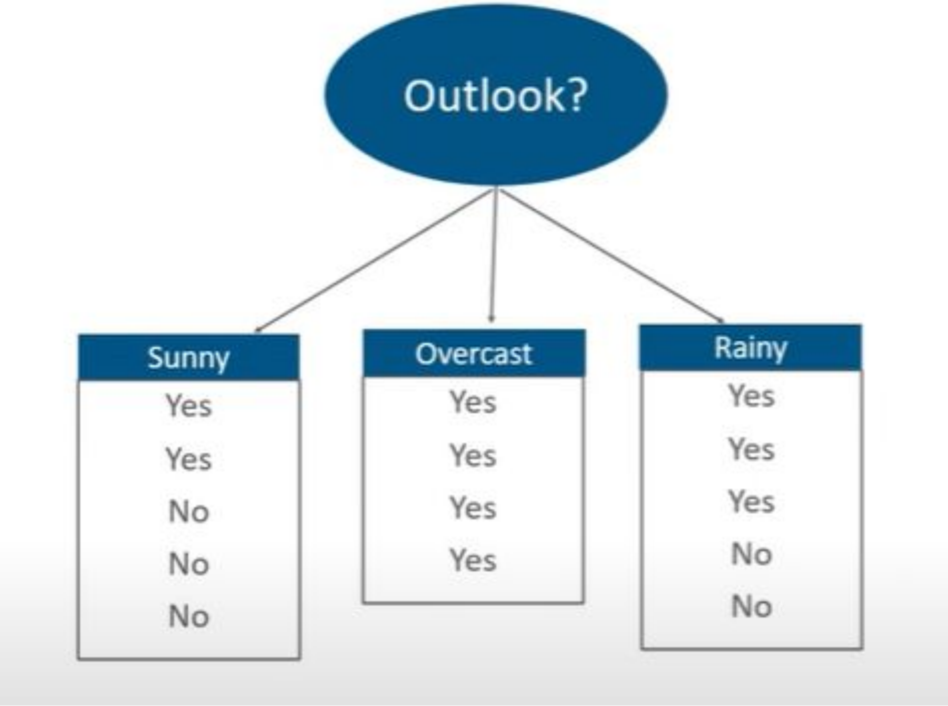
E(Parent) = - (P(Yes)*log2(P(Yes)) + P(No)*log2(P(No)))

= - ((9/14)*log2(9/14) + (5/14)*log2(5/14))

= 0.41+0.53 = 0.94


Outlook?
Temperature?
Humidity?
Windy?

b. Information gain of child node (Outlook)



$E(Outlook = Sunny) = -2/5 \log_2 2/5 - 3/5 \log_2 3/5 = 0.971$

$E(Outlook = Overcast) = -1 \log_2 1 - 0 \log_2 0 = 0$

$E(Outlook = Sunny) = -3/5 \log_2 3/5 - 2/5 \log_2 2/5 = 0.971$

Information from outlook,

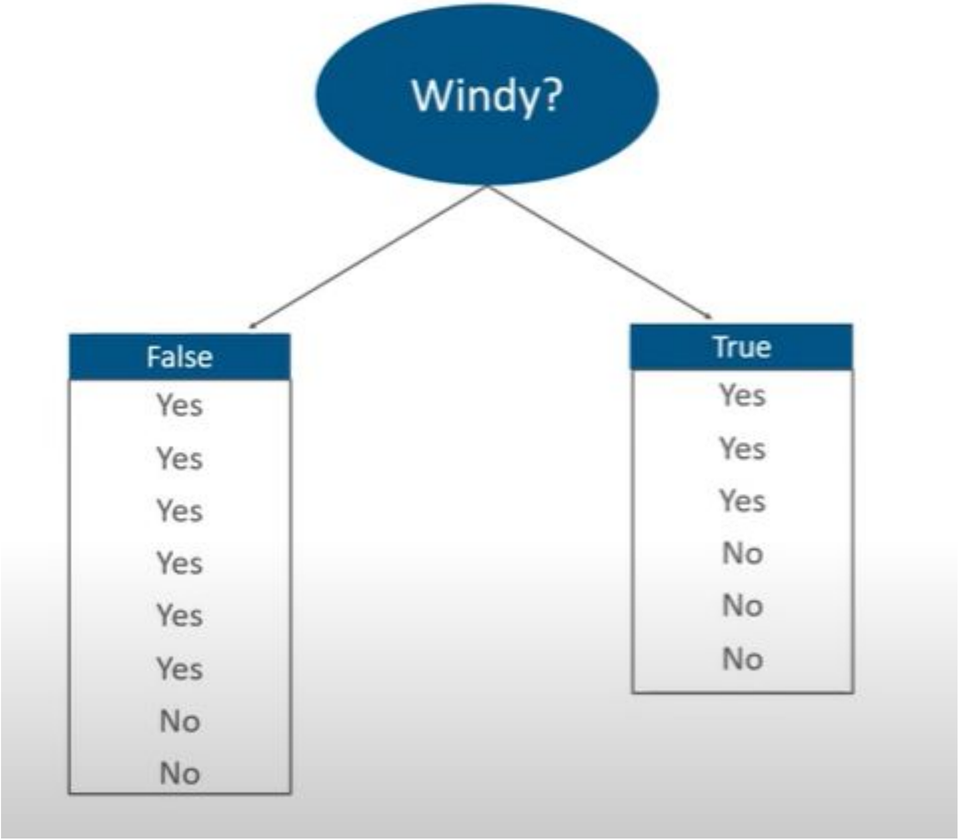
$I(Outlook) = 5/14 \times 0.971 + 4/14 \times 0 + 5/14 \times 0.971 = 0.693$

