Fundamentals of Artificial Intelligence

COMP307/AIML420
Decision Tree

Dr. Heitor Murilo Gomes
heitor.gomes@vuw.ac.nz
http://www.heitorgomes.com

Outline

1. **Why** should we care about decision trees?
2. **What** are decision trees?
3. **How** can we build decision trees?
4. Wrap-up and other considerations

Why?

1. Decision Trees (DTs) are the building blocks of ensemble methods, e.g. Random Forests [1] and XGBoost [2];

2. DTs are interpretable*, and their predictions can be explained* to domain experts;

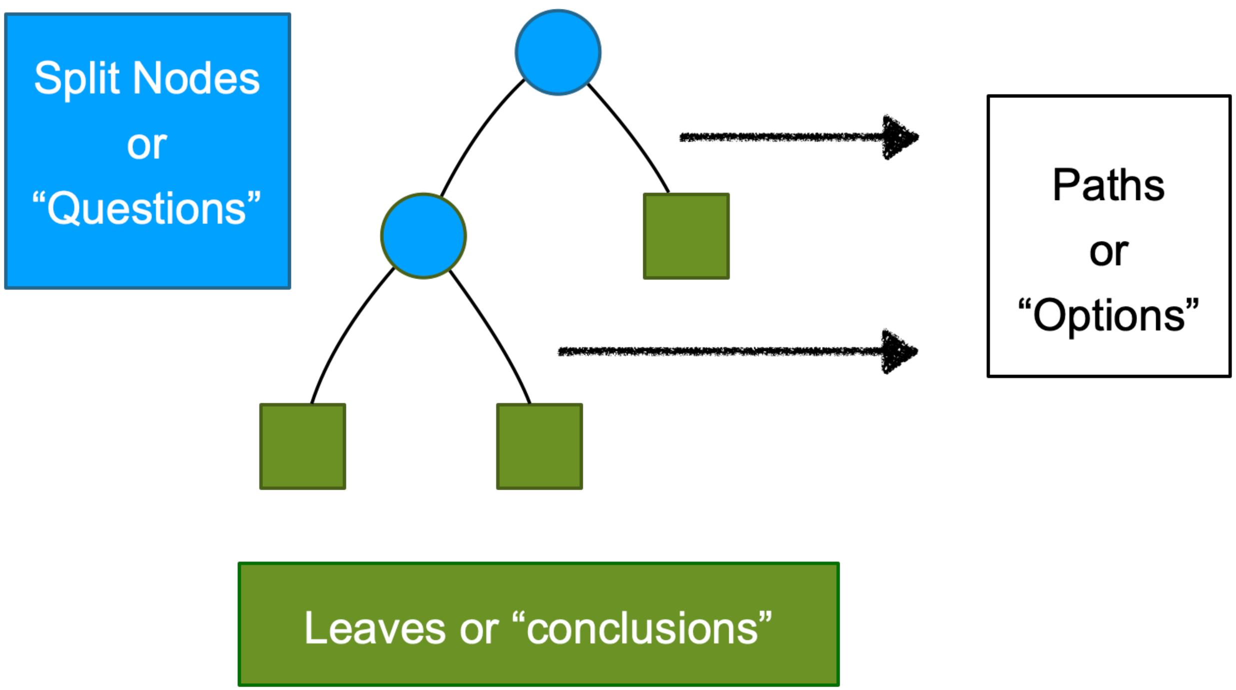
3. DTs are versatile and efficient: fast to train and predict, handle missing values, and they can be used for regression and classification.

[1] Breiman, Leo. "Random forests." Machine learning 45.1 (2001): 5-32.
[2] Chen, T., & Guestrin, C. “Xgboost: A scalable tree boosting system.” ACM SIGKDD (2016)

What?

The **"Decision Tree"** is a general algorithm used for building a model that makes predictions by learning a tree-structure based on the data

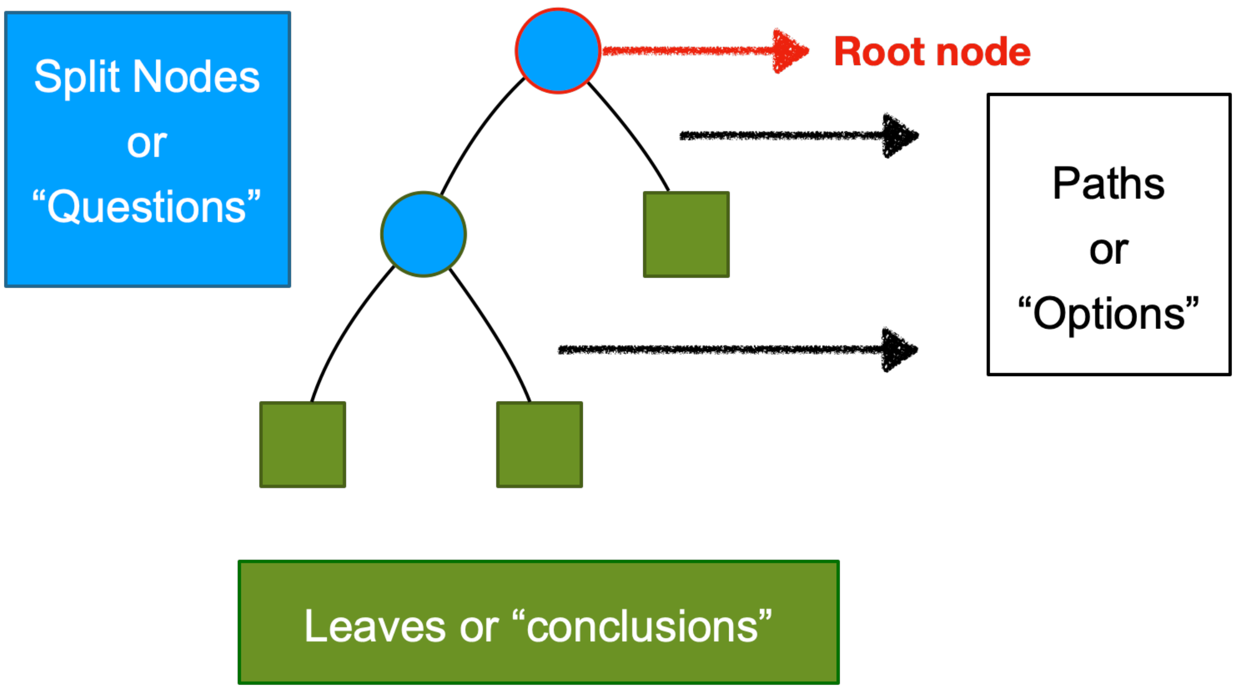
The 1st step is to understand the **structure** of a decision tree:



What?

