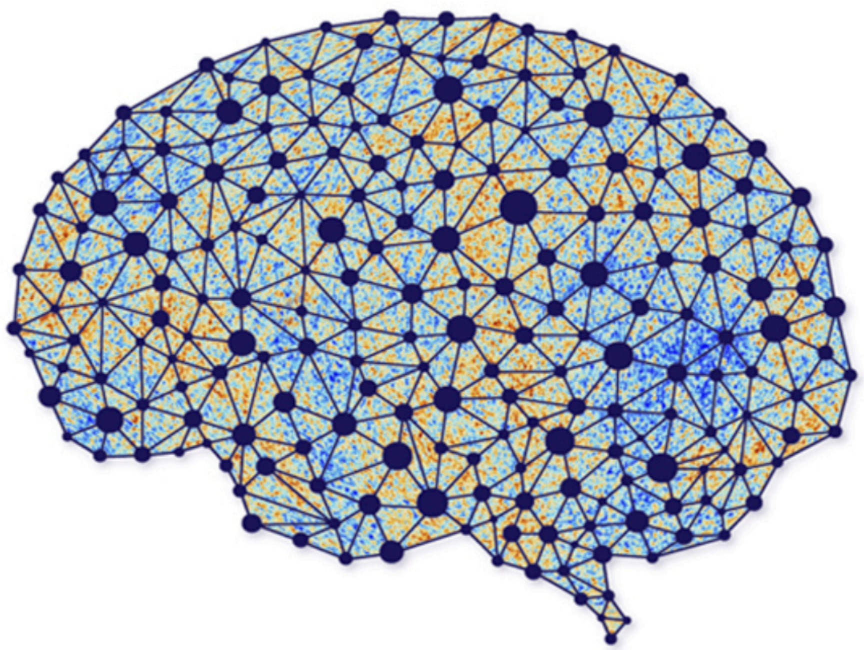
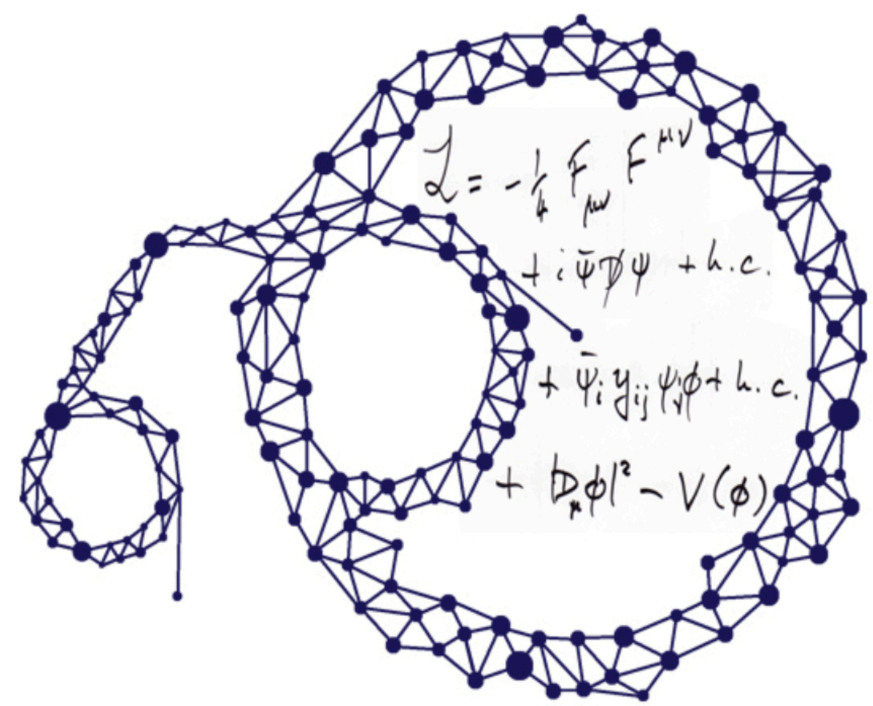
PHY 835: Machine Learning in Physics

Lecture 15: Decision Trees & kNN

March 12, 2024



AI
∩
Universe



Gary Shiu

Outline for today

- Many ML models are designed to solve **classification** or **regression** problems.

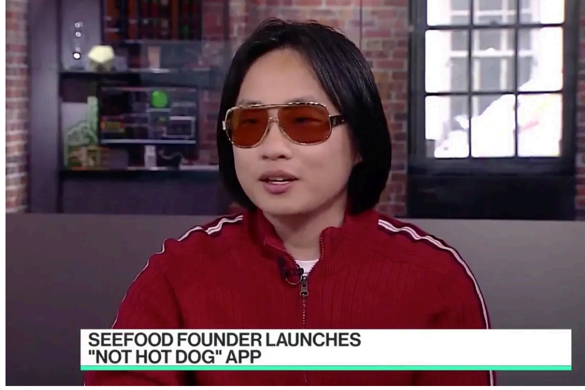
- Simple Classifiers:

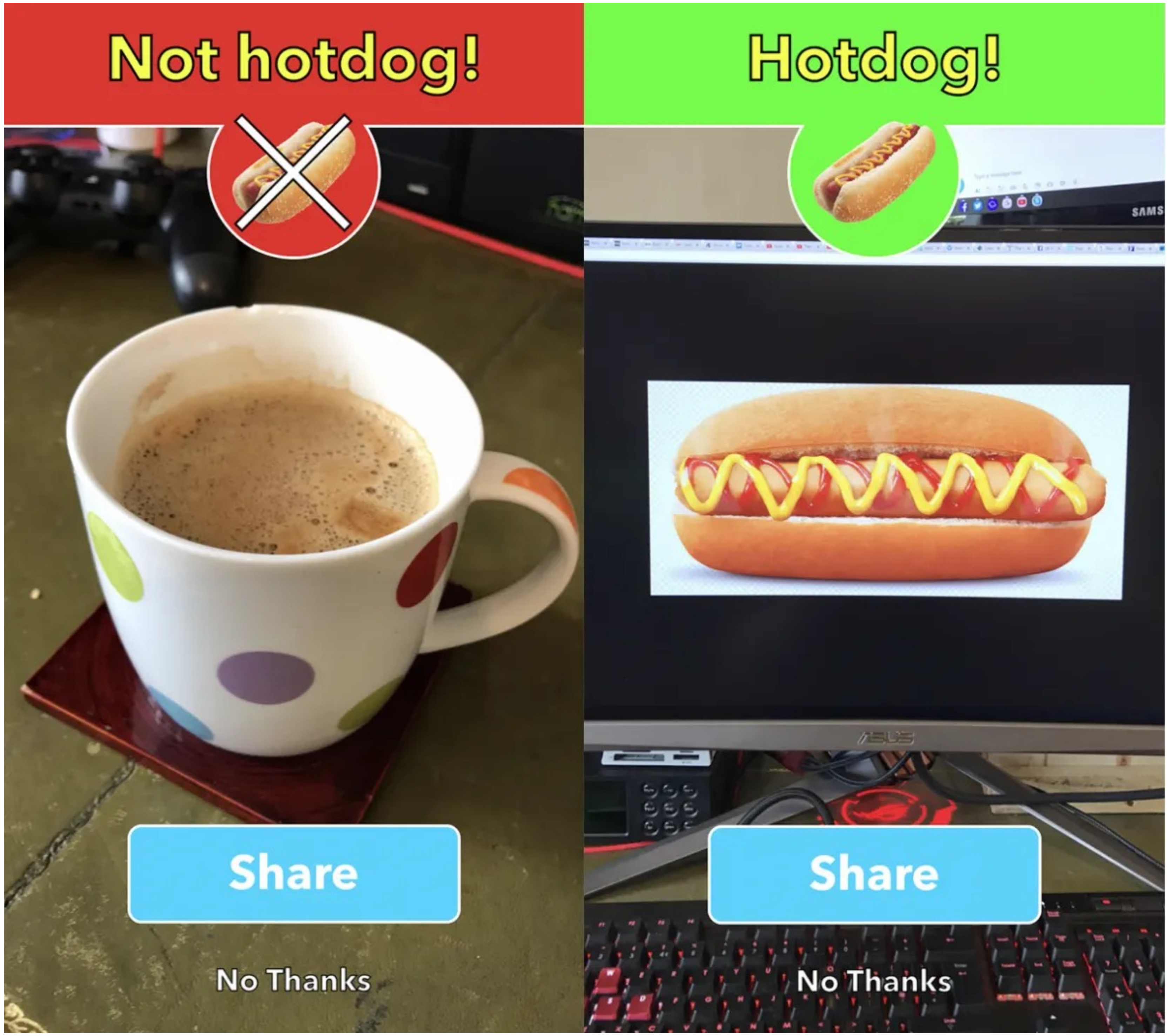
  - Decision Trees

  - kNN: Finding neighbors

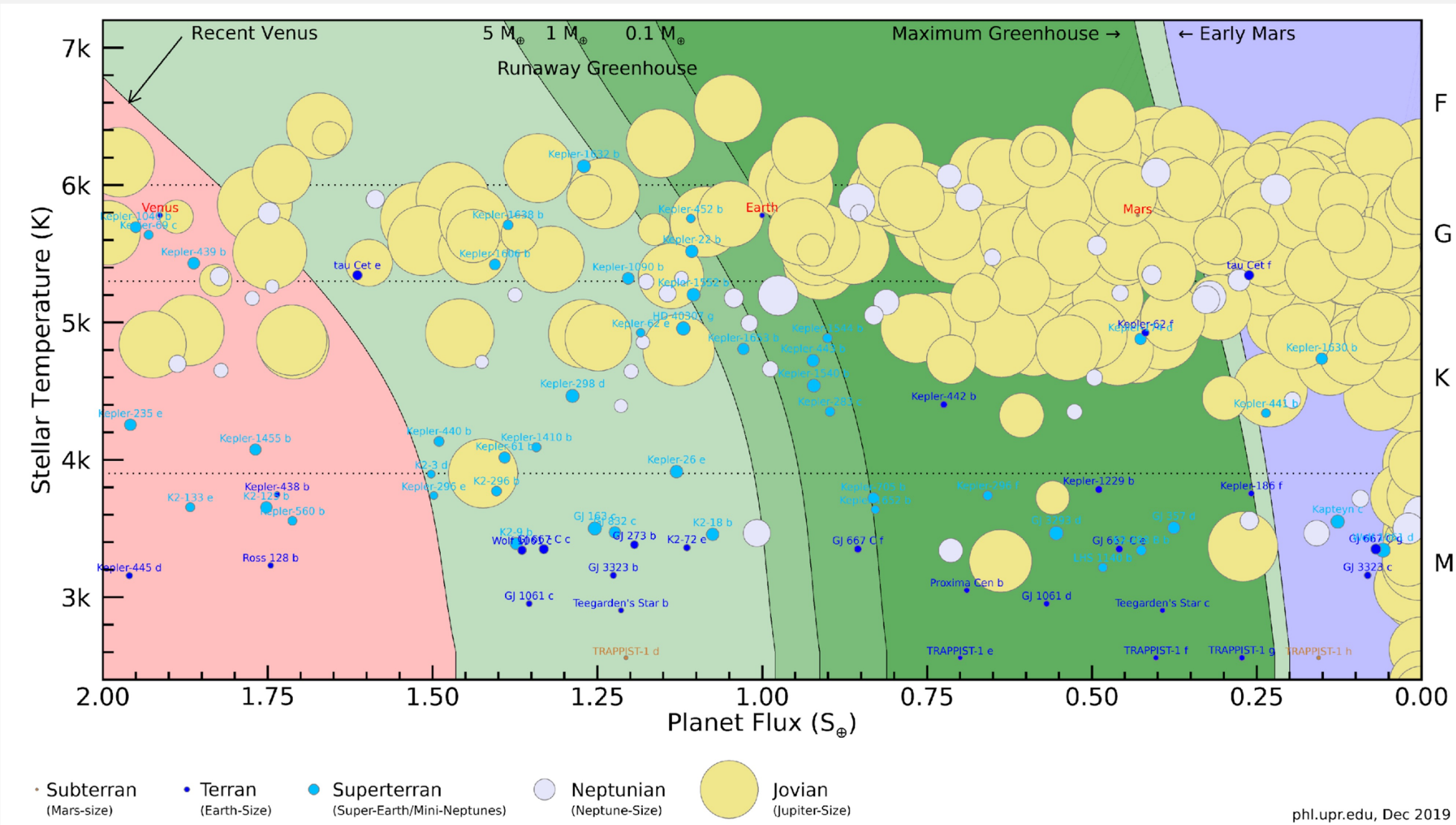
- Reference: “Machine Learning for Physics and Astronomy” by Viviana Acquaviva, Princeton University Press, Chapter 2.

Hotdog Classifier





DATA FROM THE PLANET HABITABILITY LAB AT ARECIBO OBSERVATORY



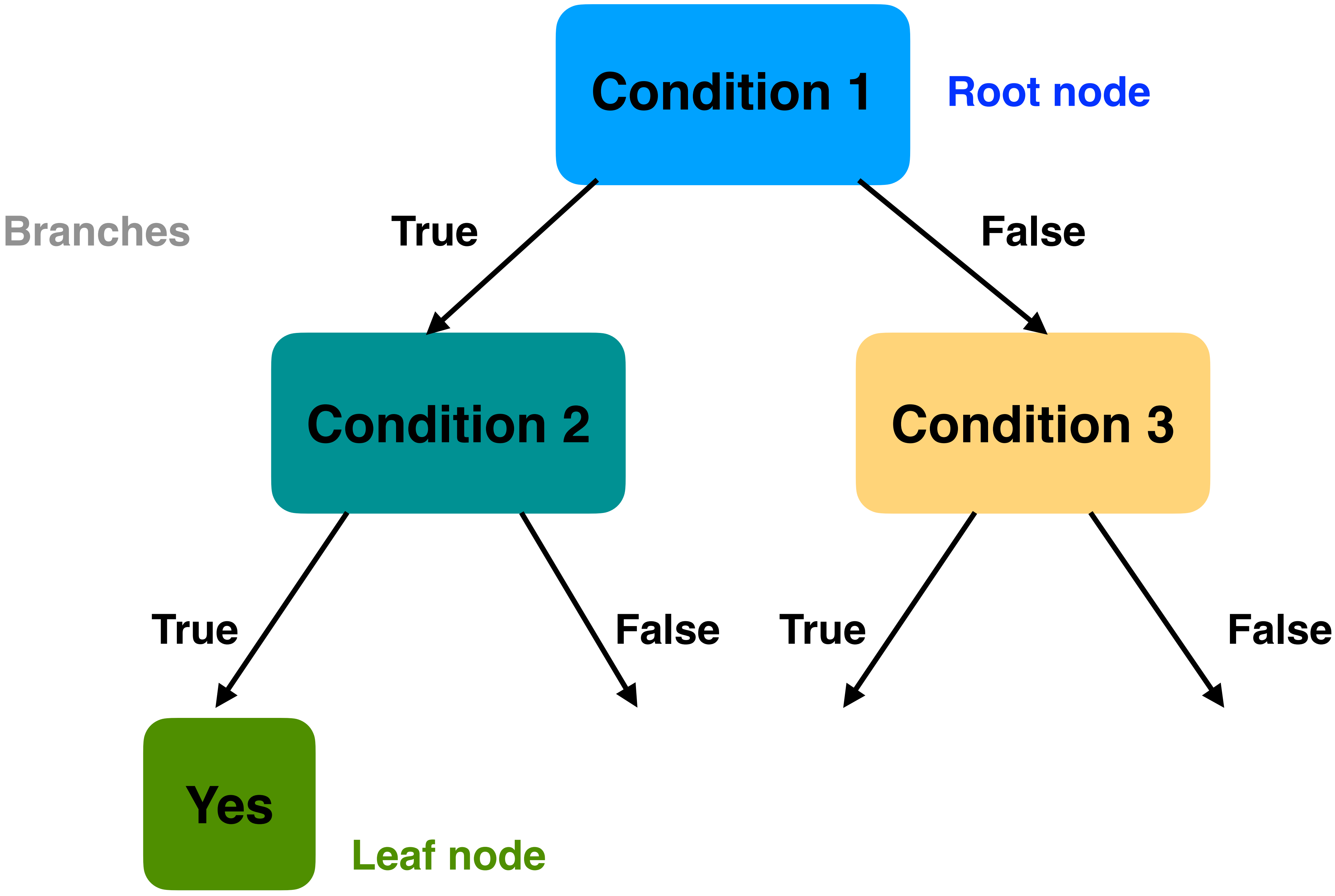
NAME	Stellar Mass ($M_\odot$)	Orbital Period (days)	Distance (AU)	Habitable?
Kepler-736 b	0.86	3.60	0.0437	0
Kepler-636 b	0.85	16.08	0.1180	0
Kepler-887 c	1.19	7.64	0.0804	0
Kepler-442 b	0.61	112.30	0.4093	1
Kepler-772 b	0.98	12.99	0.1074	0
Teegarden's Star b	0.09	4.91	0.0252	1
K2-116 b	0.69	4.66	0.0481	0
GJ 1061 c	0.12	6.69	0.035	1
HD 68402 b	1.12	1103	2.1810	0
Kepler-1544 b	0.81	168.81	0.5571	1
Kepler-296 e	0.5	34.14	0.1782	1
Kepler-705 b	0.53	56.06	0.2319	1
Kepler-445 c	0.18	4.87	0.0317	0
HD 104067 b	0.62	55.81	0.26	
GJ 4276 b	0.41	13.35	0.0876	
Kepler-296 f	0.5	63.34	0.2689	
Kepler-63 b	0.98	9.43	0.0881	
GJ 3293 d	0.42	48.13	0.1953	