Information gained from outlook,

$Gain(Outlook) = E(S) - I(Outlook)$

$\mathbf{0.94 - 0.693 = 0.247}$

b. Information gain of child node (Windy)



E(Windy=True) = -(3/6)log(3/6) –(3/6)log(3/6) = 1

E(Windy=False) = -(6/8)log(6/8) –(2/8)log(2/8) = 0.811

Information from windy,

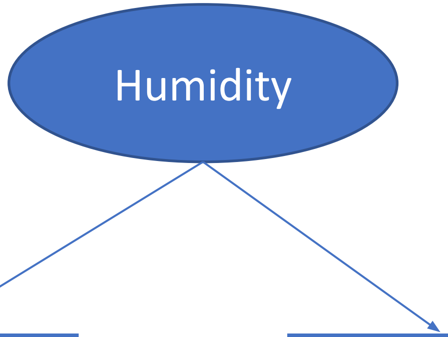
I(Windy) = 8/14 x 0.811+ 6/14 x 1 = 0.892

Information gained from outlook,

Gain(Windy) = E(S) – I(Windy)

0.94 – 0.892 = 0.048

b. Information gain of child node (Humidity)



High	Normal
No	Yes
No	No
Yes	Yes
Yes	Yes
No	Yes
Yes	Yes
No	Yes

E(Humidity=High) = -(3/7)log(3/7) –(4/7)log(4/7) = 0.985

E(Humidity=Normal) = -(6/7)log(6/7) –(1/7)log(1/7) = 0.592

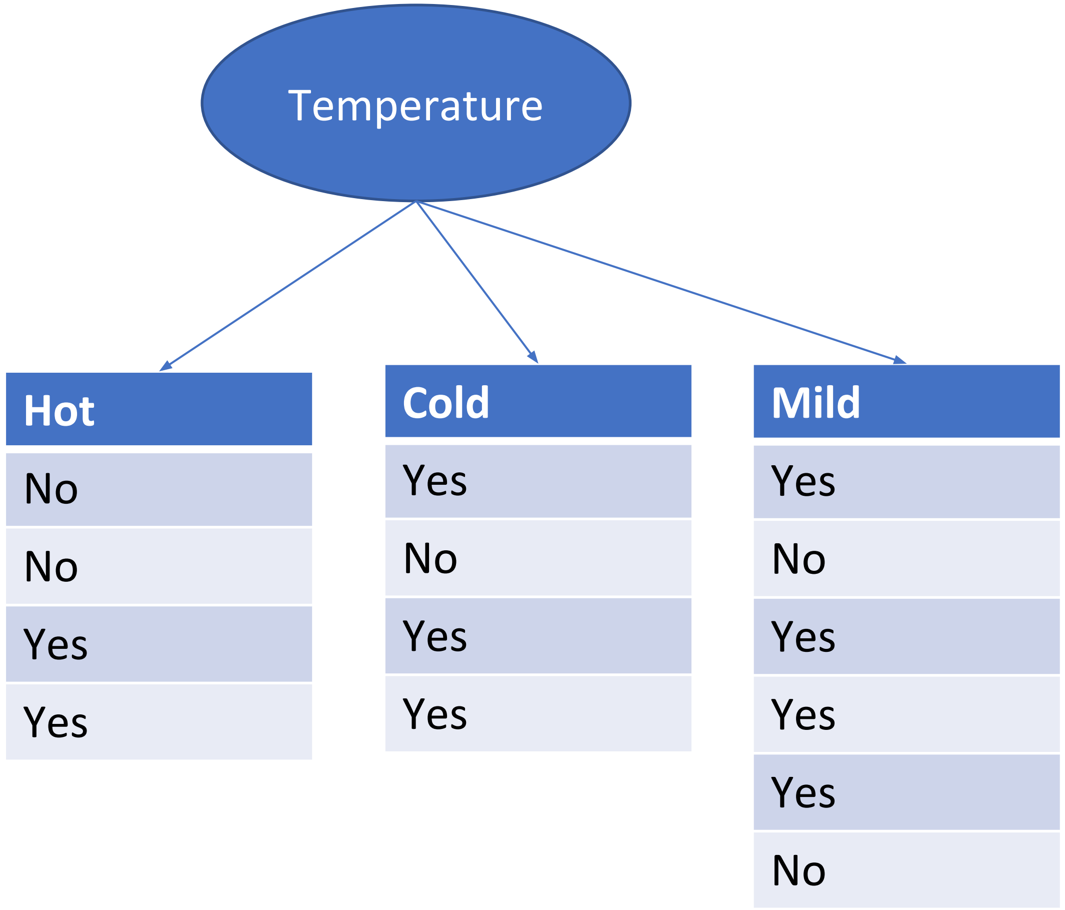
Information from Humidity,

$$E\,(\text{PlayGolf, Humidity}) = 7/14 * 0.985 + 7/14 * 0.592 = 0.788$$

Information gained from Humidity,

Gain(Humidity) = **0.94-0.788 = 0.152**

b. Information gain of child node (Temperature)