The **“Decision Tree”** is a general algorithm used for building a model that makes predictions by learning a tree-structure based on the data

The 1st step is to understand the **structure** of a decision tree:



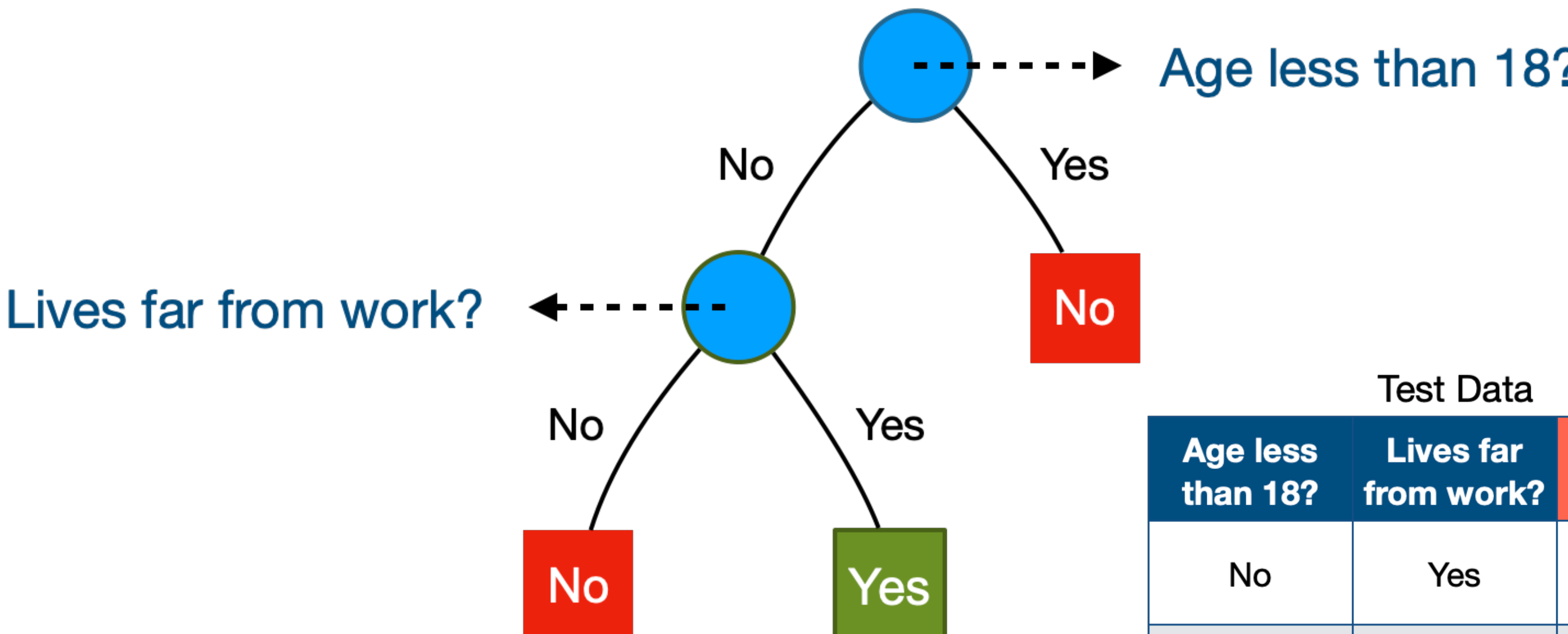
DT as a classifier

The **overall question** to be answered by the Decision Tree

> *Example: "Should you buy a car?*

Intermediary questions to answer the overall question

> *Example:*



Test Data

Age less than 18?	Lives far from work?	Buy a car?
No	Yes	
Yes	Yes	

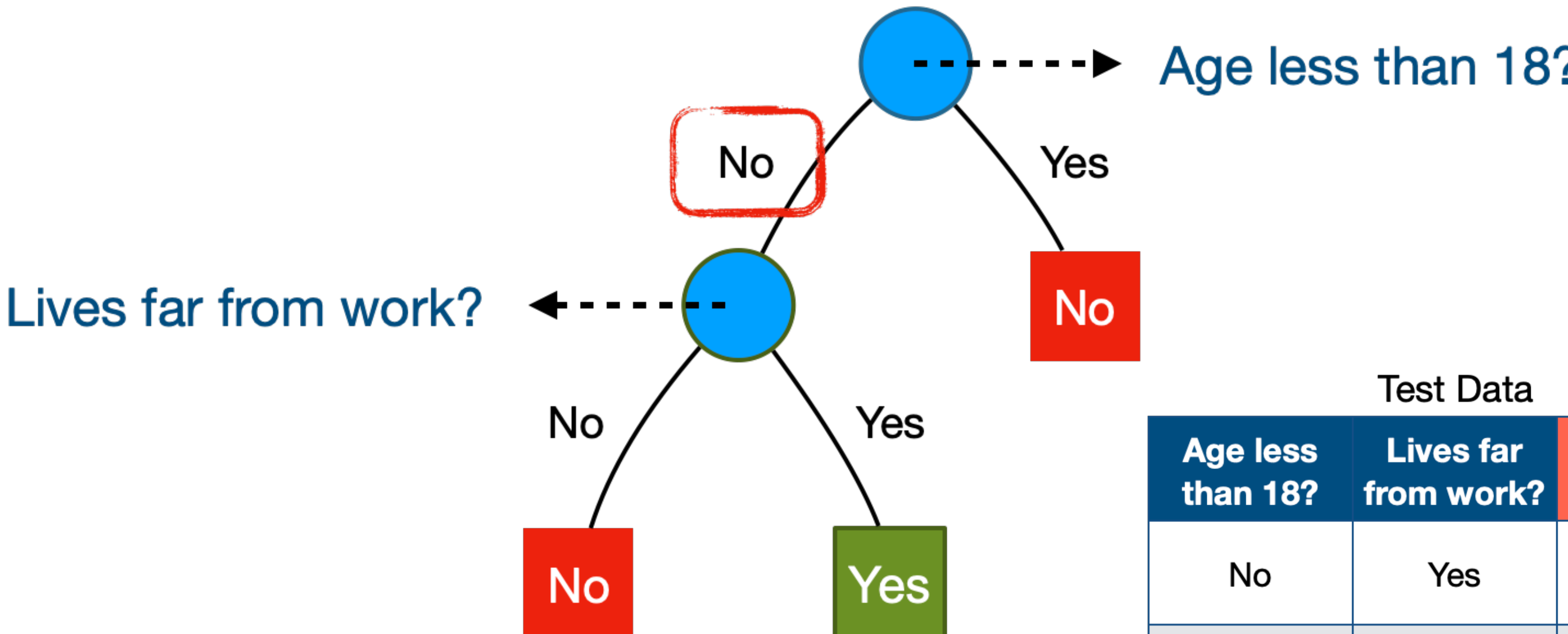
DT as a classifier

The **overall question** to be answered by the Decision Tree

> *Example: "Should you buy a car?*

Intermediary questions to answer the overall question

> *Example:*



Test Data

Age less than 18?	Lives far from work?	Buy a car?