Table 2.1: Learning set for the habitable planets problem.

DECISION TREES

- Work by splitting data on different values of features
- Simplest trees are binary trees
- If categorical features, the split would be on yes/no
- If numerical, the split would be on a certain value (e.g. x > 100 or x < 100)

Decision Trees



Decision Trees

- Depth of tree = maximum number of splitting conditions.

- Stop growing the tree when 1) all items on a branch have the same features (values) or 2) other stopping criterion is met.

- Usually have maximum criterion to avoid overfitting.

- At each splitting node, look for features which provide the best splitting condition. How do we quantify best?

- **Maximize "information gain"** or **maximize decrease in impurity**. (defined more precisely later)

EXAMPLE: THIS 2-FEATURE DATA SET.

HOW SHOULD WE SPLIT?

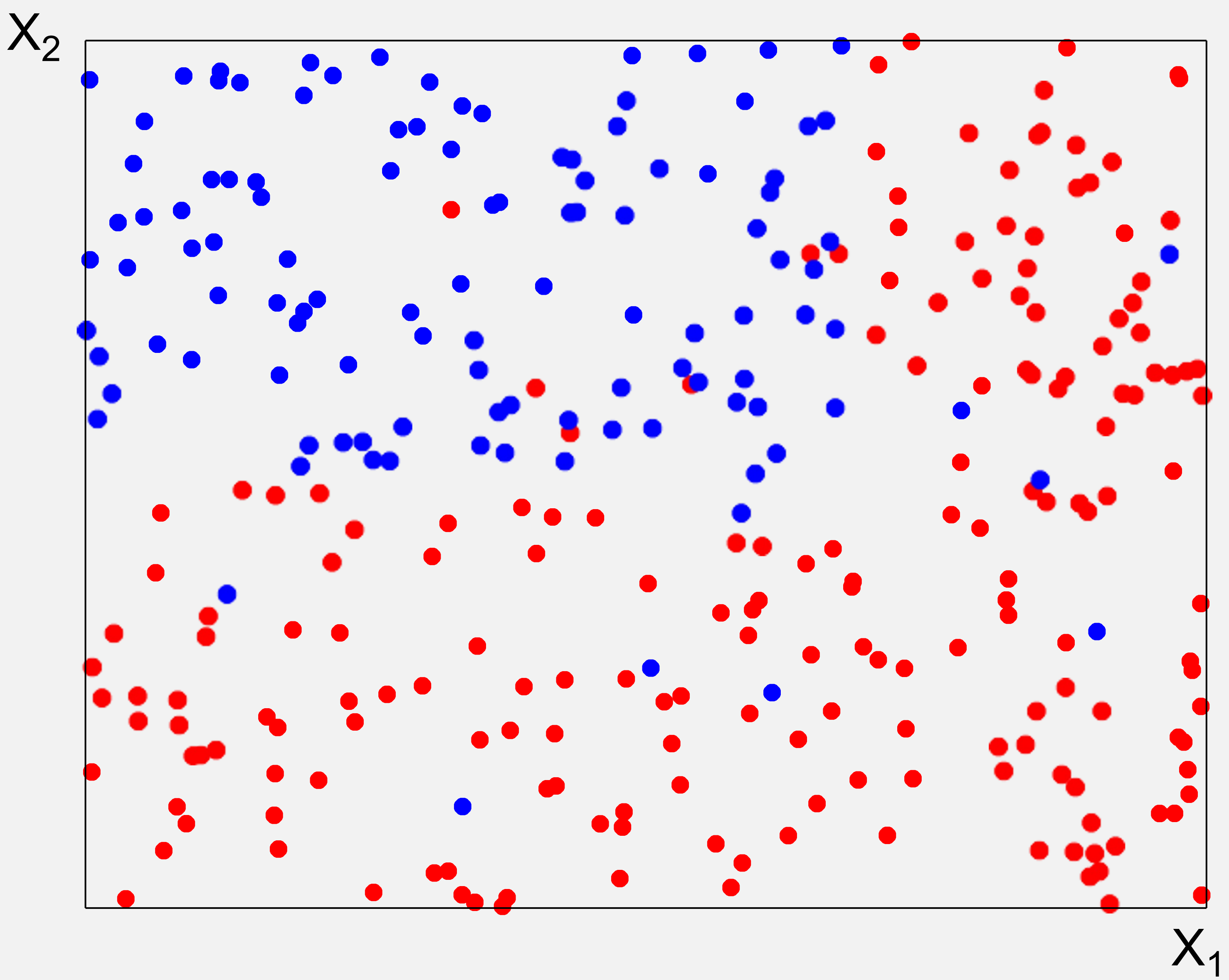


Figure credit:
Gilles Louppe

EXAMPLE: THIS 2-FEATURE DATA SET.

HOW SHOULD WE SPLIT?

