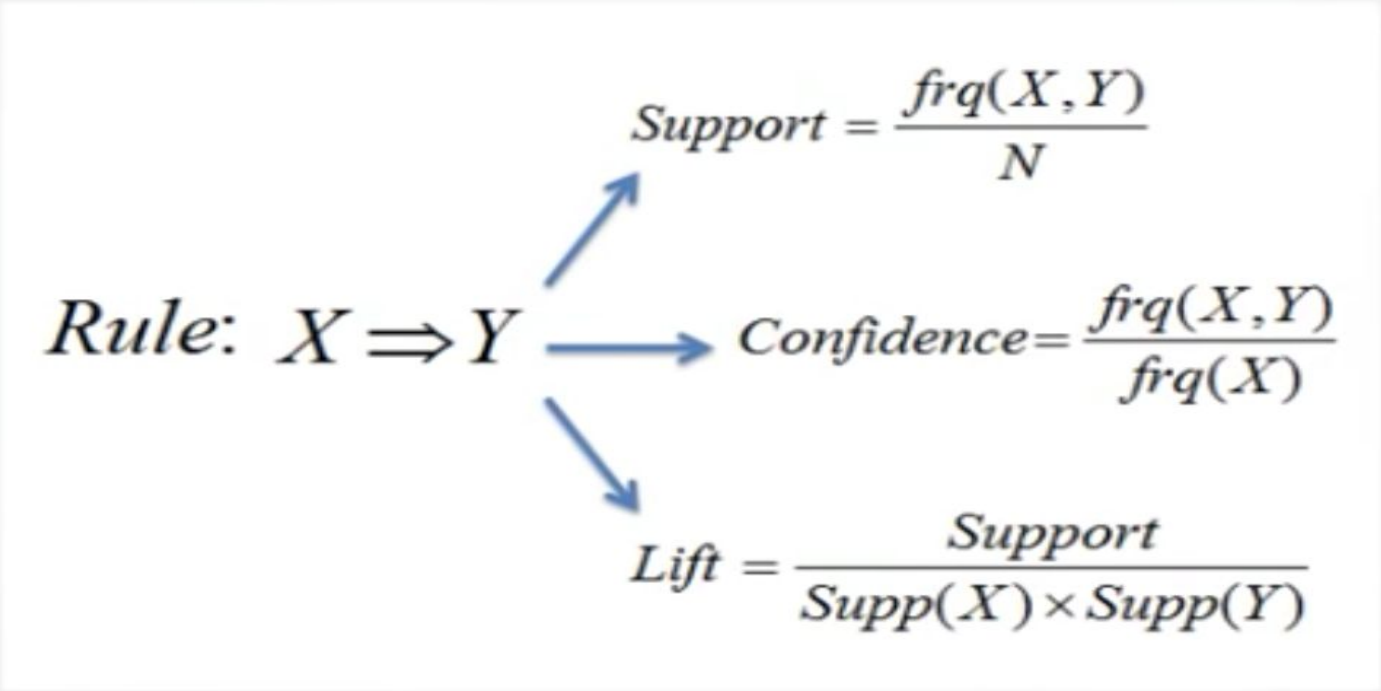
Odds and Ends
--- Association Rules

Association Rules

- An association rule is a pattern that states when *an event* occurs, *another event* occurs with certain probability
- ARs are if/then statements that help discover relationships between unrelated data in a data repository
  - i.e. to find the relationships between the objects which are frequently used together
- Association Rules find all sets of items (itemsets) that have support greater than the minimum support
  - then using the large itemsets to generate the desired rules that have confidence greater than the minimum confidence
- A typical and widely used example of association rules application is market basket analysis

Association Rules

- For example, if the customer buys milk then he may also buy cereal, or, If the customer buys a tablet computer then he may also buy a case (cover)
- There are two basic criteria that association rules use, support and confidence
  - to identify the relationships and rules generated by analyzing data for frequently used if/then patterns
- Association rules are usually needed to satisfy a user-specified minimum support and a user-specified minimum confidence at the same time


Rule: $X \Rightarrow Y$
$Support = \frac{frq(X,Y)}{N}$
$Confidence = \frac{frq(X,Y)}{frq(X)}$
$Lift = \frac{Support}{Supp(X) \times Supp(Y)}$

Concepts

- *A set of all items in a store* $I = \{i_1, i_2, \ldots, i_m\}$
- A set of all transactions (Transaction Database T)
  - $T = \{t_1, t_2, \ldots, t_N\}$
- Each t_i is a set of items s.t. $t \subseteq I$
- Each transaction t_i has a transaction ID (TID)

Rule	Support	Confidence	Lift
$A \Rightarrow D$	2/5	2/3	10/9
$C \Rightarrow A$	2/5	2/4	5/6
$A \Rightarrow C$	2/5	2/3	5/6
$B \& C \Rightarrow D$	1/5	1/3	5/9