No	Yes	
Yes	Yes	

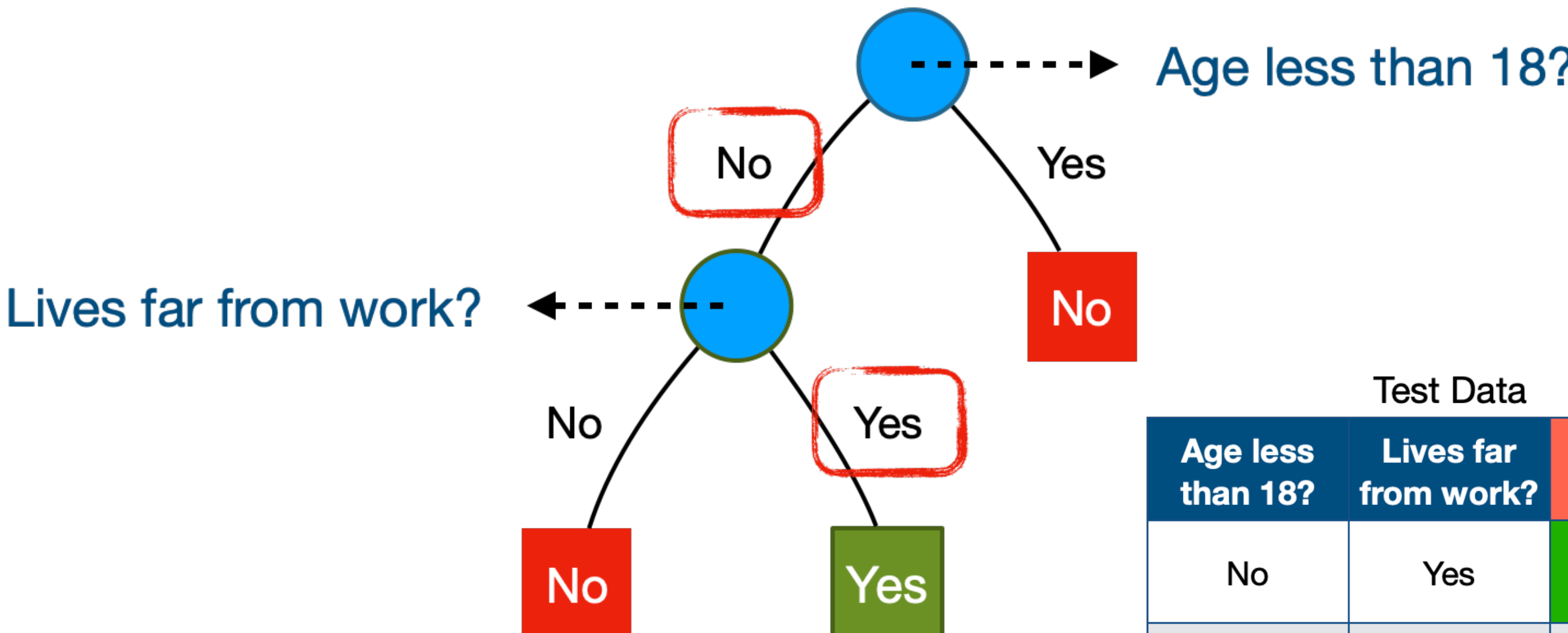
DT as a classifier

The **overall question** to be answered by the Decision Tree

> *Example: "Should you buy a car?*

Intermediary questions to answer the overall question

> *Example:*



Test Data

Age less than 18?	Lives far from work?	Buy a car?
No	Yes	Yes
Yes	Yes	

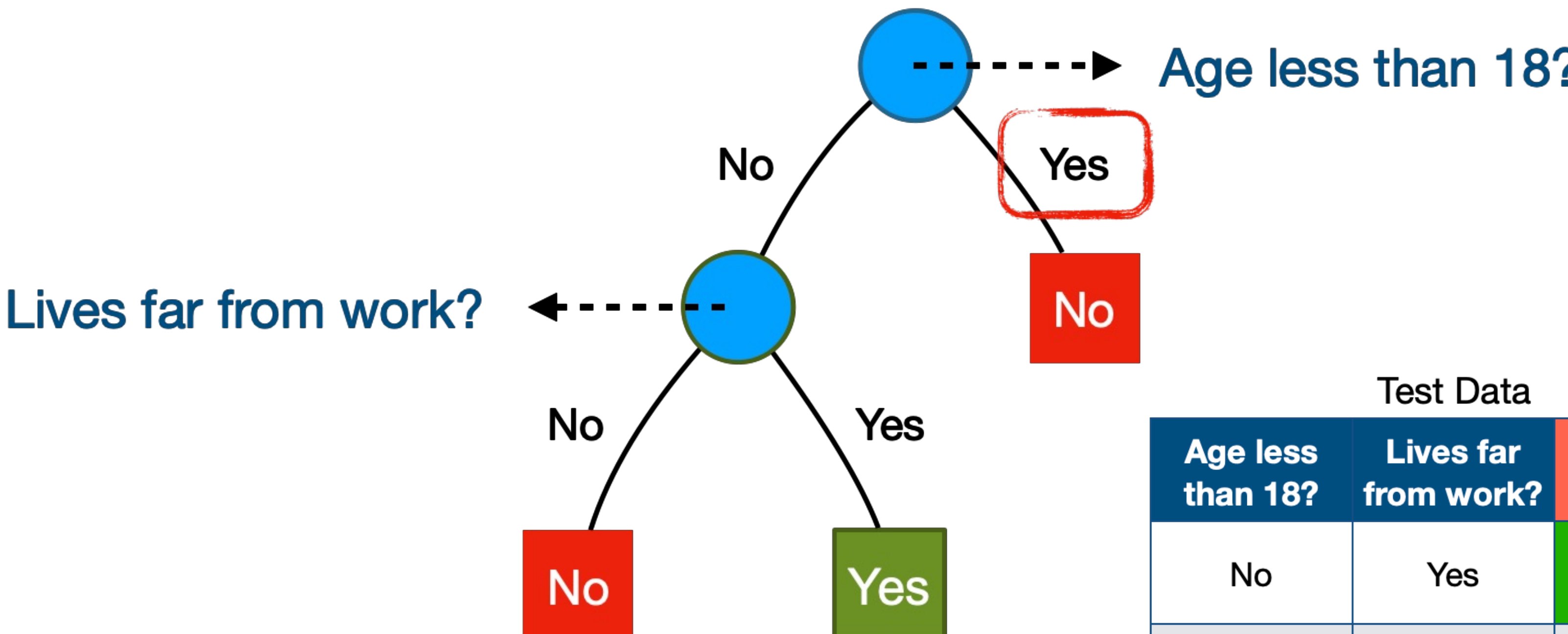
DT as a classifier

The **overall question** to be answered by the Decision Tree

> *Example: "Should you buy a car?*

Intermediary questions to answer the overall question

> *Example:*



Test Data

Age less than 18?	Lives far from work?	Buy a car?
No	Yes	Yes
Yes	Yes	

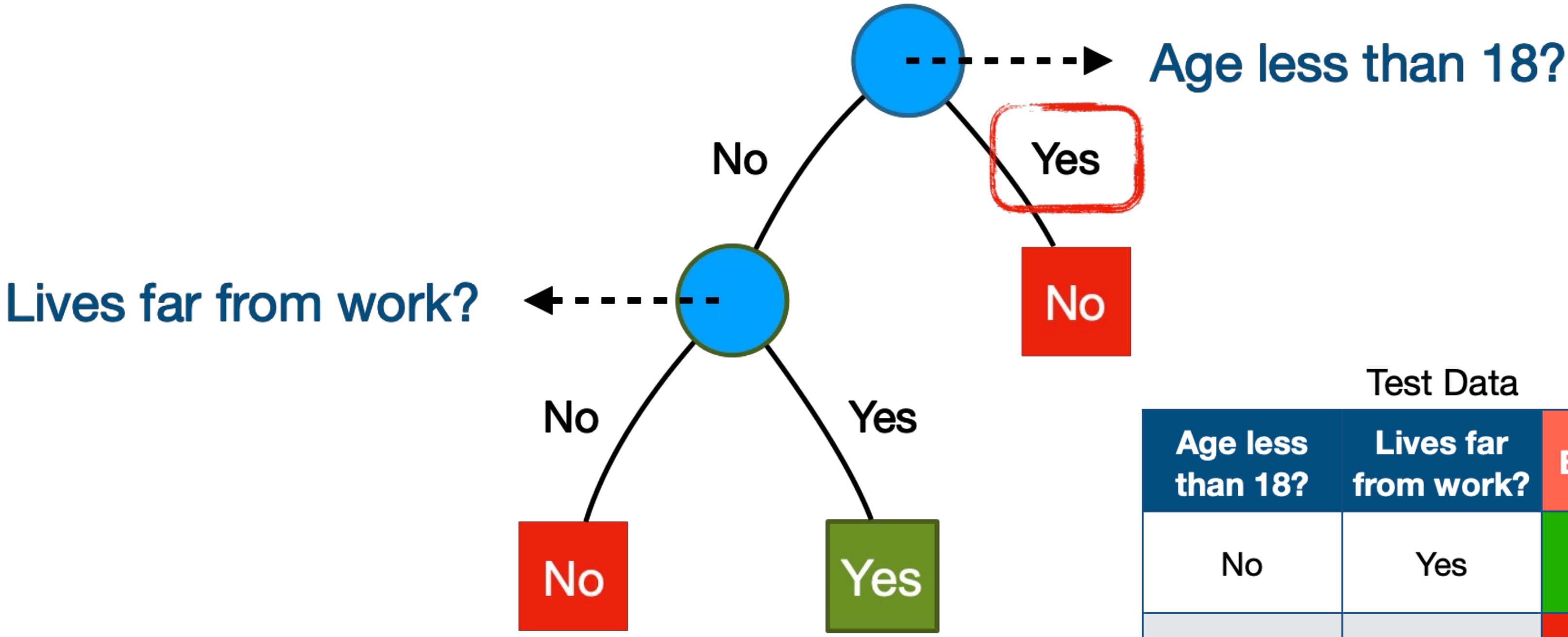
DT as a classifier

The **overall question** to be answered by the Decision Tree

> *Example: "Should you buy a car?*

