Classification

Decision Trees

Decision Trees

- What are decision trees and where are they used ?
- Structure of a decision tree
- Building a decision tree
- Entropy and Information Gain

DECISION TREES: WHAT-WHERE-WHY

Decision Tree:

Supervised learning predictive model that uses a set of binary rules to calculate a target value. It is used for either **classification** (**categorical target variable**) or **regression** (**continuous target variable**).

Types of Decision Trees:

- ✓ **Classification Trees**
- ✓ **Regression Trees**

Why Use Decision Trees:

- Easy to build
- Easy to interpret
- Less data cleaning is needed
- Regression trees work with non-linear data

In this session, we will explore Classification Trees.

CLASSIFICATION TREES

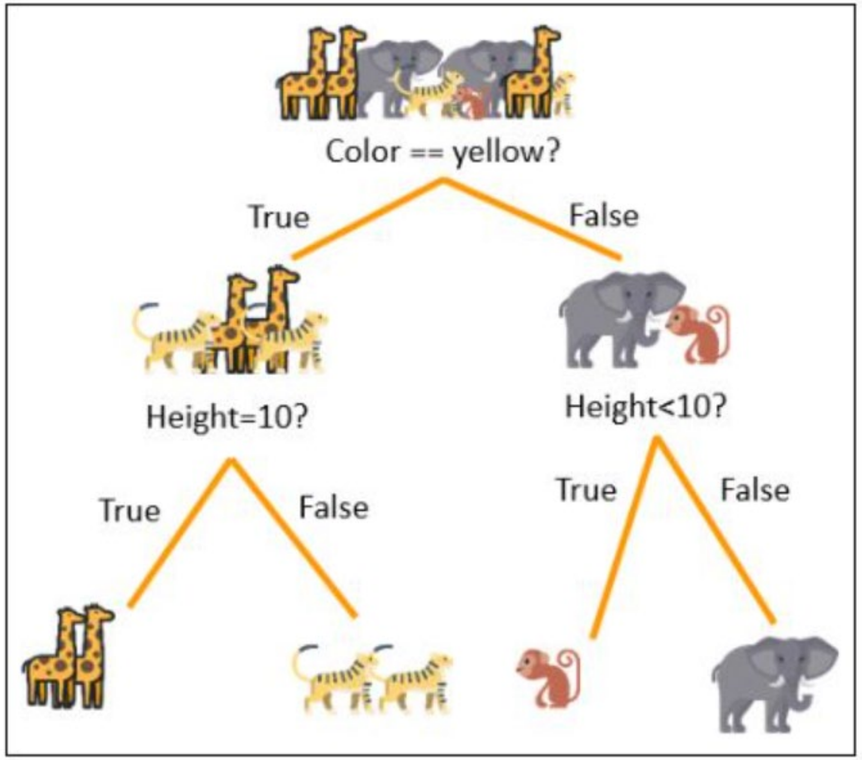


Image source:
https://www.simplilearn.com/tutorials/machine-learning-tutorial/decision-tree-in-python

Use cases for classification trees:

- Credit scoring
- Medical diagnosis
- Customer behaviour models

STRUCTURE OF A DECISION TREE

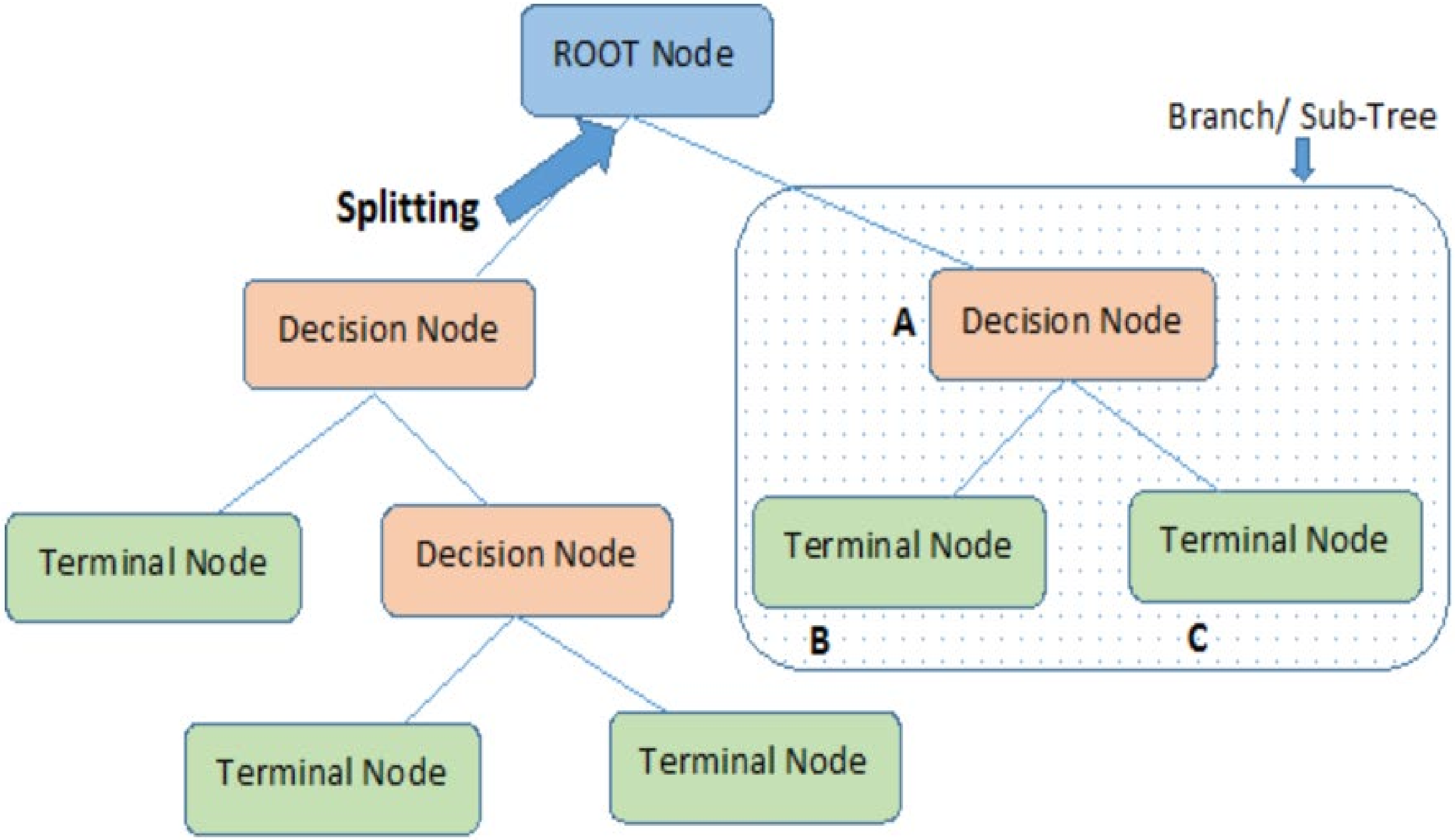


Image source: https://medium.com/analytics-vidhya/a-guide-to-machine-learning-in-r-for-beginners-decision-trees-c24dfd490abb

BUILDING A DECISION TREE
(ID3 ALGORITHM)

To build a decision tree (ID3 algorithm):

- Select the best predictor (attribute)
- Split the data in branches according to the values of the attribute
- For each branch, check if it is sufficiently pure