- *Support* of the association rule $X \rightarrow Y$:

$$\text{Support}(X, Y) \equiv P(X, Y) = \frac{\#\{\text{customers who bought } X \text{ and } Y\}}{\#\{\text{customers}\}}$$

- *Confidence* of the association rule $X \rightarrow Y$:

$$\begin{aligned} \text{Confidence}(X \rightarrow Y) \equiv P(Y|X) &= \frac{P(X, Y)}{P(X)} \\ &= \frac{\#\{\text{customers who bought } X \text{ and } Y\}}{\#\{\text{customers who bought } X\}} \end{aligned}$$

- *Lift*, also known as *interest* of the association rule $X \rightarrow Y$:

$$\text{Lift}(X \rightarrow Y) = \frac{P(X, Y)}{P(X)P(Y)} = \frac{P(Y|X)}{P(Y)}$$

If the lift is more than 1, we can say that X makes Y more likely, and if the lift is less than 1, having X makes Y less likely.

Odds and Ends
--- Decision Tree

Decision Tree Learning Algorithm

- There are many specific decision-tree algorithms. Notable ones include:
- ID3 (Iterative Dichotomiser 3)
- C4.5 (successor of ID3)
- CART (Classification And Regression Tree)
- CHAID (Chi-squared Automatic Interaction Detector). Performs multi-level splits when computing classification trees.
- MARS: extends decision trees to handle numerical data better.

Use Case – Loan Repayment prediction

PROBLEM STATEMENT

TO PREDICT IF A CUSTOMER WILL REPAY LOAN AMOUNT OR NOT USING DECISION TREE ALGORITHM IN PYTHON

ID3 Algorithm

- ID3 is one of the most common decision tree algorithm
- Dichotomisation means dividing into two completely opposite things.
- Algorithm iteratively divides attributes into two groups which are the most dominant attribute and others to construct a tree.
- Then, it calculates the Entropy and Information Gains of each attribute. In this way, the most dominant attribute can be founded.
- After then, the most dominant one is put on the tree as decision node.
- Entropy and Gain scores would be calculated again among the other attributes.
- Procedure continues until reaching a decision for that branch.

ID3 Algorithm

- Calculate the entropy of every attribute using the data set S
- $\text{Entropy}(S) = \sum - p(I) \cdot \log_2 p(I)$
- Split the set S into subsets using the attribute for which the resulting entropy (after splitting) is minimum (or, equivalently, information gain is maximum)
- $\text{Gain}(S, A) = \text{Entropy}(S) - \sum [\, p(S|A) \cdot \text{Entropy}(S|A) \,]$
- Make a decision tree node containing that attribute
- Recurse on subsets using remaining attributes.