Intermediary questions to answer the overall question

> *Example:*



Test Data

Age less than 18?	Lives far from work?	Buy a car?
No	Yes	Yes
Yes	Yes	No

DT as a classifier

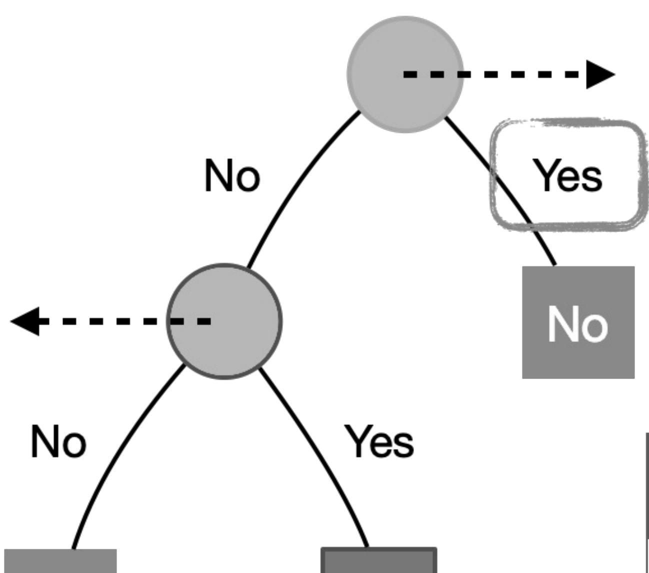
The **overall question** to be answered by the Decision Tree

> *Example: “Should you buy a car?*

Intermediary questions to answer the overall question

> *Example:*

The **class label**



Age less than 18?

Test Data

Age less than 18?	Lives far from work?	Buy a car?

DT as a classifier

The **overall question** to be answered by the Decision Tree

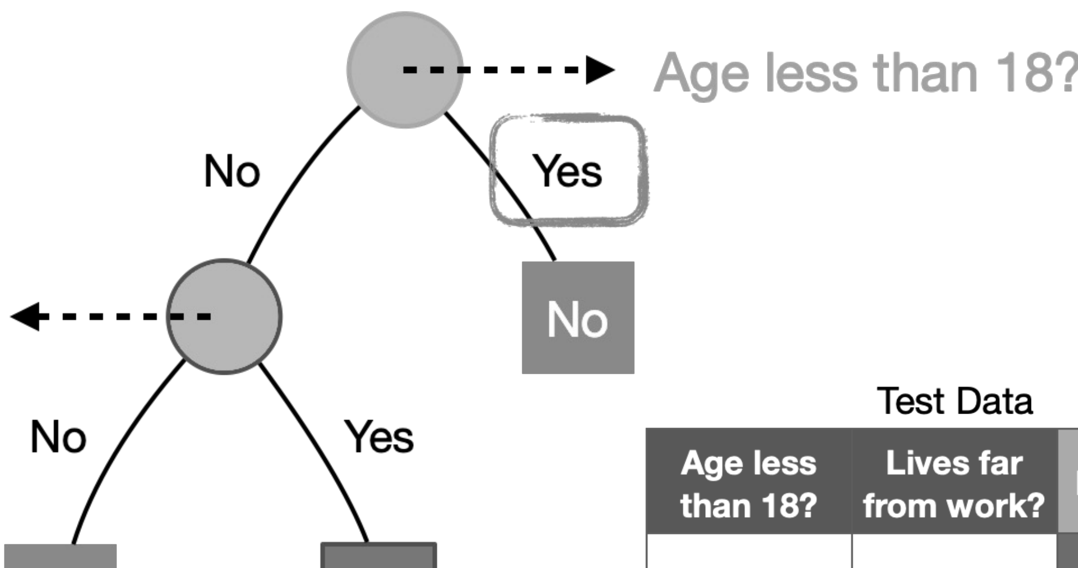
> *Example: “Should you buy a car?*

Intermediary questions to answer the overall question

> *Example:*

The **input features**

The **class label**



Test Data

Age less than 18?	Lives far from work?	Buy a car?