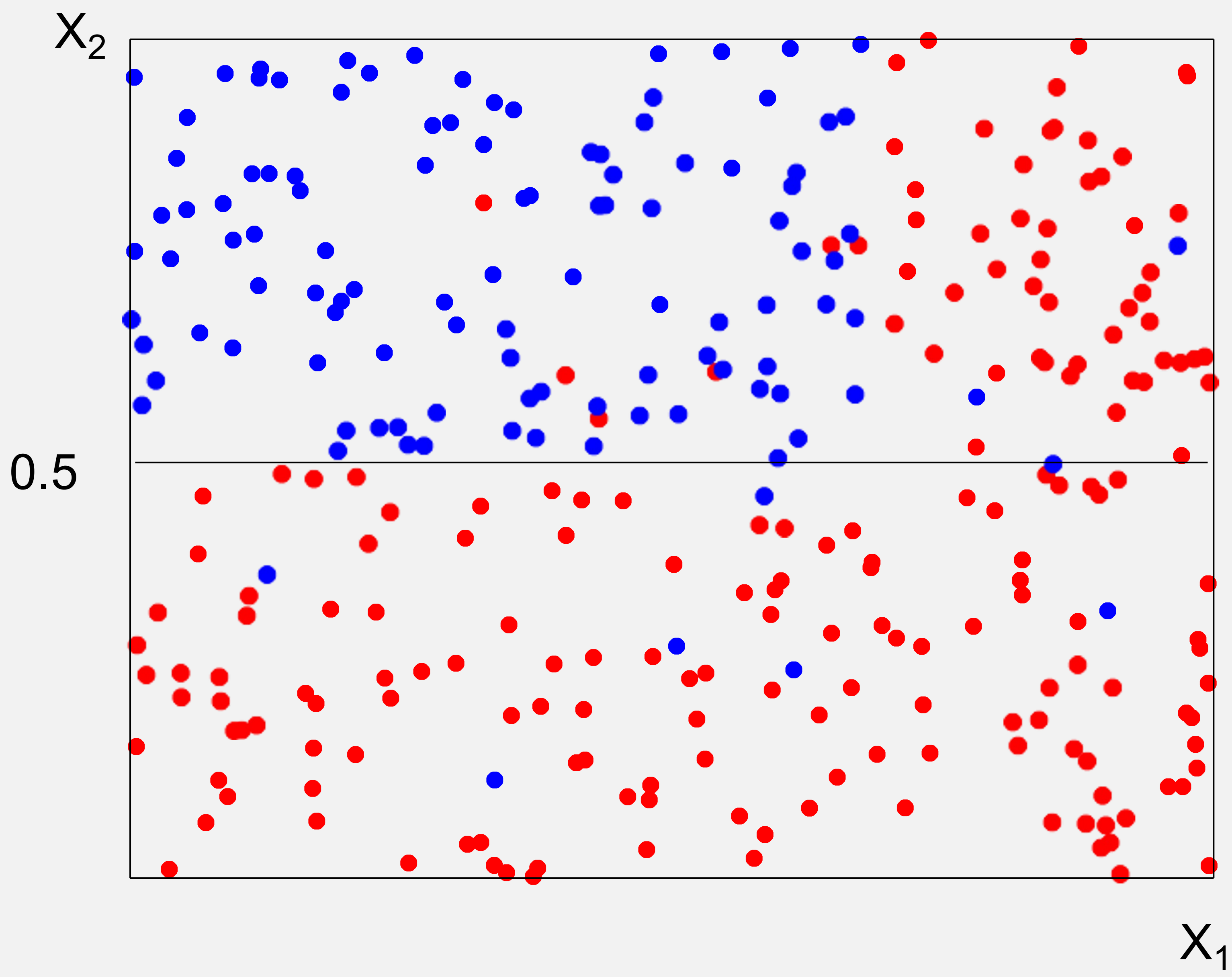


Figure credit:
Gilles Louppe

SHOULD WE STOP?

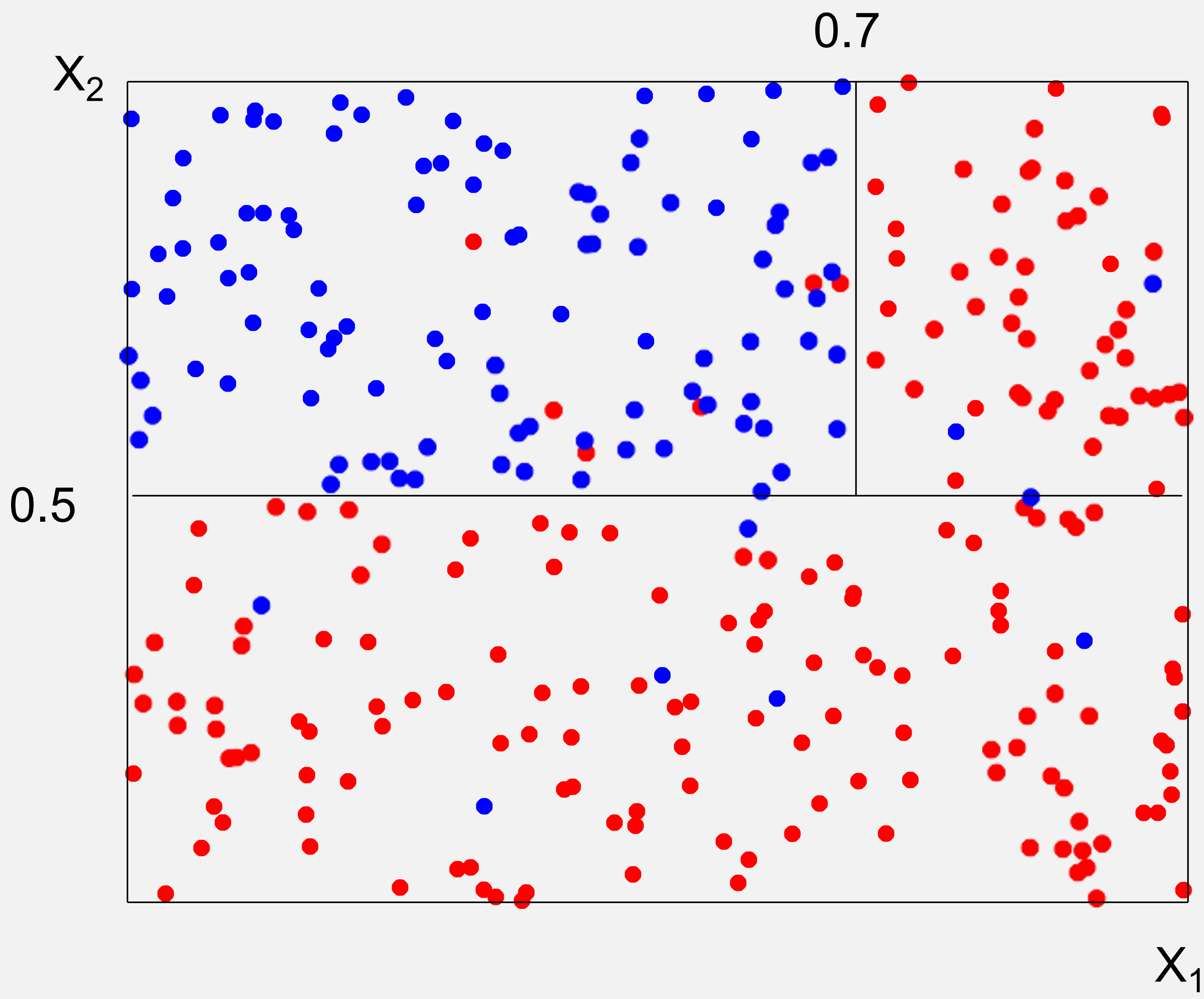
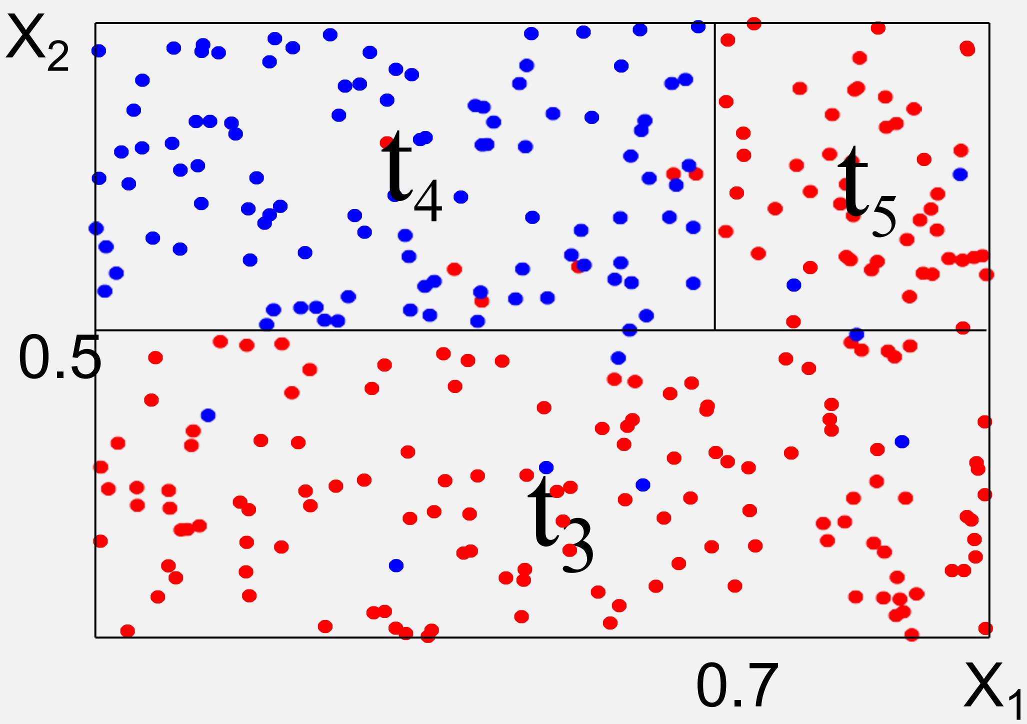
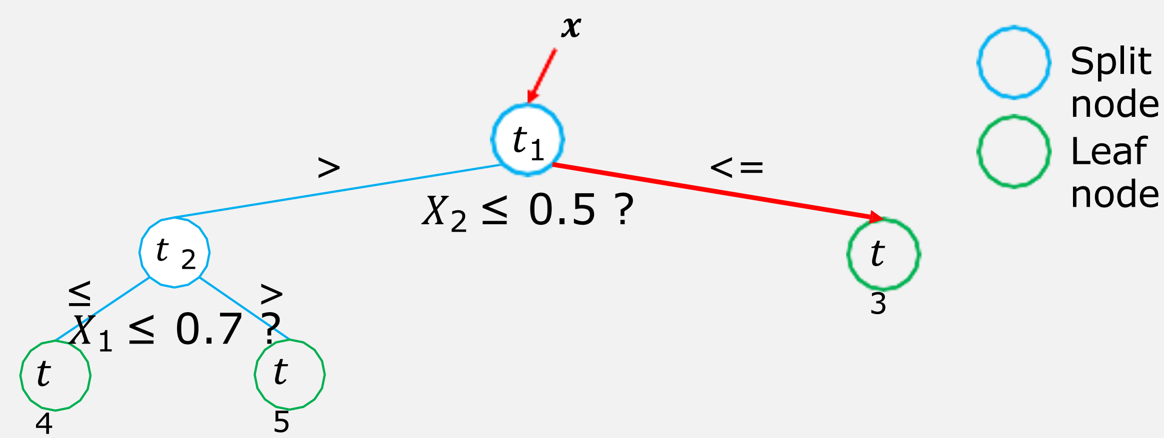