Decision Tree – Important Terms

ENTROPY

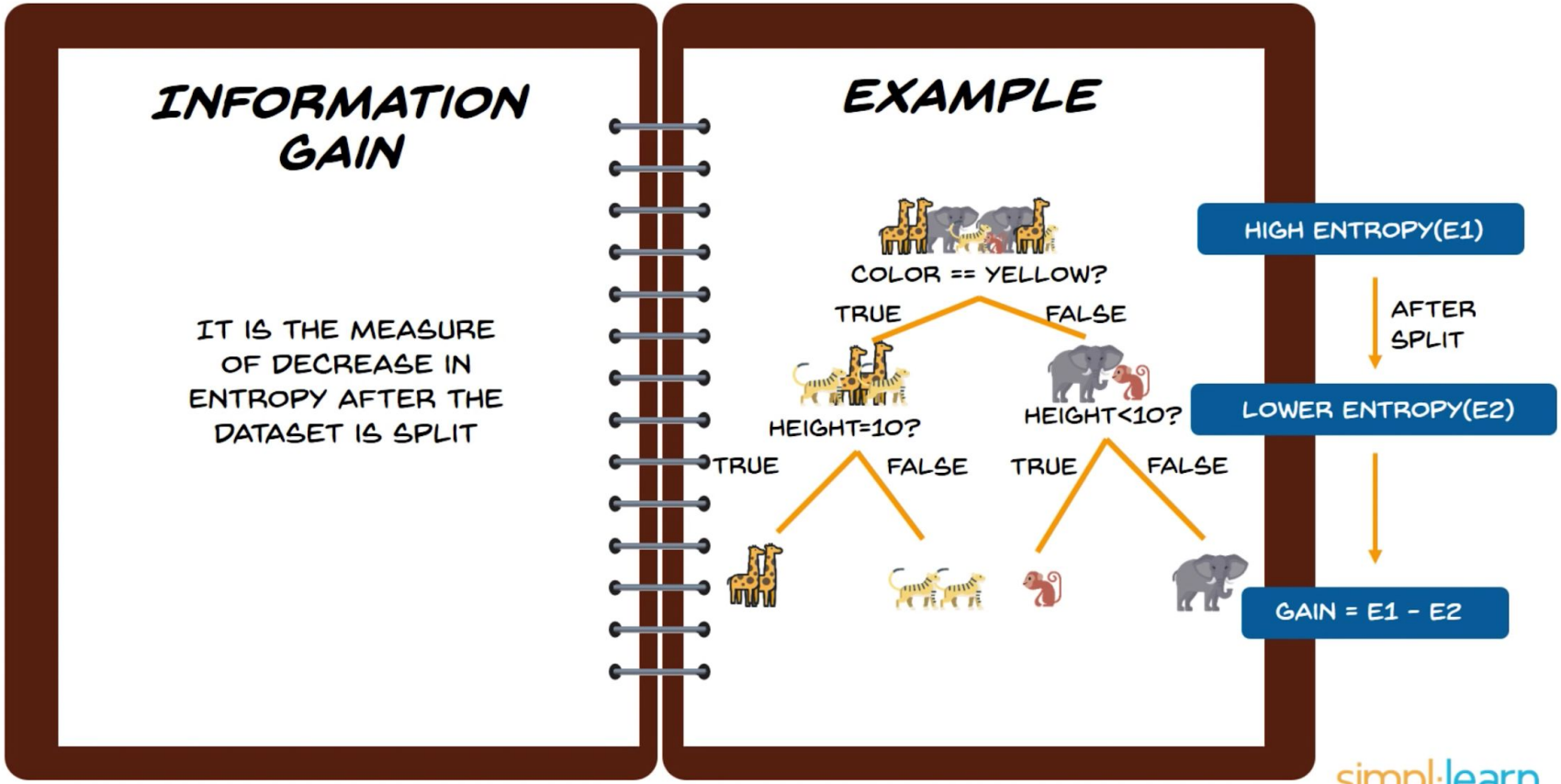
ENTROPY IS THE MEASURE OF RANDOMNESS OR UNPREDICTABILITY IN THE DATASET

EXAMPLE

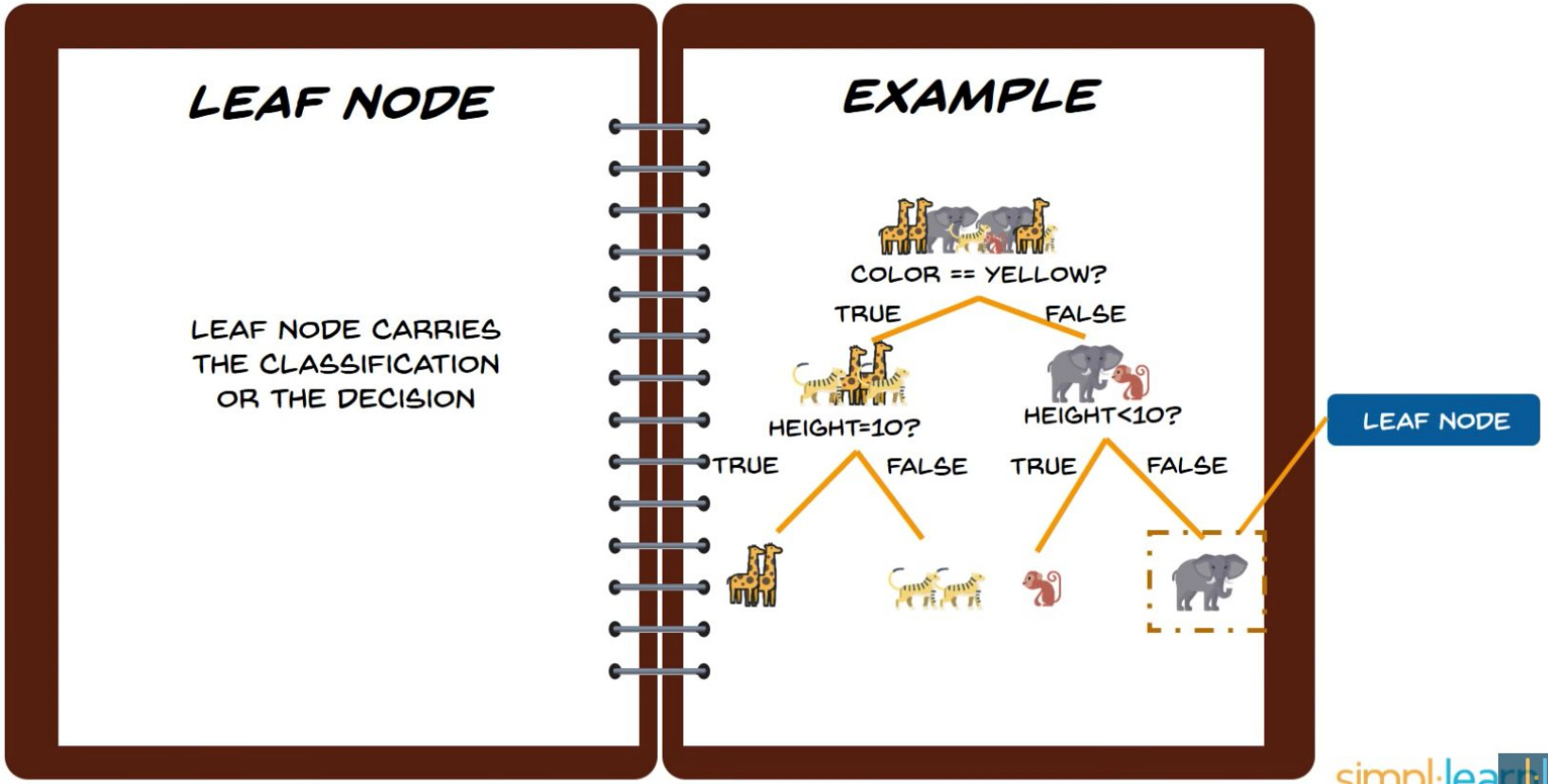
HIGH ENTROPY

THIS DATASET HAS A VERY HIGH ENTROPY

Decision Tree – Important Terms



Decision Tree – Important Terms



IN SIMPLe TERMS: Procedure to calculate the condition that gives us the highest gain

Goal: To find the attribute that will split the data so that the information gain is the highest.

- Initially, calculate the Entropy of the dataset for every attribute.
- Split the dataset based on attribute that gives the highest entropy value.
- After every split and calculate the gain. ‘Gain’ is the difference between the previous Entropy value and the current Entropy after split.
- The condition that gives us the highest gain will be used to make the next splits

How does a Decision Tree work?

PROBLEM STATEMENT

TO CLASSIFY THE DIFFERENT TYPES OF ANIMALS BASED ON THEIR FEATURES USING DECISION TREE

TRAINING DATASET

COLOR	HEIGHT	LABEL
GREY	10	ELEPHANT
YELLOW	10	GIRAFFE
BROWN	3	MONKEY
GREY	10	ELEPHANT
YELLOW	4	TIGER

NOW WE WILL TRY TO CHOOSE A CONDITION THAT GIVES US THE HIGHEST GAIN

NOTE

GAIN IS THE MEASURE OF DECREASE IN ENTROPY AFTER SPLITTING

FORMULA FOR ENTROPY

$$\sum_{i=1}^{k} P(valuei).log_2(P(value_i)$$

ENTROPY = $(\frac{3}{8})\ log_2(\frac{3}{8}) + (\frac{2}{8})\ log_2(\frac{2}{8}) + (\frac{1}{8})\ log_2(\frac{1}{8}) + (\frac{2}{8})\ log_2(\frac{2}{8})$

ENTROPY=0.571

CONDITIONS

COLOR== YELLOW?

HEIGHT>=10

COLOR== BROWN?