E(Temperature=Hot) = -(2/4)log(2/4) –(2/4)log(2/4) = 1

E(Temperature=Cold) = -(3/4)log(3/4) –(1/4)log(1/4) = 0.811

E(Temperature=Mild) = -(4/6)log(4/6) –(2/6)log(2/6) = 0.918

Information from Temperature,

Information gained from Temperature

Gain(Temperature) = **0.94-0.911 = 0.029**

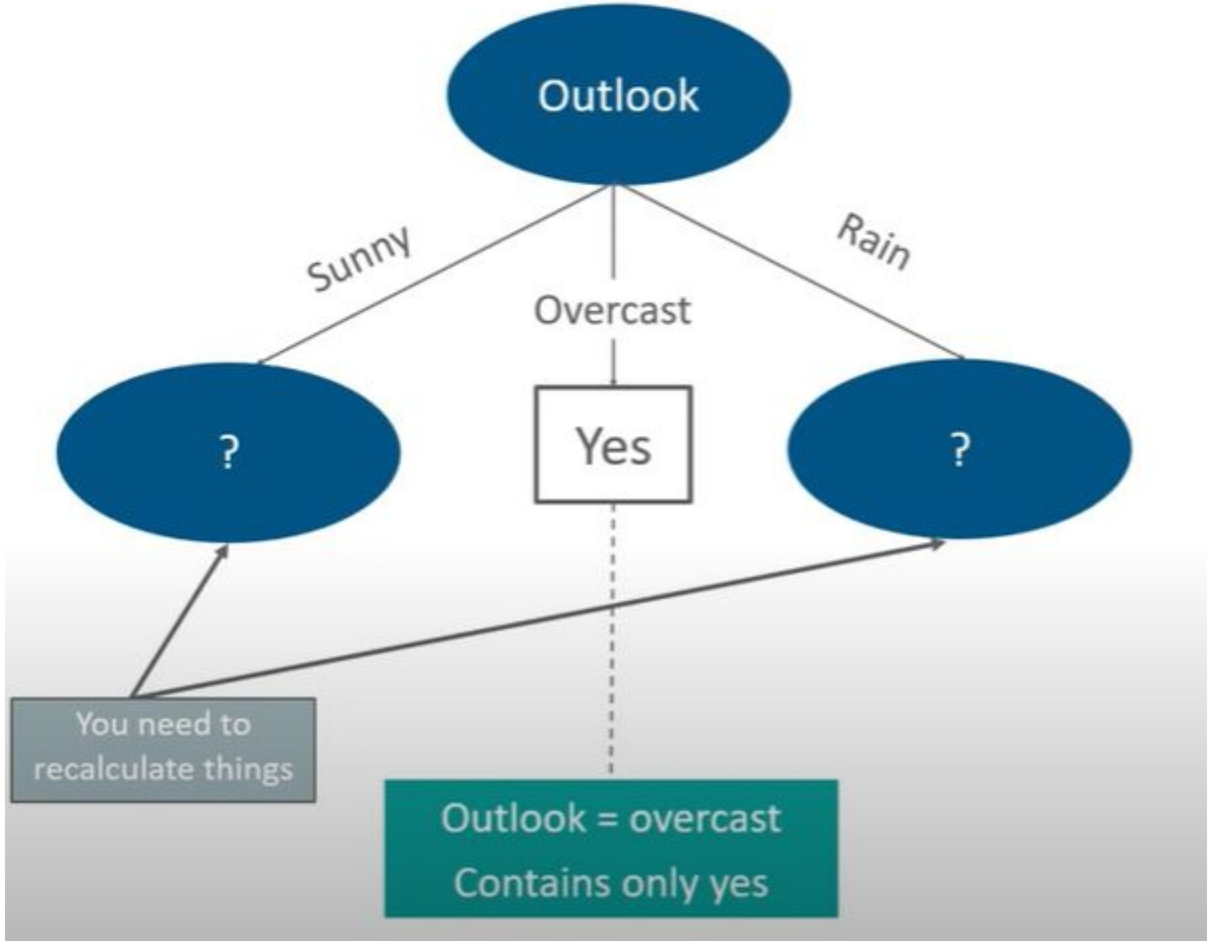
Outlook:
Info 0.693
Gain: 0.940-0.693 0.247