Figure credit:
Gilles Louppe

NODES (SPLITS AND LEAVES) DEFINE THE DECISION TREE





In a terminal node (leaf),
the model is ready to output a classification
(and all objects in that leaf have the same class)

Figure credit:
Gilles Louppe

Important questions:

How do we decide which splits to make, among the many possible ones?

How do we decide whether we should stop?

BUILDING DECISION TREES

Find measure of impurity (e.g. Gini impurity) that we want to minimize.

Find splits that maximize decrease of impurity

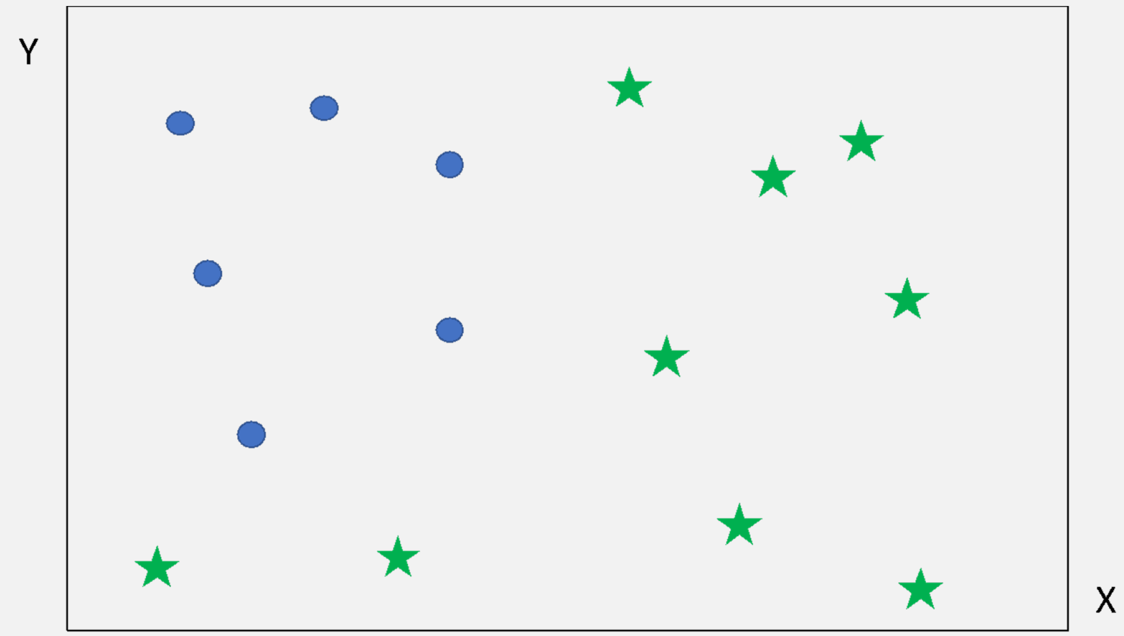
Select stopping criterion as impurity < ε (e.g. , 0)

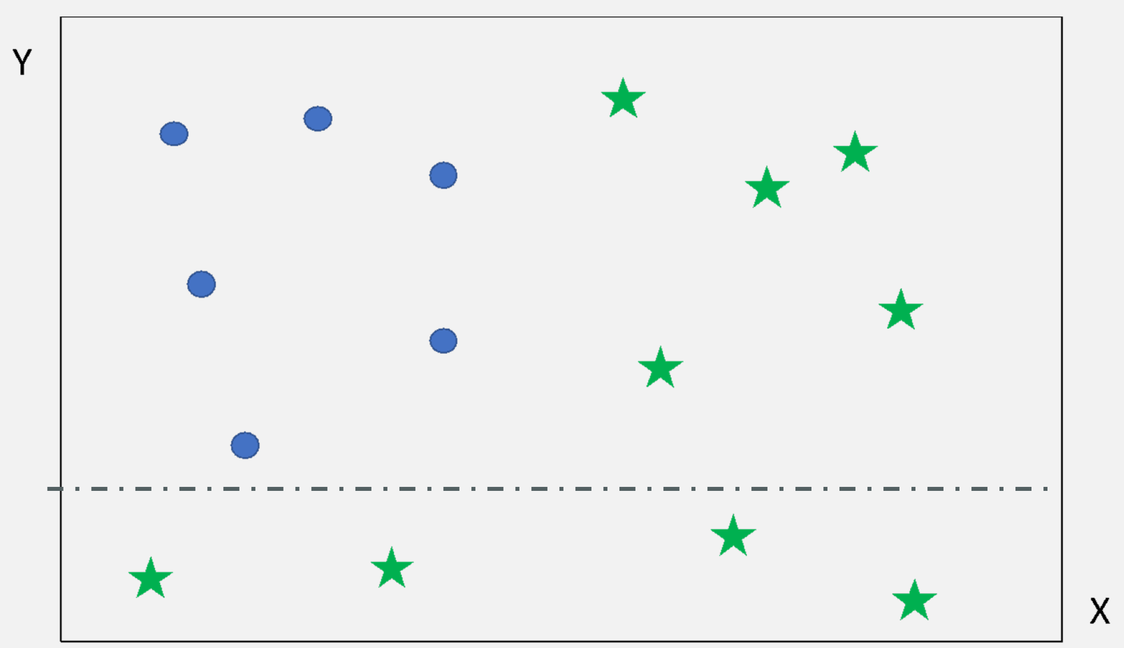
$$\text{Gini (node L)} = 1 - \sum \mathbf{f(i)}^2$$