COLOR==GREY

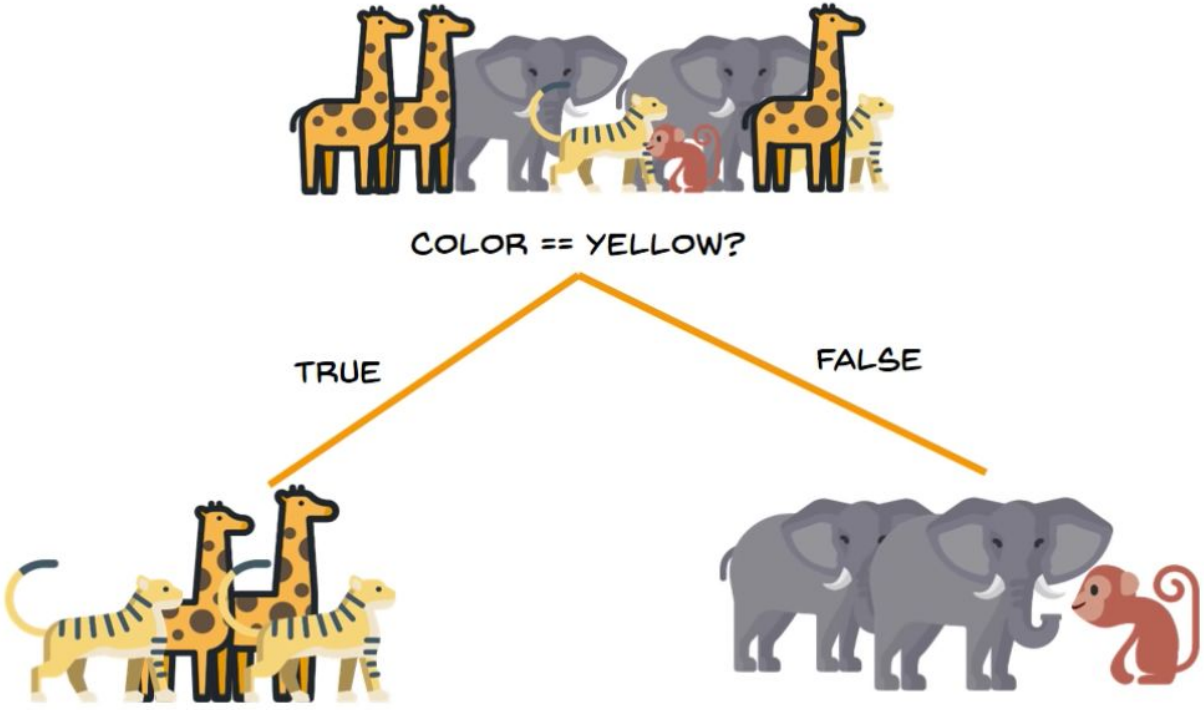
DIAMETER<10

LET'S SAY THIS CONDITION GIVES US THE MAXIMUM GAIN

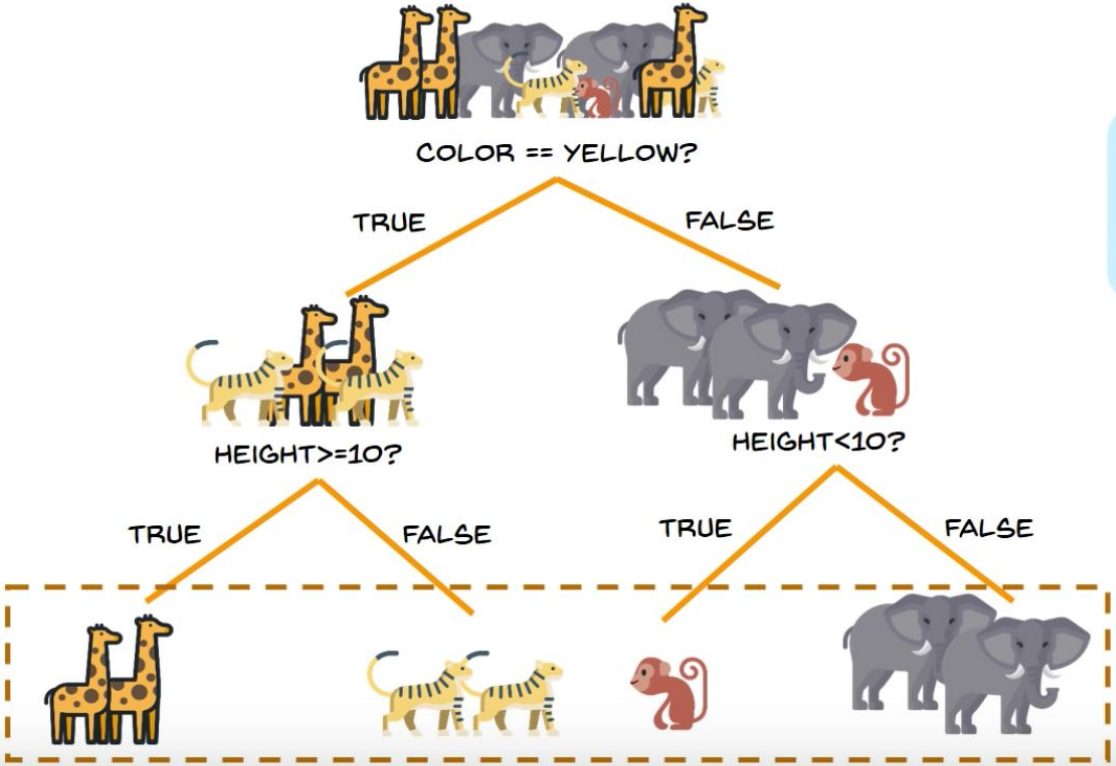
TRAINING DATASET

COLOR	HEIGHT	LABEL
GREY	10	ELEPHANT
YELLOW	10	GIRAFFE
BROWN	3	MONKEY
GREY	10	ELEPHANT
YELLOW	4	TIGER

WE SPLIT THE DATA



How does a Decision Tree work?



THIS TREE CAN NOW PREDICT ALL THE CLASSES OF ANIMALS PRESENT IN THE DATASET WITH 100% ACCURACY

Example:
To go for Outing or Not

Day	Outlook	Temp.	Humidity	Wind	Decision
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

Calculate Entropy

Decision column consists of 14 instances and includes two labels: yes and no.

There are 9 decisions labelled YES, and 5 decisions labelled NO.