How?

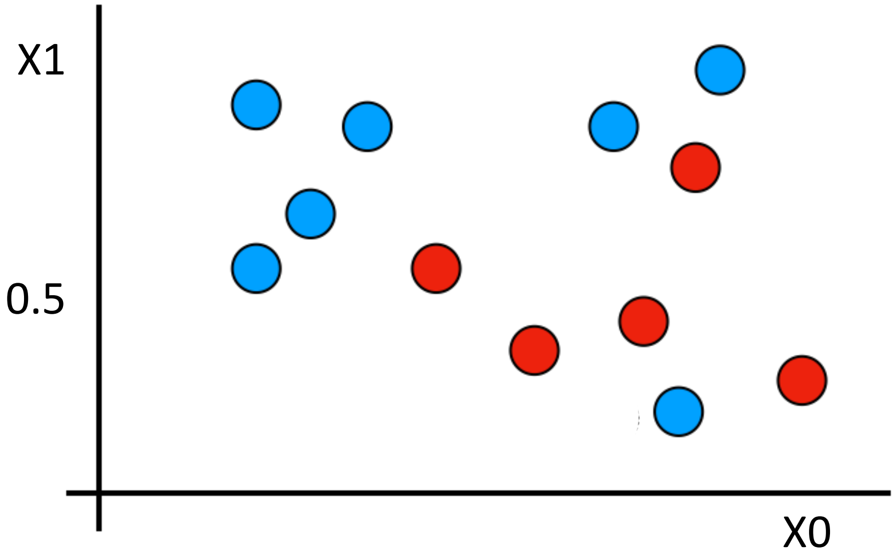
There are several algorithms for building decisions trees

- Classification and Regression Tree (CART)
- Iterative Dichotomiser 3 (ID3)
- C4.5
- C5.0
- And others…

General process: A greedy approach is used to divide the space according to some **(im)purity measure**

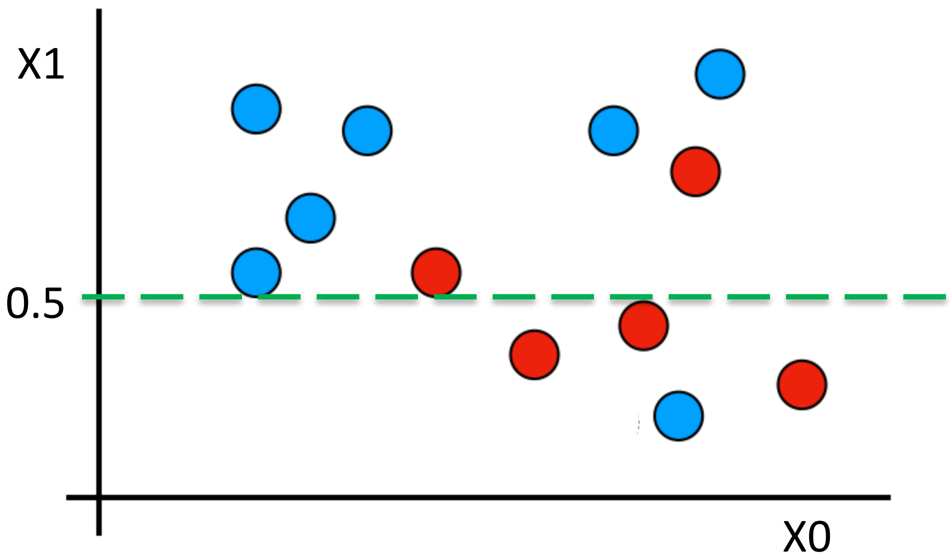
Divide the space

General process: A greedy approach is used to **divide the space** according to some **(im)purity measure**



Divide the space

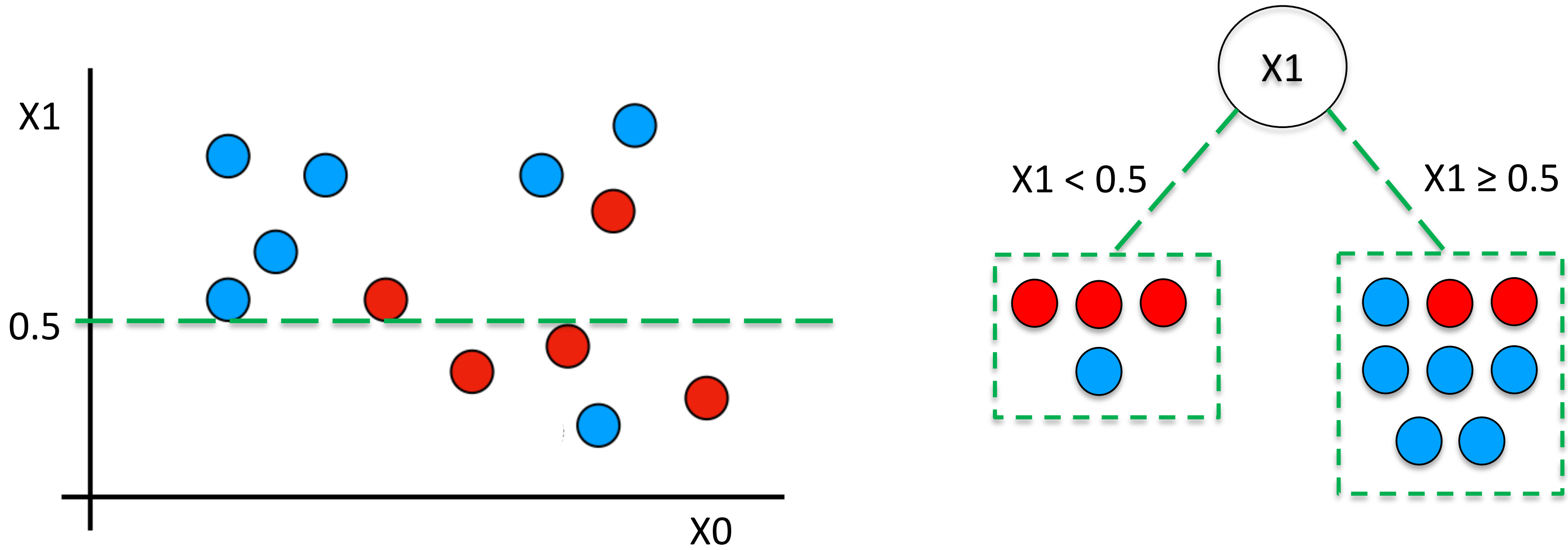
General process: A greedy approach is used to **divide the space** according to some **(im)purity measure**