    → if yes – stop

    → if no – select the best predictor (attribute) and branch further

CLASSIFICATION TREE – SIMPLE EXAMPLE

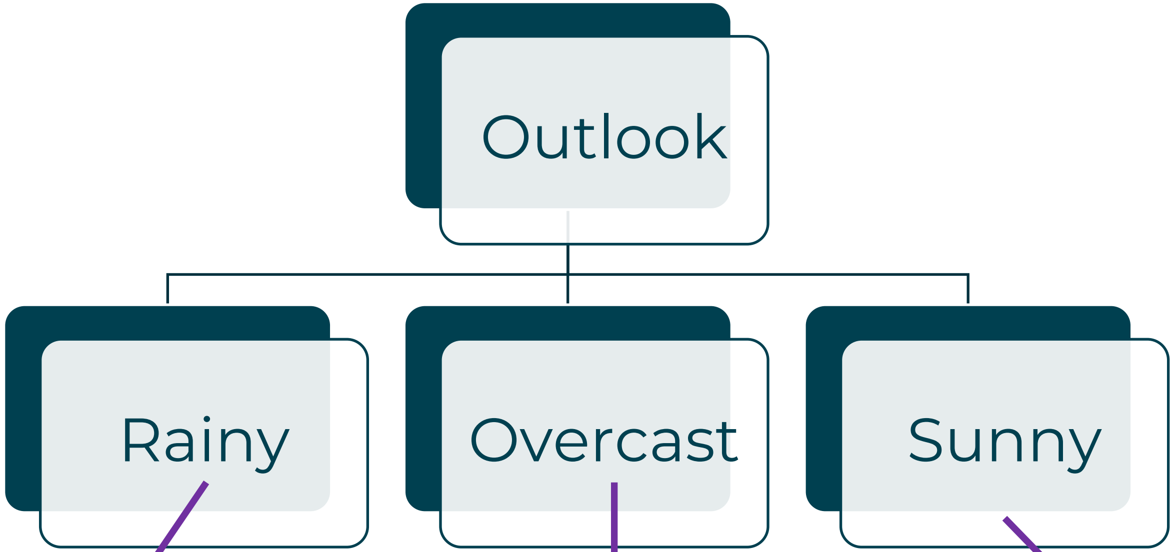
Will it be possible to play golf tomorrow ?

The data from the past:

Play	Outlook	Temperature	Humidity	Wind
yes	rainy	cool	normal	No
no	rainy	cool	normal	Yes
yes	overcast	hot	high	No
no	sunny	mild	high	No
yes	rainy	cool	normal	No
yes	sunny	cool	normal	No
yes	rainy	cool	normal	No
yes	sunny	hot	normal	No
yes	overcast	mild	high	Yes
no	sunny	mild	high	Yes

→ **Select Outlook as the first predictor (attribute) to split**

CLASSIFICATION TREE – SIMPLE EXAMPLE



Play	Outlook	Temperature	Humidity	Wind
yes	rainy	cool	normal	No
no	rainy	cool	normal	Yes
yes	rainy	cool	normal	No
yes	rainy	cool	normal	No

Impure node – split further

Play	Outlook	Temperature	Humidity	Wind
yes	overcast	hot	high	No
yes	overcast	mild	high	Yes