- Entropy(Decision) = – p(Yes) * $\log_2$p(Yes) – p(No) * $\log_2$p(No)
- Entropy(Decision) = – (9/14) * $\log_2$(9/14) – (5/14) * $\log_2$(5/14) = 0.940

Strong wind factor

Day	Outlook	Temp.	Humidity	Wind	Decision
2	Sunny	Hot	High	Strong	No
6	Rain	Cool	Normal	Strong	No
7	Overcast	Cool	Normal	Strong	Yes
11	Sunny	Mild	Normal	Strong	Yes
12	Overcast	Mild	High	Strong	Yes
14	Rain	Mild	High	Strong	No

$$\text{Entropy}(D|W=\text{Strong}) = -p(\text{No}) * \log_2 p(\text{No}) - p(\text{Yes}) * \log_2 p(\text{Yes})$$

$$\text{Entropy}(D|W=\text{Strong}) = -(3/6) * \log_2(3/6) - (3/6) * \log_2(3/6) = 1$$

Weak wind factor on decision

$$\text{Entrophy}(D|W=\text{Weak}) = -\,p(\text{No}) * \log_2 p(\text{No}) - p(\text{Yes}) * \log_2 p(\text{Yes})$$

$$\text{Entrophy}(D|W=\text{Weak}) = -\,(2/8) * \log_2(2/8) - (6/8) * \log_2(6/8) = 0.811$$

Day	Outlook	Temp.	Humidity	Wind	Decision
1	Sunny	Hot	High	Weak	No
3	Overcast	Hot	High	Weak	Yes
4	Rain	Mild	High	Weak	Yes
5	Rain	Cool	Normal	Weak	Yes
8	Sunny	Mild	High	Weak	No
9	Sunny	Cool	Normal	Weak	Yes
10	Rain	Mild	Normal	Weak	Yes
13	Overcast	Hot	Normal	Weak	Yes

Gain(D, W) = Entropy(D) – [p(D|W=Weak) * Entropy(D|W=Weak)] – [p(D|W=Strong) * Entropy(D|W=Strong)] = 0.940 – [(8/14) * 0.811] – [(6/14)* 1] = 0.048

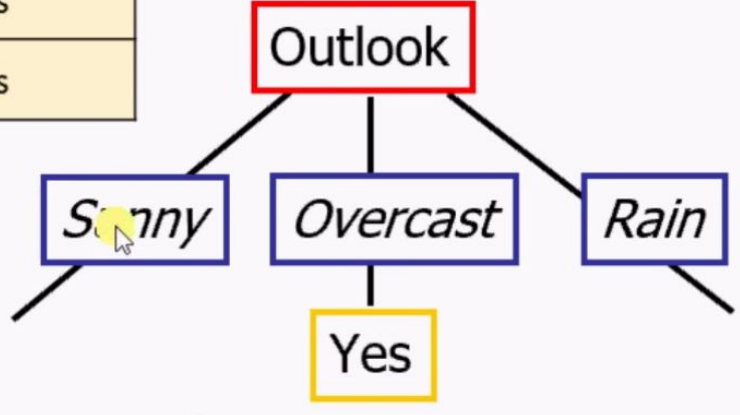
Other factors on decision

Applied similar calculation on the other columns.

- Gain(Decision, Outlook) = 0.246
- Gain(Decision, Temperature) = 0.029
- Gain(Decision, Humidity) = 0.151

Overcast outlook on decision

Day	Outlook	Temp.	Humidity	Wind	Decision
3	Overcast	Hot	High	Weak	Yes
7	Overcast	Cool	Normal	Strong	Yes
12	Overcast	Mild	High	Strong	Yes
13	Overcast	Hot	Normal	Weak	Yes

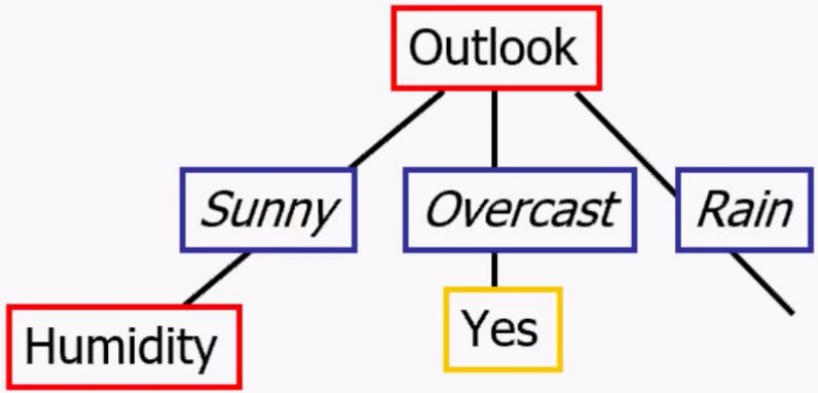


Decision will always be yes if outlook were overcast.