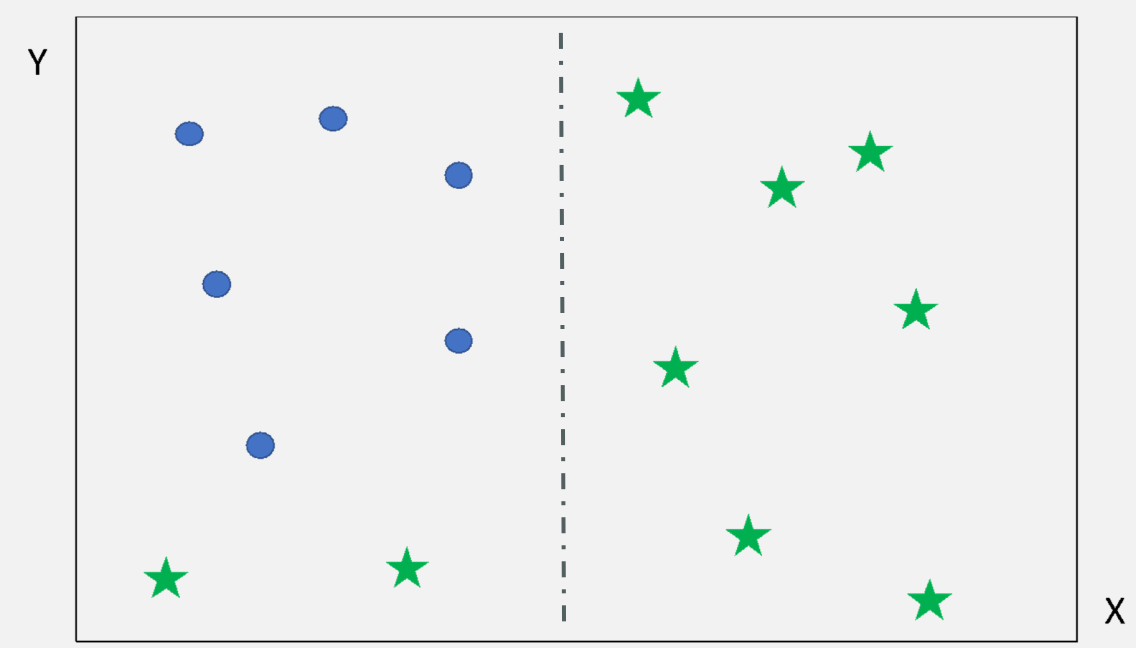
where f(i) is the frequency (=fractional abundance) of the i-th class

$$L_L/L * (1 - \sum f(i)^2)_L + L_R/L * (1 - \sum f(i)^2)_R$$

L = total # of objects in original split; L_L and L_R = # of objects in each of the new splits