* axis-parallel split

Divide the space

General process: A greedy approach is used to **divide the space** according to some **(im)purity measure**

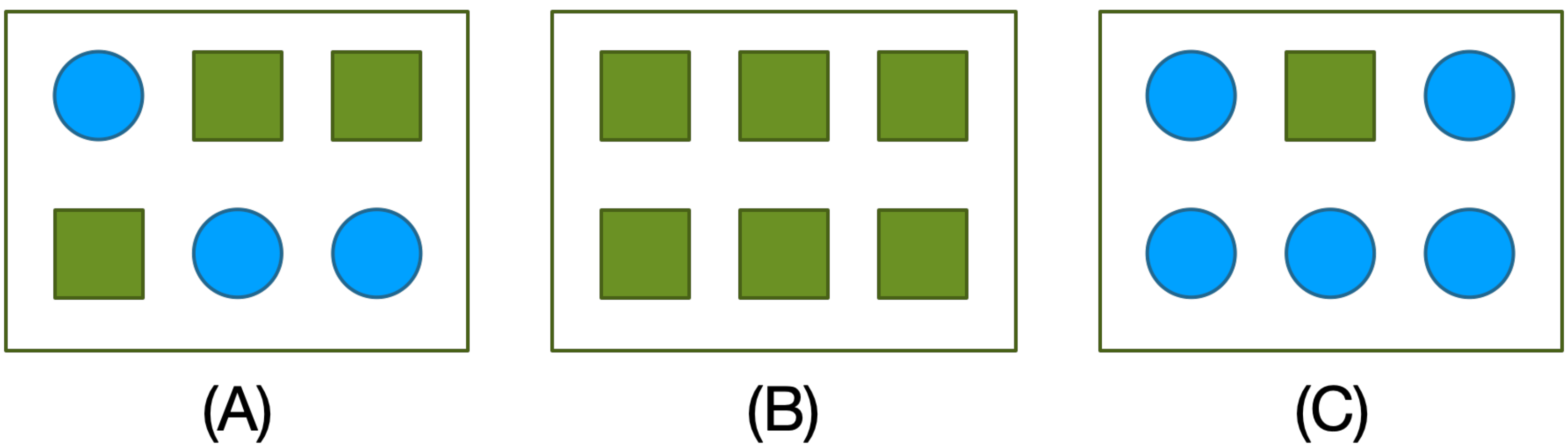


* axis-parallel split

(Im)purity measure

General process: A greedy approach is used to **divide the space** according to some **(im)purity measure**

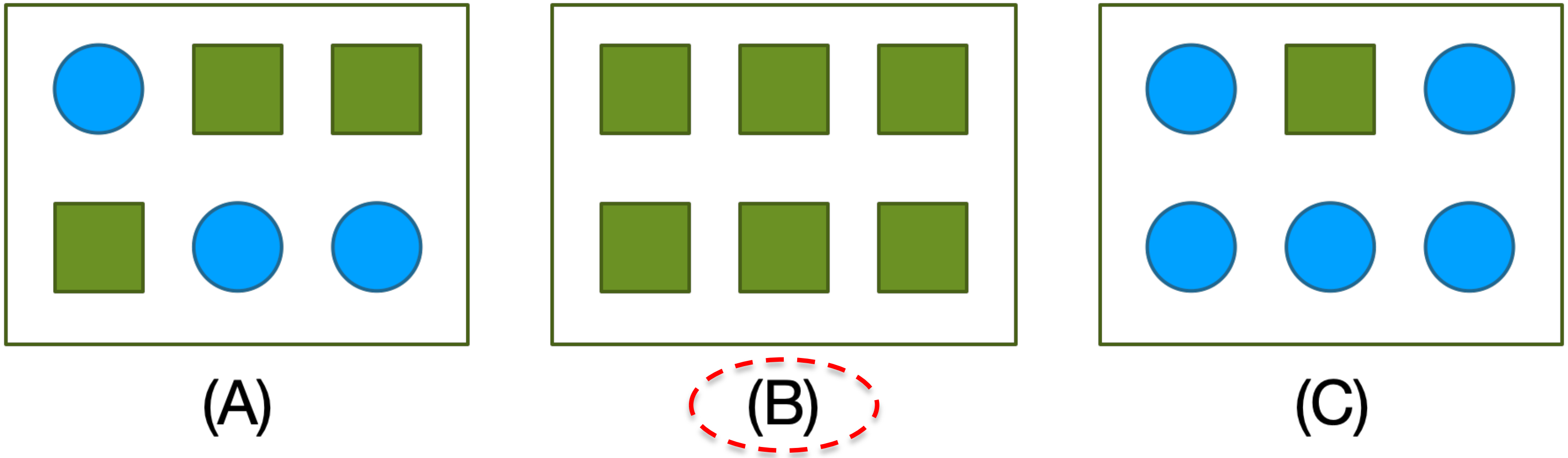
Which one of these sets is "purer"?