Sunny outlook on decision

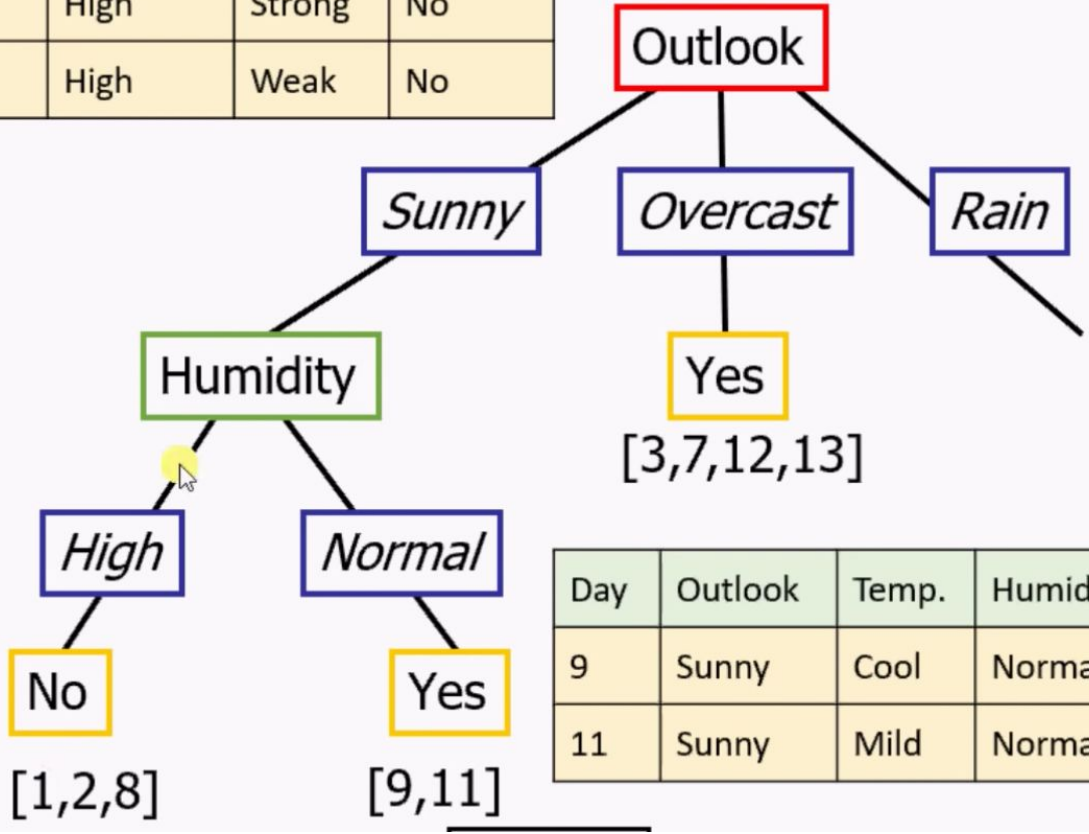
- (Outlook=Sunny|Temperature)gain = 0.570
- (Outlook=Sunny|Humidity)gain = 0.970
- (Outlook=Sunny|Wind)gain = 0.019

Humidity is the decision



Day	Outlook	Temp.	Humidity	Wind	Decision
1	Sunny	Hot	High	Weak	No
2	Sunny	Hot	High	Strong	No
8	Sunny	Mild	High	Weak	No
9	Sunny	Cool	Normal	Weak	Yes
11	Sunny	Mild	Normal	Strong	Yes

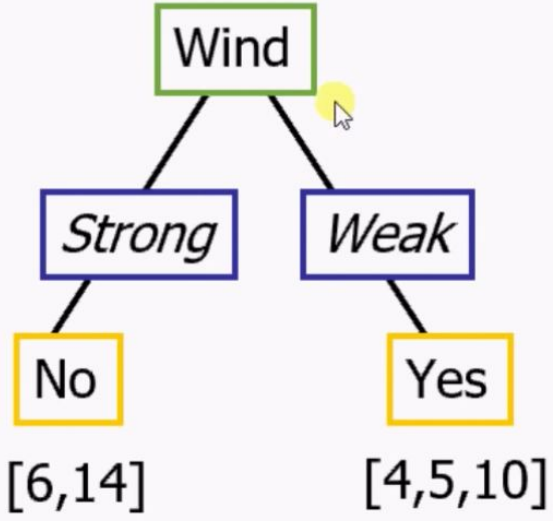
Day	Outlook	Temp.	Humidity	Wind	Decision
1	Sunny	Hot	High	Weak	No
2	Sunny	Hot	High	Strong	No
8	Sunny	Mild	High	Weak	No



Day	Outlook	Temp.	Humidity	Wind	Decision
9	Sunny	Cool	Normal	Weak	Yes
11	Sunny	Mild	Normal	Strong	Yes