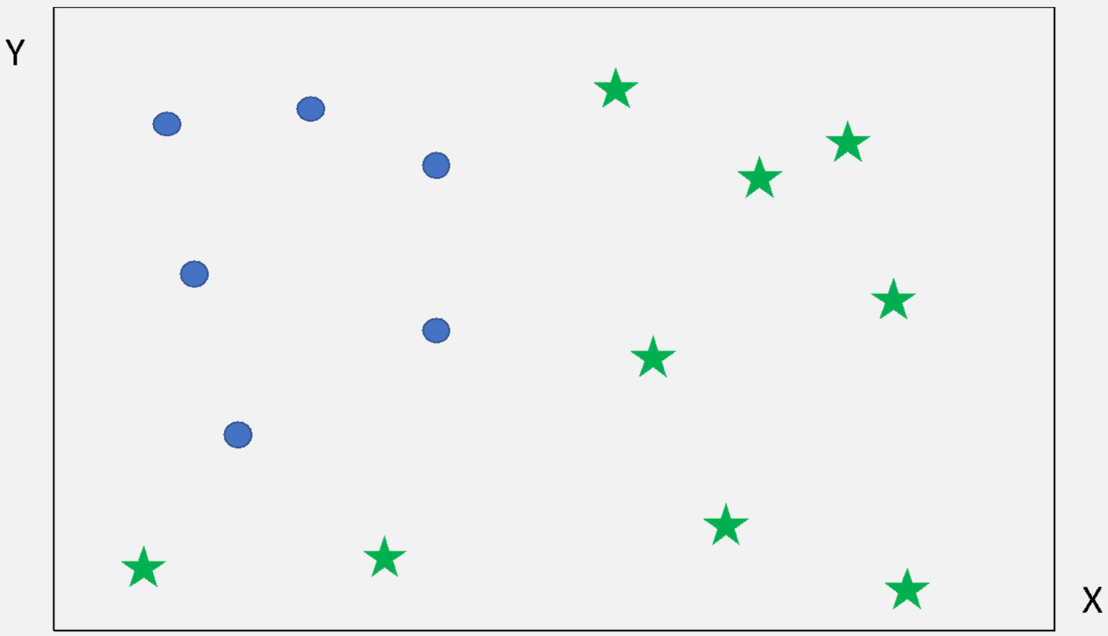




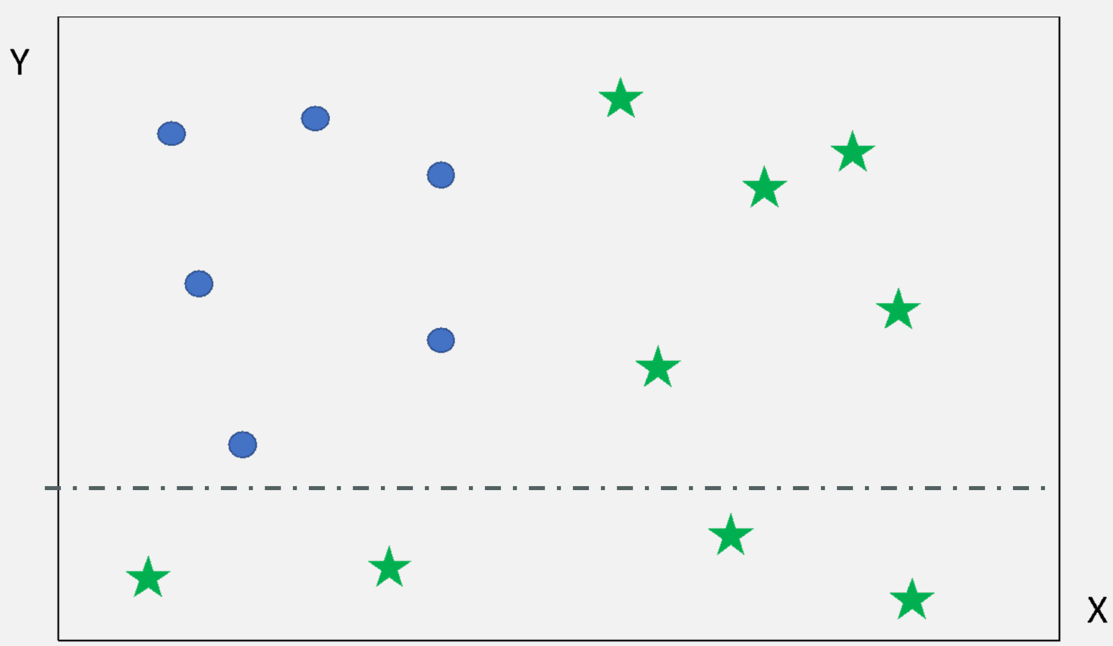


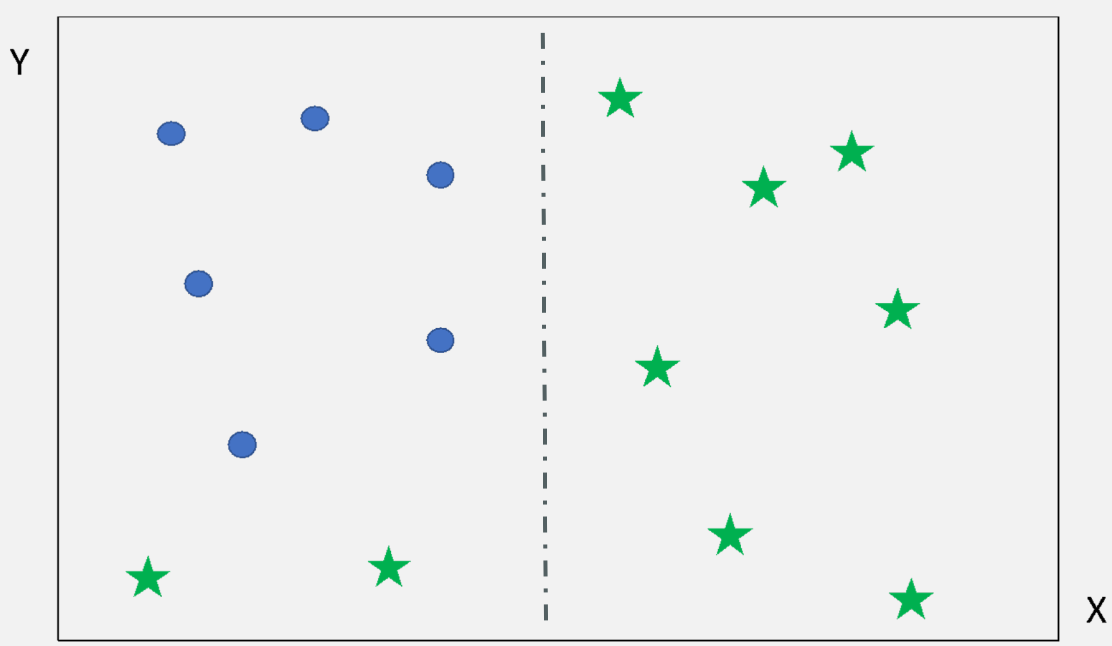
Which split should we do first?
Let's calculate the Gini impurity in the original and each of the two.

$$L_L/L * (1 - \sum f(i)^2)_L + L_R/L * (1 - \sum f(i)^2)_R$$



$$(1 - \sum f(i)^2) =$$
$$1 - (6/15)^2 - (9/15)^2$$
$$= 0.48$$





$$L_L/L * (1 - \sum f(i)^2)_L + L_R/L * (1 - \sum f(i)^2)_R$$
$$= 4/15 * 0 + 11/15 * (1 - (6/11)^2 - (5/11)^2) = 0.363$$

$$L_L/L * (1 - \sum f(i)^2)_L + L_R/L * (1 - \sum f(i)^2)_R$$
$$= 7/15 * 0 + 8/15 * (1-(2/8)^2 - (6/8)^2) = 0.2$$

PSEUDO CODE FOR DECISION TREES

```
function BuildDecisionTree(L)
   Create node t from the learning sample Lt = L;
   calculate (im)purity
   if the stopping criterion is met for t then
     y^t = some constant value/class  (MAKE PREDICTION)
   else
      Find the split on Lt that maximizes impurity
      decrease
       s* = arg max Δi (s, t)
                   s∈Q
       Partition Lt into LtL ∪ LtR according to s*
             tL = BuildDecisionTree(LL)
             tR = BuildDecisionTree(LR )
   end if
   return t
end function
```

Code adapted from Gilles Louppe

stopping criterion
Gini (im)purity = 0

Gini (node L) =

$1 - \sum f(i)^2$

where f(i) is the frequency of the i-th class

Gini (splits L_L and Lr) =

$L_L/L * (1 - \sum f(i)^2) +$
$L_R/L * (1 - \sum f(i)^2)$

where f(i) is the frequency of the i-th class

Note:
splits happen along (single) features!

Decision Trees: Hiking Example

Day	Outlook	Temp.	Humidity	Wind	Go hiking?
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

Maximizing Entropy Gain

$$\text{Entropy }(S) = -\sum_i p_i \log_2 p_i$$

$$\text{Gain }(S, \mathscr{A}) = \text{Entropy }(S) - \sum_{\nu \in \mathscr{A}} \frac{|S^\nu|}{|S|} \text{Entropy }(S^\nu);$$

set **attributes e.g. outlook** **sizes of sets**

In this example, we will decide how to split that maximizes the entropy gain (instead of maximizing the Gini impurity decreases).

Decision Trees: Hiking Example

What conditions should we pick?

Day	Outlook	Temp.	Humidity	Wind	Go hiking?
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

$$\begin{aligned}&\text{Gain }(S,\text{outlook})\\ &= \text{Entropy }(S)\\ &\quad-\frac{5}{14}\text{ Entropy }(S,\text{outlook}=\text{sunny})\\ &\quad-\frac{4}{14}\text{ Entropy }(S,\text{outlook}=\text{outcast})\\ &\quad-\frac{5}{14}\text{ Entropy }(S,\text{outlook}=\text{rain})\\ &= 0.246\end{aligned}$$

$$\text{Entropy }(S) = -\frac{5}{14}\log\frac{5}{14} - \frac{9}{14}\log\frac{9}{14} = 1.245$$

irrespective of outlook

$$\text{Entropy }(S,\text{outlook}=\text{sunny}) = -\frac{2}{5}\log\frac{2}{5} - \frac{3}{5}\log\frac{3}{5} = 0.971$$

2 yes and 3 no

$$\text{Entropy }(S,\text{outlook}=\text{overcast}) = 0$$

all yes

$$\text{Entropy }(S,\text{outlook}=\text{rain}) = -\frac{3}{5}\log\frac{3}{5} - \frac{2}{5}\log\frac{2}{5} = 0.971$$

3 yes and 2 no

Decision Trees: Hiking Example

What conditions should we pick?