Temperature:
Info 0.911
Gain: 0.940-0.911 0.029

Humidity:
Info 0.788
Gain: 0.940-0.788 0.152

Windy:
Info 0.892
Gain: 0.940-0.982 0.048

outlook	temp.	humidity	windy	play
sunny	hot	high	false	no
sunny	hot	high	true	no
overcast	hot	high	false	yes
rainy	mild	high	false	yes
rainy	cool	normal	false	yes
rainy	cool	normal	true	no
overcast	cool	normal	true	yes
sunny	mild	high	false	no
sunny	cool	normal	false	yes
rainy	mild	normal	false	yes
sunny	mild	normal	true	yes
overcast	mild	high	true	yes
overcast	hot	normal	false	yes
rainy	mild	high	true	no



outlook	temp.	humidity	windy	play
sunny	hot	high	false	no
sunny	hot	high	true	no
overcast	hot	high	false	yes
rainy	mild	high	false	yes
rainy	cool	normal	false	yes
rainy	cool	normal	true	no
overcast	cool	normal	true	yes
sunny	mild	high	false	no
sunny	cool	normal	false	yes
rainy	mild	normal	false	yes
sunny	mild	normal	true	yes
overcast	mild	high	true	yes
overcast	hot	normal	false	yes
rainy	mild	high	true	no

Step 3: Choosing the splitting attribute that has the highest Information Gain

$$\mathrm{Gain}(D\,,\boldsymbol{Outlook}) = \mathbf{0.2468} \quad > \quad \mathrm{Gain}(D\,,Humidity) = 0.1518 \quad > \quad \mathrm{Gain}(D\,,Wind) = 0.048$$

Splitting Attribute = Outlook

Outlook ?

sunny

Day	Outlook	Humidity	Wind	Play
1	Sunny	High	Weak	No
2	Sunny	High	Strong	No
8	Sunny	High	Weak	No
9	Sunny	Normal	Weak	Yes
11	Sunny	Normal	Strong	Yes

cloudy

Yes

Day	Outlook	Humidity	Wind	Play
3	Cloudy	High	Weak	Yes
7	Cloudy	Normal	Strong	Yes
12	Cloudy	High	Strong	Yes
13	Cloudy	Normal	Weak	Yes

rain

Day	Outlook	Humidity	Wind	Play
4	Rain	High	Weak	Yes
5	Rain	Normal	Weak	Yes
6	Rain	Normal	Strong	No
10	Rain	Normal	Weak	Yes
14	Rain	High	Strong	No

Repeat from step 2-3 for the branch "Sunny"

Calculate entropy for the branch

$$E(sunny) = \sum_{i=1}^{c} -p_i \log_2 p_i$$

$$E(sunny) = -{}^{2}/_{5} \log_2 {}^{2}/_{5} - {}^{3}/_{5} \log_2 {}^{3}/_{5}$$

$$E(sunny) = 0.9709$$

Day	Outlook	Humidity	Wind	Play
1	Sunny	High	Weak	No
2	Sunny	High	Strong	No
8	Sunny	High	Weak	No
9	Sunny	Normal	Weak	Yes
11	Sunny	Normal	Strong	Yes

Class	p_i
Yes	2/5
No	3/5

Calculate entropy and Information Gain for every possible Attribute : Humidity, Wind

$$E(D,X) = \sum_{c \in X} P(c)E(c)$$

Day	Humidity	Wind	Play
1	High	Weak	No
2	High	Strong	No
8	High	Weak	No
9	Normal	Weak	Yes
11	Normal	Strong	Yes

$$E(sunny, Humidity) = {}^{3}/_{5}\left(-{}^{3}/_{3} \log_2 {}^{3}/_{3}\right) + {}^{2}/_{5}\left(-{}^{2}/_{2} \log_2 {}^{2}/_{2}\right)$$

$$E(sunny, Humidity) = 0 \qquad E(sunny) = 0.9709$$

$$\text{Gain}(sunny, Humidity) = E(sunny) - \text{E}(sunny, Humidity)$$

$$\text{Gain}(sunny, Humidity) = 0.9709 - 0$$

$$\text{Gain}(sunny, Humidity) = 0.9709$$

$$E(sunny, Wind) = {}^{3}/_{5}\left(-{}^{1}/_{3} \log_2 {}^{1}/_{3} - {}^{2}/_{3} \log_2 {}^{2}/_{3}\right) + {}^{2}/_{5}\left(-{}^{1}/_{2} \log_2 {}^{1}/_{2} - {}^{1}/_{2} \log_2 {}^{1}/_{2}\right)$$