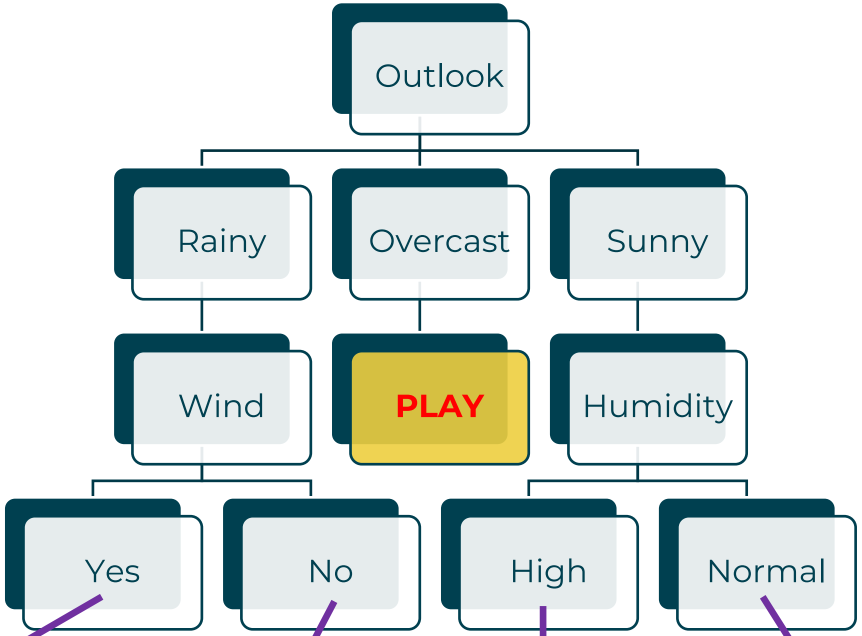
Pure node – they will play

Play	Outlook	Temperature	Humidity	Wind
no	sunny	mild	high	No
yes	sunny	cool	normal	No
yes	sunny	hot	normal	No
no	sunny	mild	high	Yes

Impure node – split further

A node is pure when all of its data belongs to a single class
(here – the class Play-yes)

CLASSIFICATION TREE – SIMPLE EXAMPLE



Play	Outlook	Temperature	Humidity	Wind
no	rainy	cool	normal	Yes

Pure node – they will not play

Play	Outlook	Temperature	Humidity	Wind
yes	rainy	cool	normal	No
yes	rainy	cool	normal	No
yes	rainy	cool	normal	No

Pure node – they will play

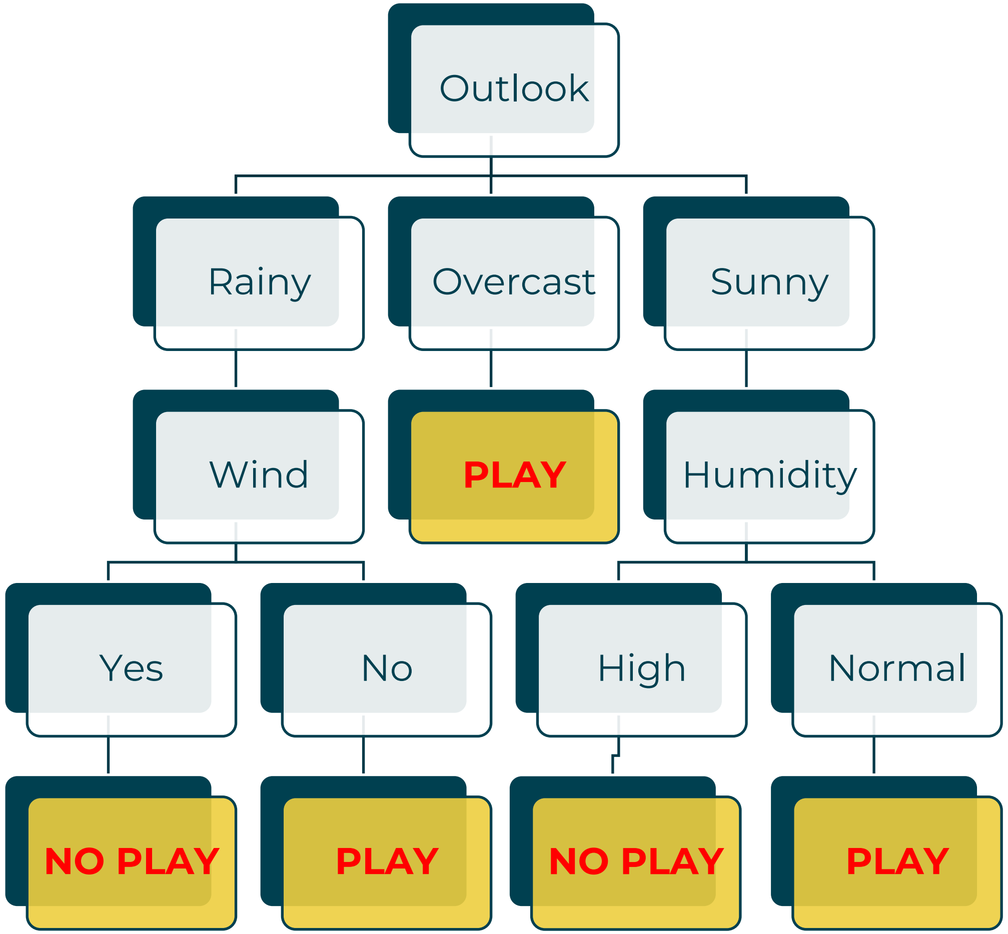
Play	Outlook	Temperature	Humidity	Wind
no	sunny	mild	high	No
no	sunny	mild	high	Yes