(Im)purity measure

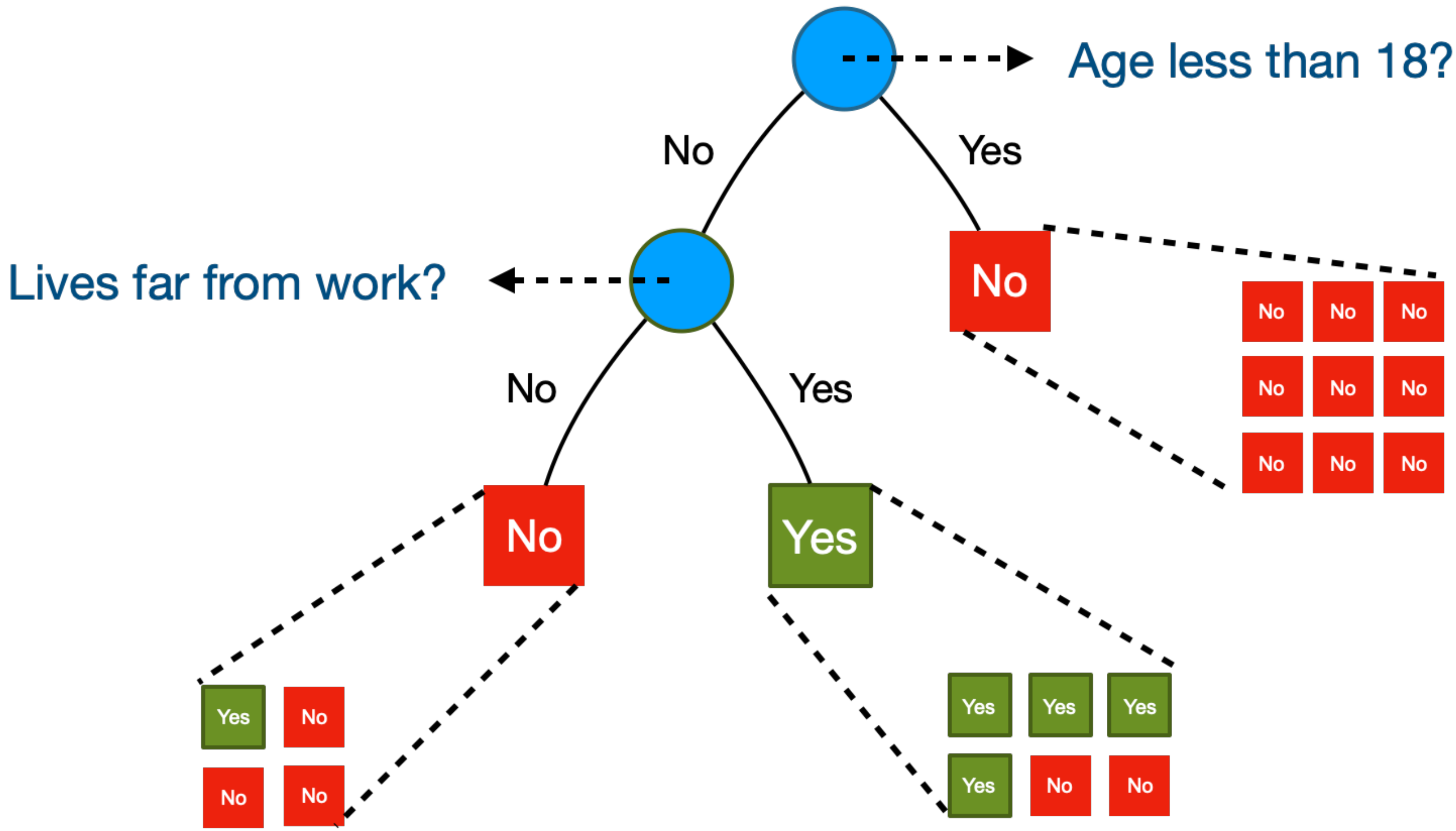
General process: A greedy approach is used to **divide the space** according to some **(im)purity measure**

Which one of these sets is more "pure"?



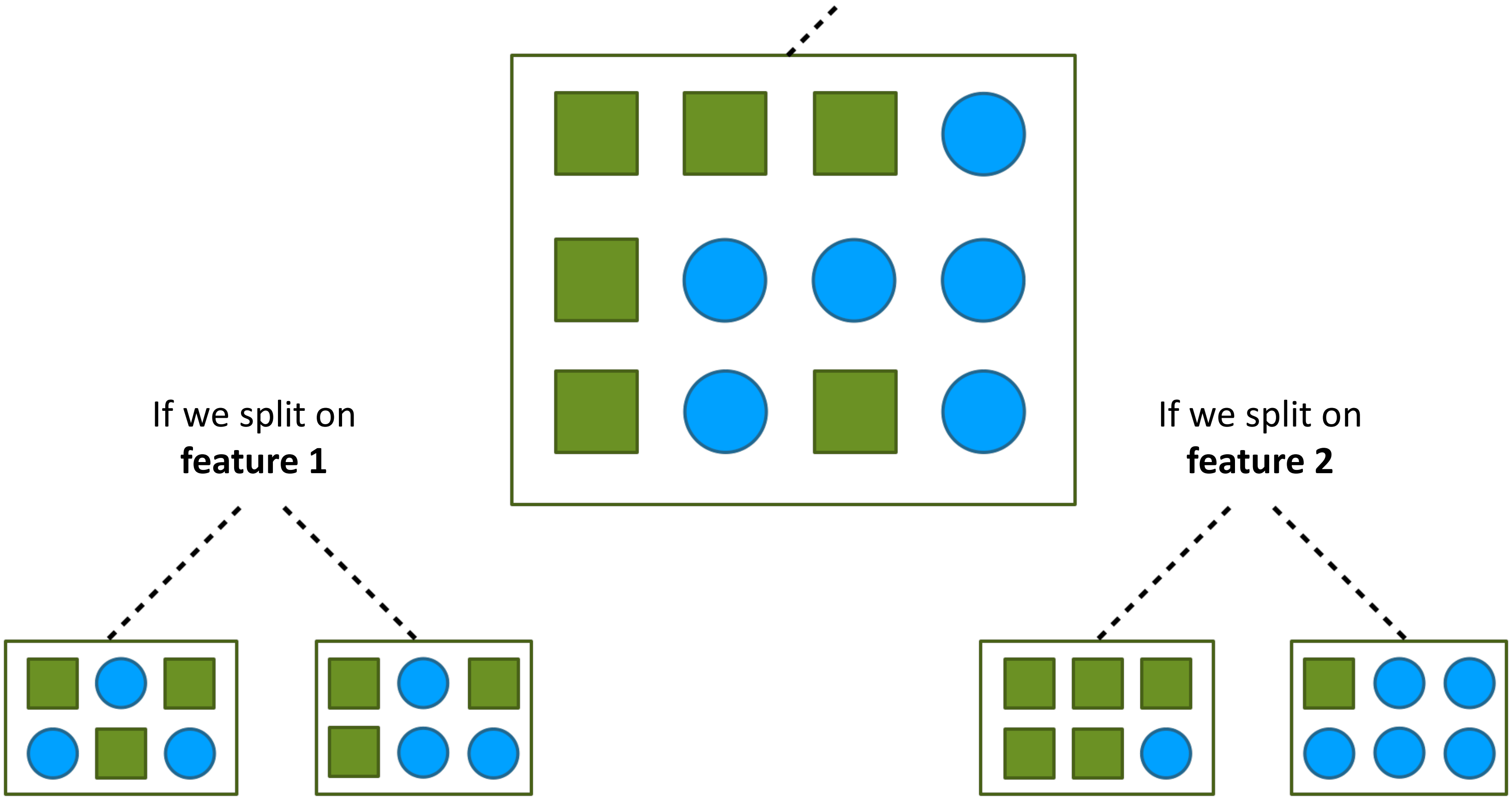
(Im)purity measure

Why do we care about the "purity"?



(Im)purity measure

What is the best approach for splitting this node?