$$E(sunny, Wind) = 0.5508 + 0.4$$

$$E(sunny, Wind) = 0.9508 \qquad E(sunny) = 0.9709$$

$$\text{Gain}(sunny, Wind) = E(sunny) - \text{E}(sunny, Wind)$$

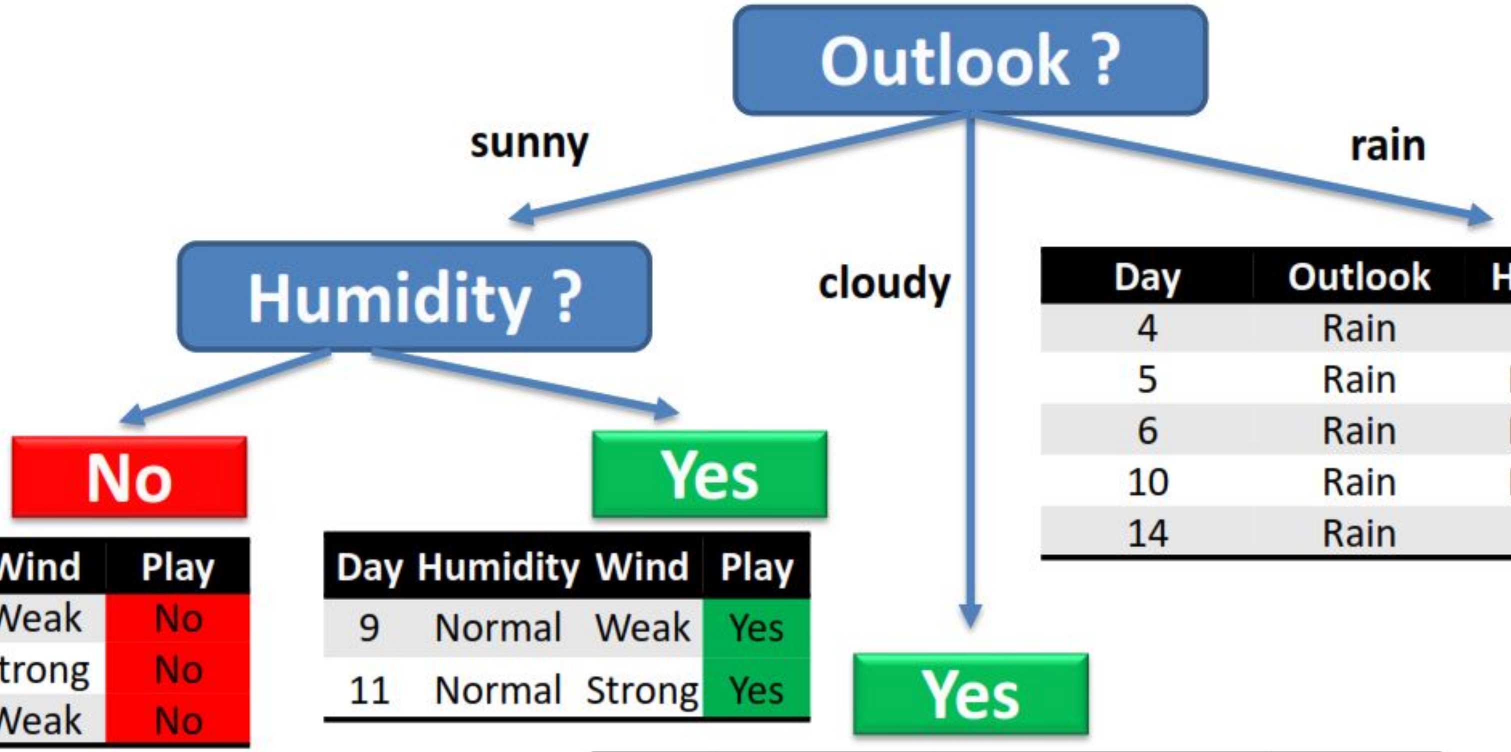
$$\text{Gain}(sunny, Wind) = 0.9709 - 0.9508$$

$$\text{Gain}(sunny, Wind) = 0.02$$

Choosing the splitting attribute that has the highest Information Gain

$$\text{Gain}(sunny\,, Humidity) = 0.9709 \quad > \quad \text{Gain}(sunny\,, Wind) = 0.02$$