Rain outlook on decision

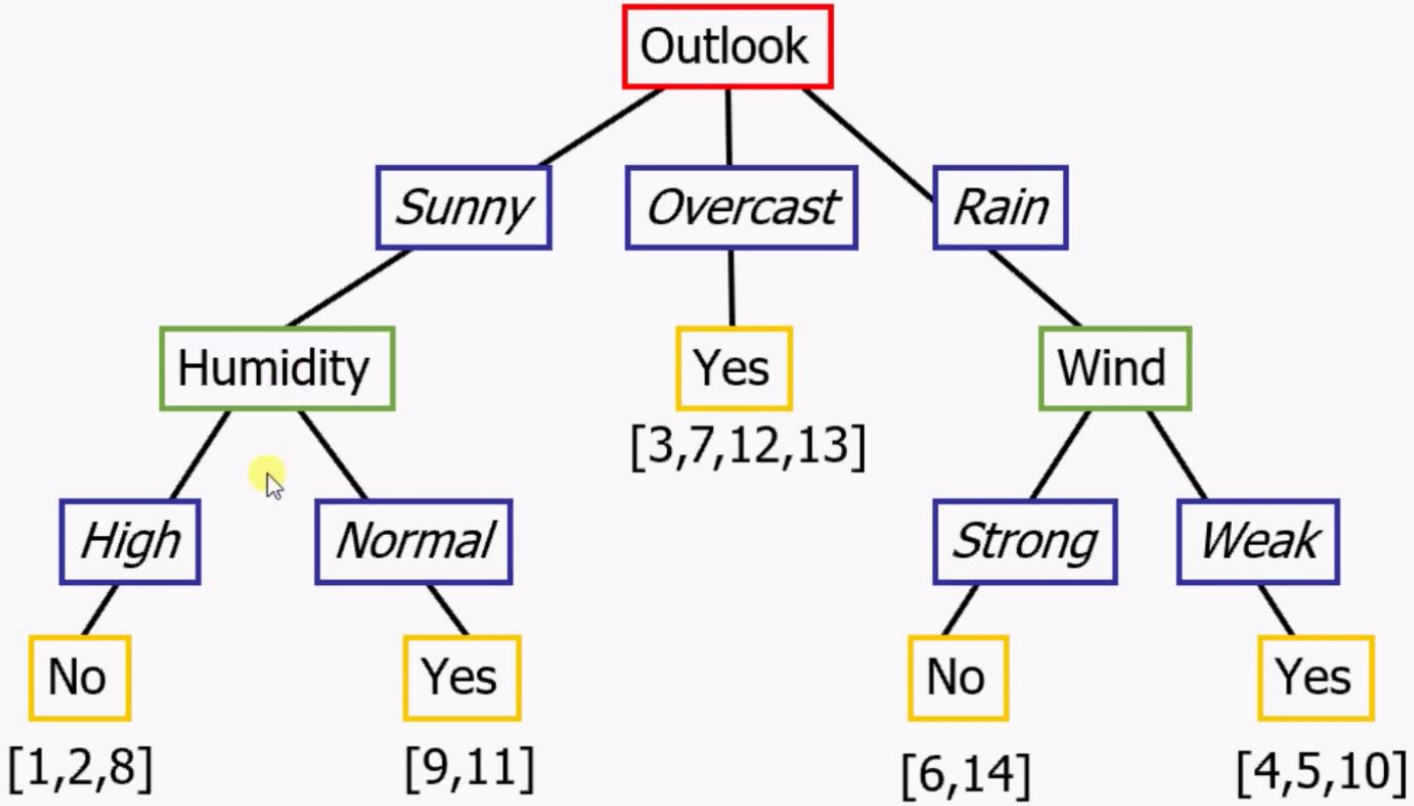
Day	Outlook	Temp.	Humidity	Wind	Decision
4	Rain	Mild	High	Weak	Yes
5	Rain	Cool	Normal	Weak	Yes
6	Rain	Cool	Normal	Strong	No
10	Rain	Mild	Normal	Weak	Yes
14	Rain	Mild	High	Strong	No



- Gain(Outlook=Rain | Temperature)
- Gain(Outlook=Rain | Humidity)
- Gain(Outlook=Rain | Wind)

Wind produces the highest score if outlook were rain

Complete Decision Tree



Difference between ID3 and CART

	Splitting Criteria	Attribute type	Missing values	Pruning Strategy	Outlier Detection
ID3	Information Gain	Handles only Categorical value	Do not handle missing values.	No pruning is done	Susceptible to outliers
CART	Towing Criteria	Handles both Categorical & Numeric value	Handle missing values.	Cost-Complexity pruning is used	Can handle Outliers
C4.5	Gain Ratio	Handles both Categorical & Numeric value	Handle missing values.	Error Based pruning is used	Susceptible to outliers