Day	Outlook	Temp.	Humidity	Wind	Go hiking?
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

$$\text{Gain}\ (S, \text{Outlook}) = 0.246$$

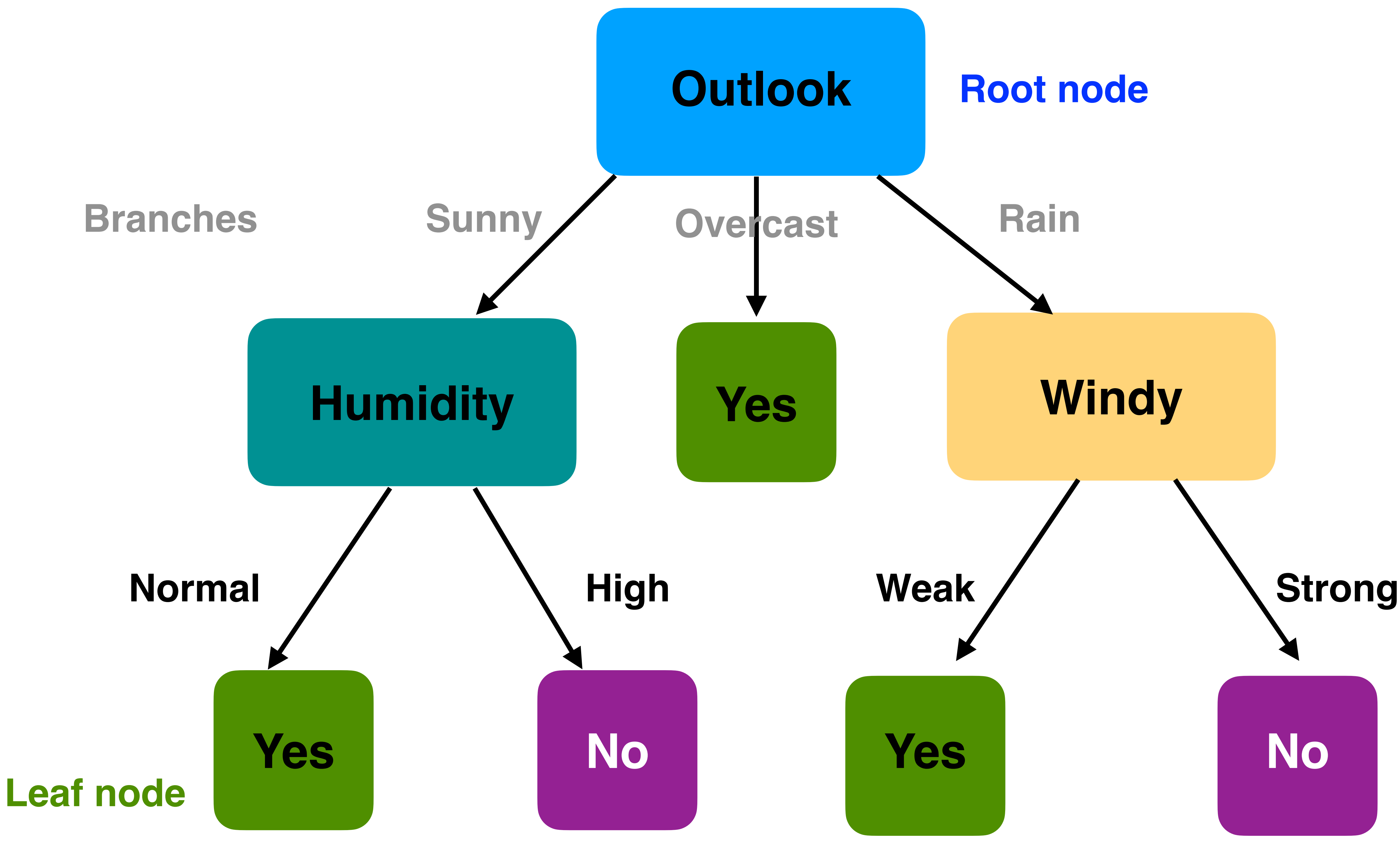
$$\text{Gain}\ (S, \text{Humidity}) = 0.151$$

$$\text{Gain}\ (S, \text{Wind}) = 0.048$$

$$\text{Gain}\ (S, \text{Temp.}) = 0.029$$

⇒ Choose Outlook maximizes the information gain

Decision Trees: Hiking Example

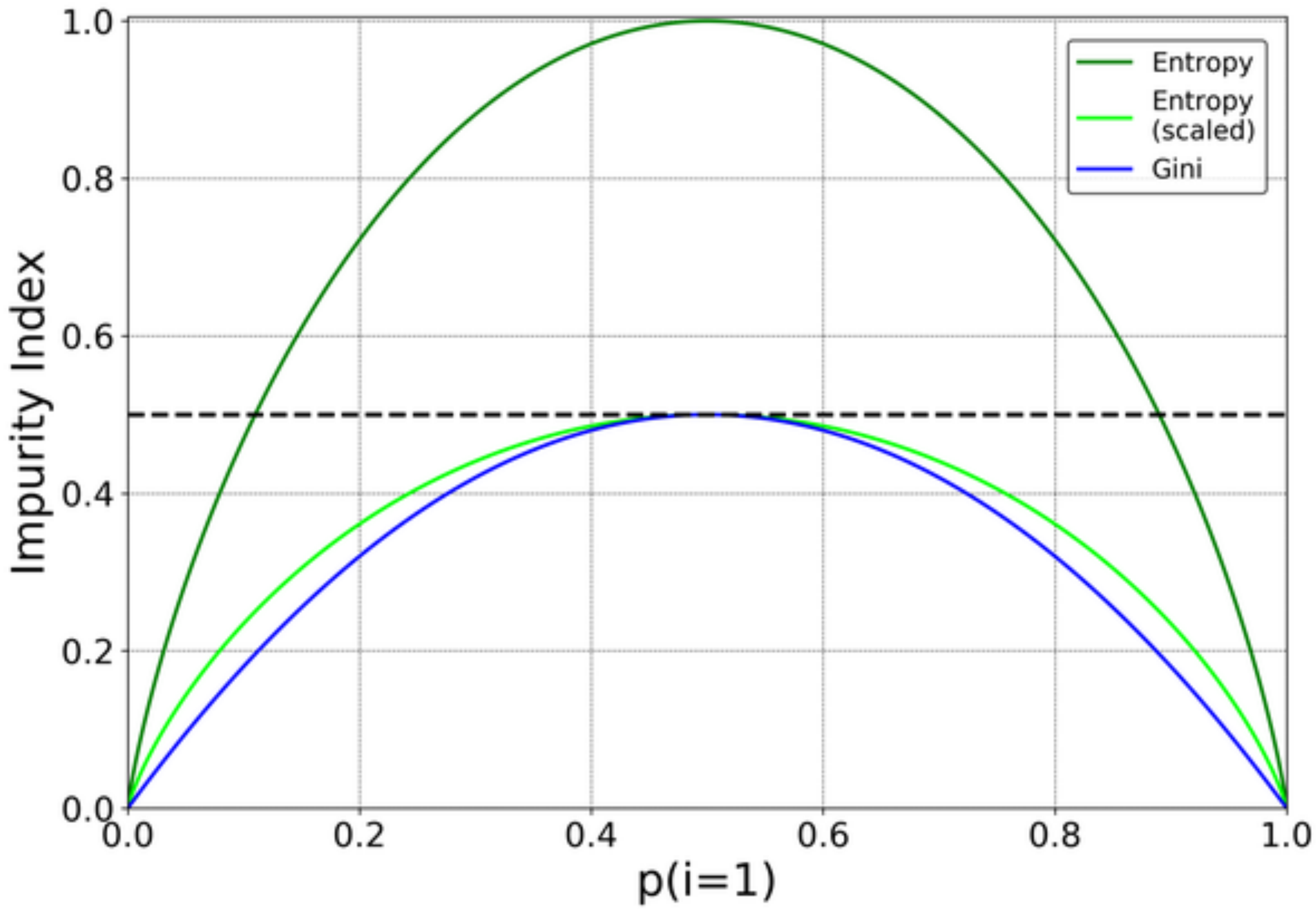


Gini vs Entropy

$$\text{Entropy }(S) = -\sum_i p_i \log_2 p_i$$

$$\text{Gini }(S) = 1 - \sum_i p_i^2 = 1 - p^2 - (1-p)2 = -2p^2 + 2p$$

two classes



options in sklearn implementation