Splitting Attribute = Humidity



Day	Humidity	Wind	Play
1	High	Weak	No
2	High	Strong	No
8	High	Weak	No

Day	Humidity	Wind	Play
9	Normal	Weak	Yes
11	Normal	Strong	Yes

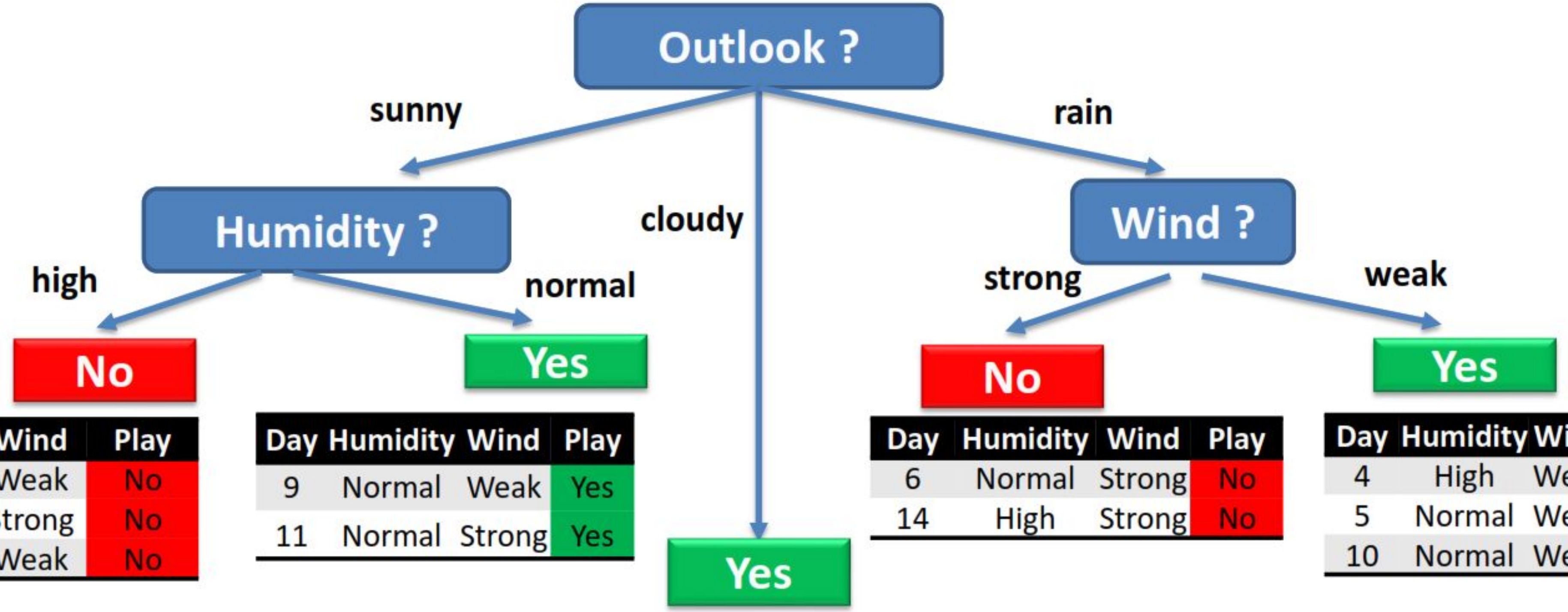
Day	Outlook	Humidity	Wind	Play
3	Cloudy	High	Weak	Yes
7	Cloudy	Normal	Strong	Yes
12	Cloudy	High	Strong	Yes
13	Cloudy	Normal	Weak	Yes

Day	Outlook	Humidity	Wind	Play
4	Rain	High	Weak	Yes
5	Rain	Normal	Weak	Yes
6	Rain	Normal	Strong	No
10	Rain	Normal	Weak	Yes
14	Rain	High	Strong	No

Repeat from step 2-3 for the branch "Rain"

$$\text{Gain}(rain\ , wind) = 0.9709 \quad > \quad \text{Gain}(rain\ , Humidity) = 0.0201$$

Splitting Attribute = Wind



Day	Humidity	Wind	Play
1	High	Weak	No
2	High	Strong	No
8	High	Weak	No

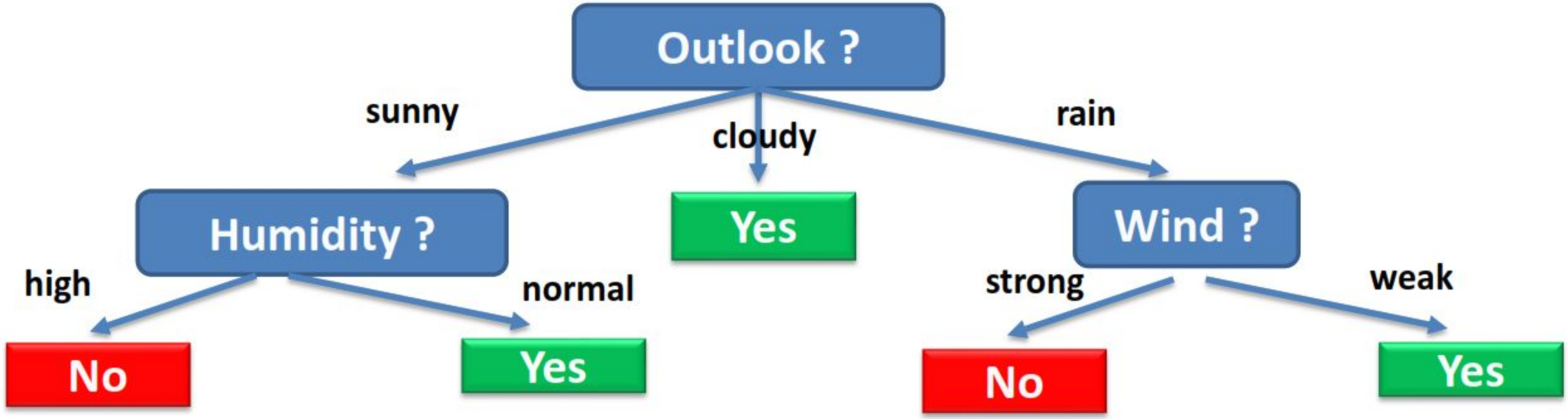
Day	Humidity	Wind	Play
9	Normal	Weak	Yes
11	Normal	Strong	Yes

Day	Outlook	Humidity	Wind	Play
3	Cloudy	High	Weak	Yes
7	Cloudy	Normal	Strong	Yes
12	Cloudy	High	Strong	Yes
13	Cloudy	Normal	Weak	Yes