Pure node – they will not play

Play	Outlook	Temperature	Humidity	Wind
yes	sunny	cool	normal	No
yes	sunny	hot	normal	No

Pure node – they will play

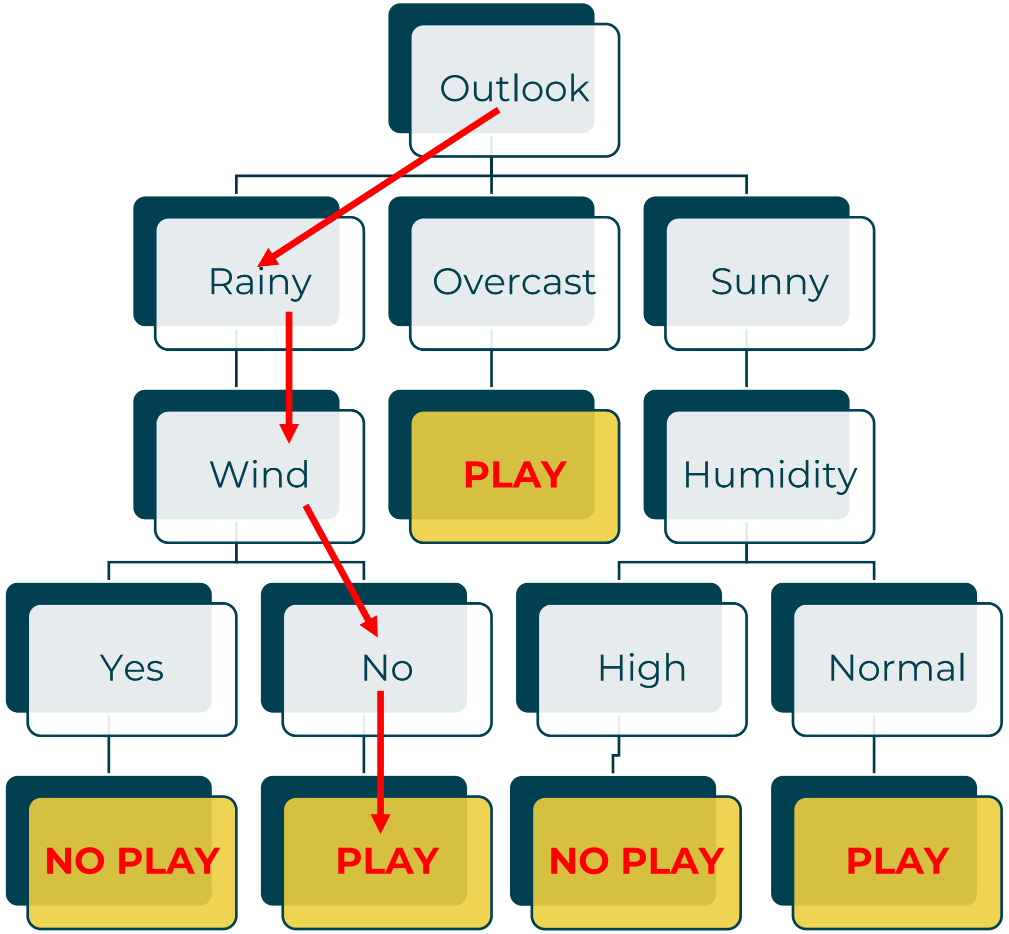
CLASSIFICATION TREE – SIMPLE EXAMPLE - SOLUTION



CLASSIFICATION TREE – SIMPLE EXAMPLE - PREDICTION

Weather forecast:

Outlook: rainy, **Temperature:** cool, **Humidity:** high, **Wind:** No



YES – there will be play !

ENTROPY

The big question:

How do we decide on what attribute to split ?

There are several criteria used to decide on what attribute to split. **Entropy** is probably the most popular one.

Entropy is a measure of the **impurity** or **uncertainty** in the data. The mathematical formula for entropy is:

$$H = \sum_{i=1}^{n} -p_i \log_2 p_i$$

Where p_i is the probability of an element of the data.

Let's assume that the data can belong to one of two classes – "+" or "-".

→ *If all elements are of the same class the entropy is 0 (there is no uncertainty)*

→ *If the elements are evenly split between the two classes the entropy is 1 (the uncertainty is the highest)*

ENTROPY (ILLUSTRATION)

Illustration: entropy depending on the composition of the data (classes "+" and "-").

$$H = \sum_{i=1}^{n} -p_i \log_2 p_i$$

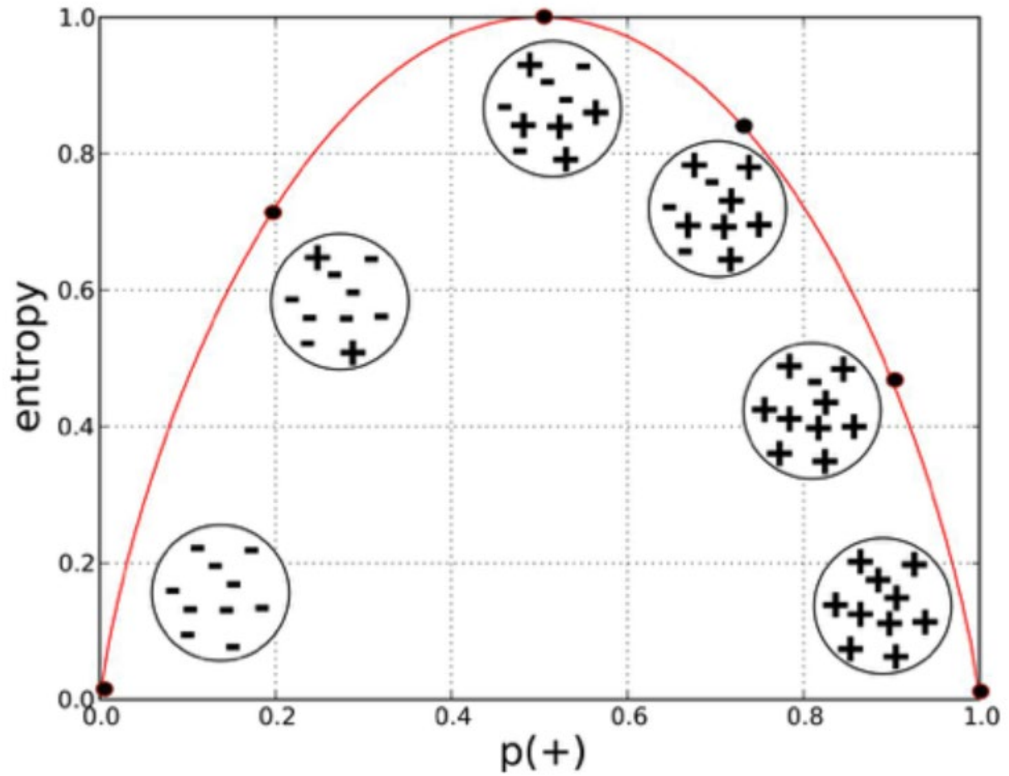


Image source: https://towardsdatascience.com/entropy-how-decision-trees-make-decisions-2946b9c18c8

INFORMATION GAIN

Every split results in a decrease of the entropy (the resulting nodes become purer). This decrease of entropy is called **Information Gain** (IG).

IG = Entropy(before split) **- Entropy**(after split)

The entropy after the split depends on the predictor (attribute) that we split on.

The best predictor (attribute) is the one that leads to the **highest information gain (lowest entropy** after the split**).**