Day	Humidity	Wind	Play
6	Normal	Strong	No
14	High	Strong	No

Day	Humidity	Wind	Play
4	High	Weak	Yes
5	Normal	Weak	Yes
10	Normal	Weak	Yes

Decision tree



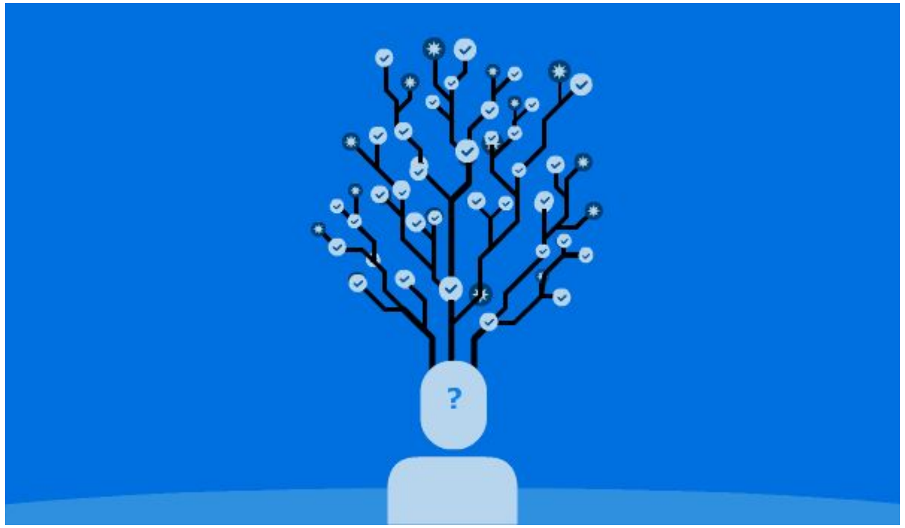
Advantages of the Decision Tree

- It is simple to understand as it follows the same process which a human follow while making any decision in real-life.
- This can be used for both classification and regression problems.
- Decision Trees can handle both continuous and categorical variables.
- No feature scaling required. (Information based & Probability based algorithms)
- Handles nonlinear parameters efficiently.
- Decision tree can automatically handle missing values and outliers.
- Less Training Period.

Disadvantages of the Decision Tree

- The decision tree contains lots of layers, which makes it complex.
- It may have an overfitting issue, which can be resolved using the Random Forest algorithm.
- High variance, leads to many errors in the final predictions and shows high inaccuracy in the results.
- For more class labels, the computational complexity of the decision tree may increase.
- Not suitable for large datasets and unstable.

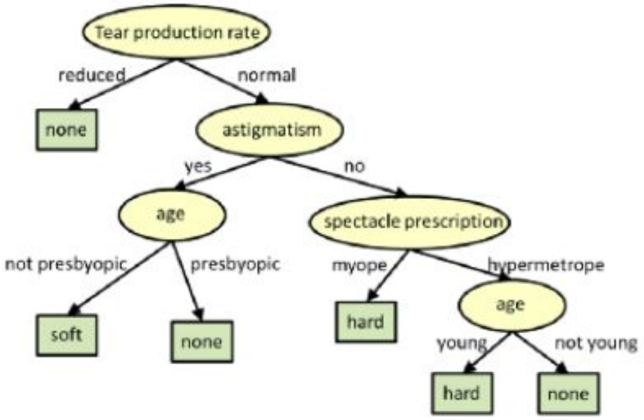
How to avoid/counter Overfitting in Decision Trees?



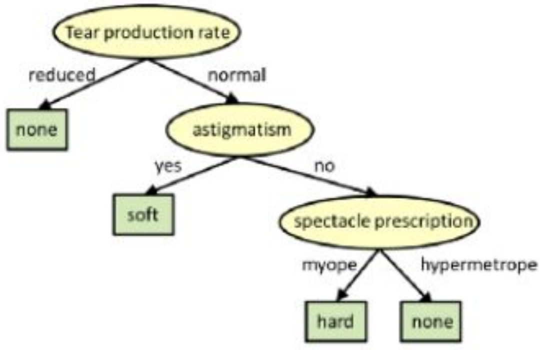
Here are two ways to remove overfitting:

- Pruning Decision Trees
- Random Forest

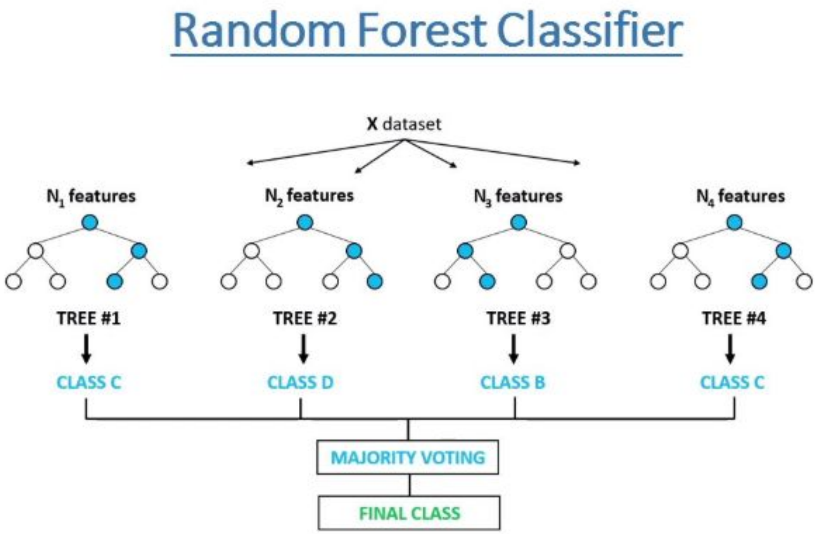
Pruning Decision Trees



Original Tree



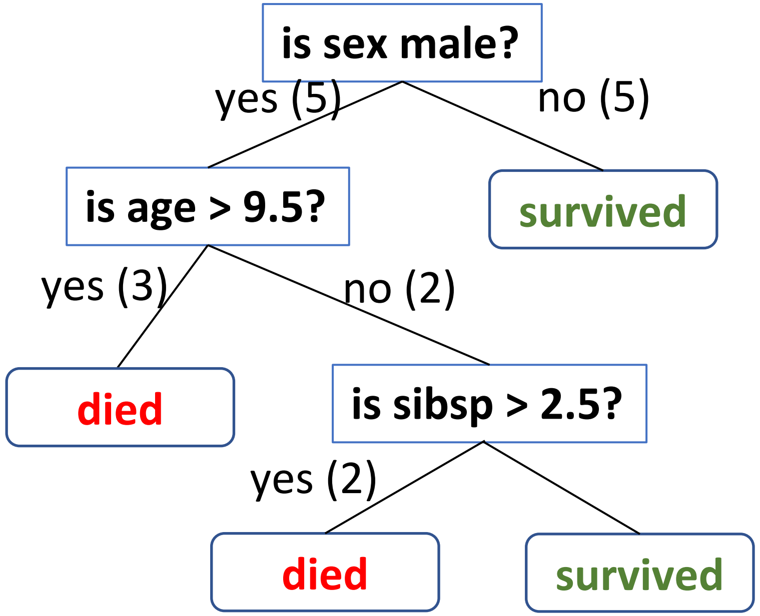
Pruned Tree

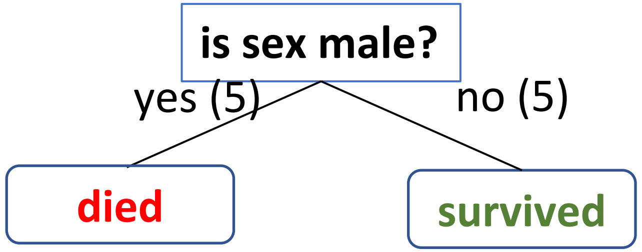


Problem statement:

Uses titanic data set for predicting whether a passenger will survive or not.

	survived	pclass	sex	age	sibsp	parch	fare	embarked	class	who	adult_male	deck	embark_town
0	0	3	male	22.0	1	0	7.2500	S	Third	man	True	NaN	Southampton
1	1	1	female	38.0	1	0	71.2833	C	First	woman	False	C	Cherbourg
2	1	3	female	26.0	0	0	7.9250	S	Third	woman	False	NaN	Southampton
3	1	1	female	35.0	1	0	53.1000	S	First	woman	False	C	Southampton
4	0	3	male	35.0	0	0	8.0500	S	Third	man	True	NaN	Southampton
5	0	3	male	NaN	0	0	8.4583	Q	Third	man	True	NaN	Queenstown
6	0	1	male	54.0	0	0	51.8625	S	First	man	True	E	Southampton
7	0	3	male	2.0	3	1	21.0750	S	Third	child	False	NaN	Southampton
8	1	3	female	27.0	0	2	11.1333	S	Third	woman	False	NaN	Southampton
9	1	2	female	14.0	1	0	30.0708	C	Second	child	False	NaN	Cherbourg





Pruning

- Construct decision tree model uses 3 features/attributes/columns from the data set, namely sex, age and sibsp (number of spouses or children along).

Python Implementation

```
# importing libraries
from sklearn.neighbors import DecisionTreeClassifier
# load data

# split data into training and testing

# fit decision tree classifier model to training dataset
dt=DecisionTreeClassifier(random_state=0,criterion="entropy")
dt.fit(X_train,y_train)

# predicting result with testing dataset
y_pred=dt.predict(x_test)
```

rec	Age	Income	Student	Credit _rating	Buys_computer
r1	<=30	Hight	No	Fair	No
r2	<=30	Hight	No	Excellent	No
r3	31...40	Hight	No	Fair	Yes
r4	>40	Medium	No	Fair	Yes
r5	>40	Low	Yes	Fair	Yes
r6	>40	Low	Yes	Excellent	No
r7	31...40	Low	Yes	Excellent	Yes
r8	<=30	Medium	No	Fair	No
r9	,=30	Low	Yes	Fair	Yes
r10	>30	Medium	Yes	Fair	Yes
r11	<=30	Medium	Yes	Excellent	Yes
r12	31...40	Medium	No	Excellent	Yes
r13	31...40	High	Yes	Fair	Yes
r14	>40	Medium	No	Excellent	No

https://www.edureka.co/blog/decision-trees/