Split candidate (1)?

Split candidate (2)?

Impurity measures

- When splitting, the goal is to find the feature that provides the **greatest reduction in impurity**

- This is typically done in a **greedy** way by examining each feature in turn and selecting the one with the highest reduction in impurity

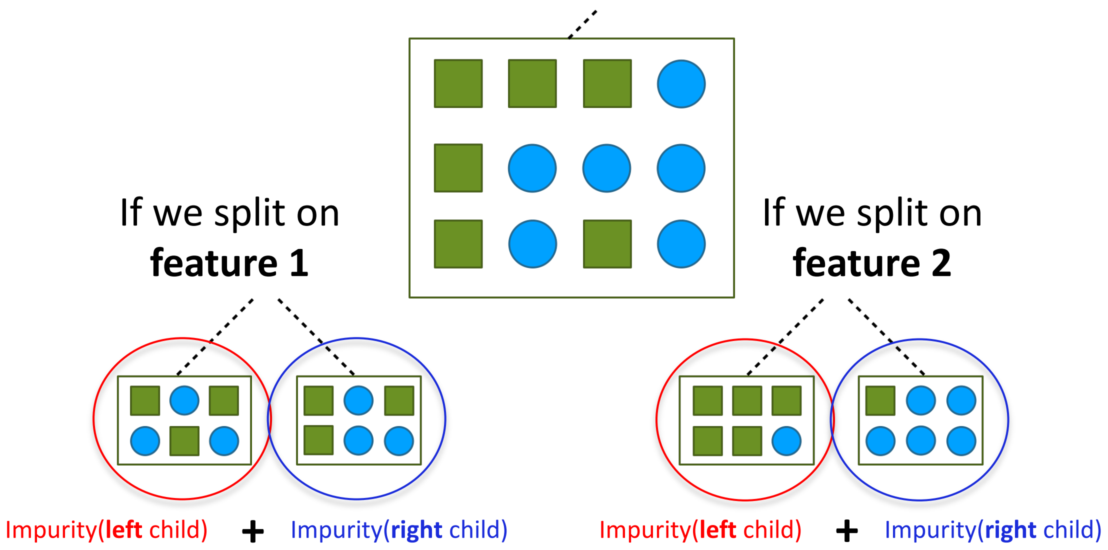
- Impurity measures
  - **Gini Impurity (or Index)**
  - **Entropy**

Choosing the next feature split

- While choosing the next split, we create two or more children* nodes, each with their own **impurity**

- We need to **combine** the children impurity and then choose whichever feature reduces impurity

- Example:



* For simplicity, here we focus on binary splits, but nominal attributes with more values can yield **multi-way splits**

Gini Impurity*

Intuition

If we pick two randomly selected instances from a population, they must belong to the same class

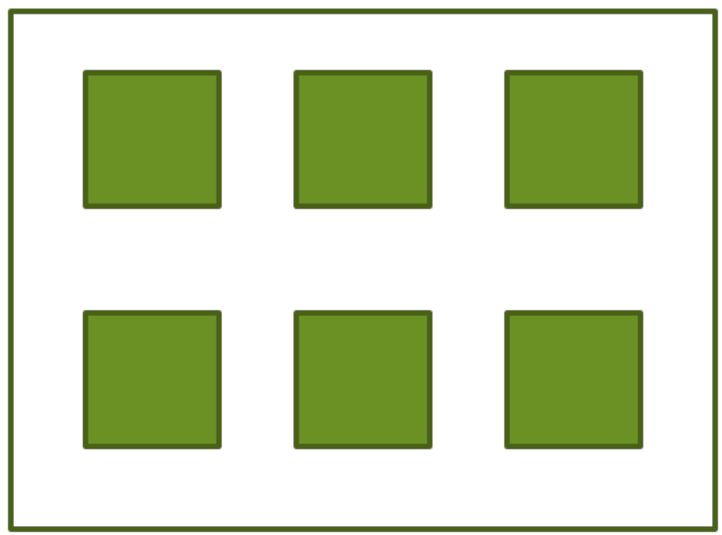
* Used in CART - Breiman, Leo. Classification and regression trees. Routledge, 1984

Gini Impurity

Intuition

If we pick two randomly selected instances from a population, they must belong to the same class

Intuitively, what is that probability if all instances belong to the same class?

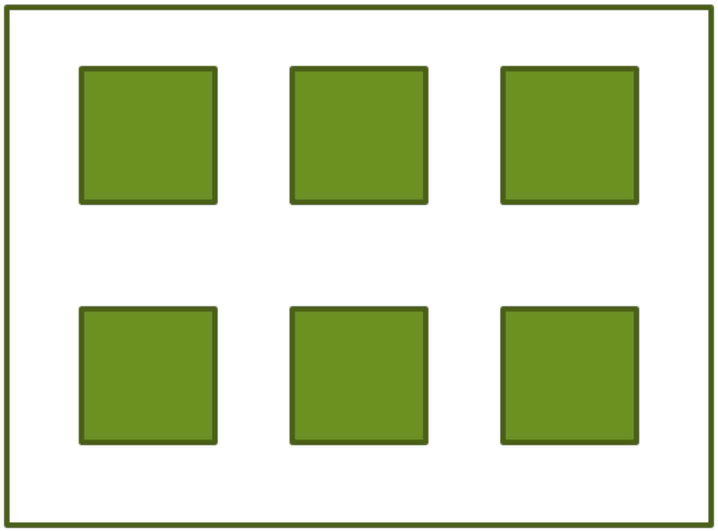


Gini Impurity

Intuition

If we pick two randomly selected instances from a population, they must belong to the same class

Intuitively, what is that probability if all instances belong to the same class?



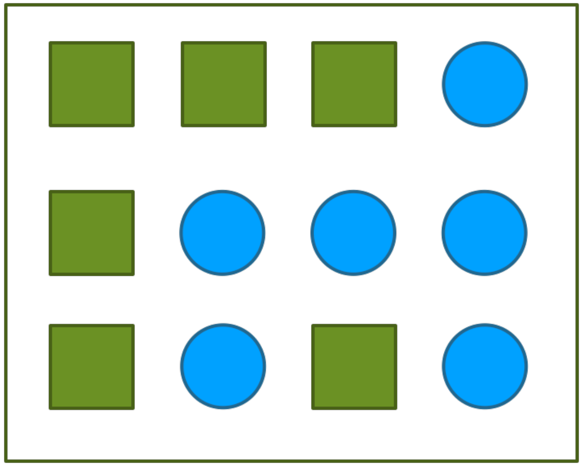
Probability = 1

Gini Impurity

Intuition

If we pick two randomly selected instances from a population, they must belong to the same class

What if they **don't belong** to the **same class**?



$$G = 1 - \sum P(i)^2$$

Where $P(i)$ is the proportion of instances in the node that belong to class *i*

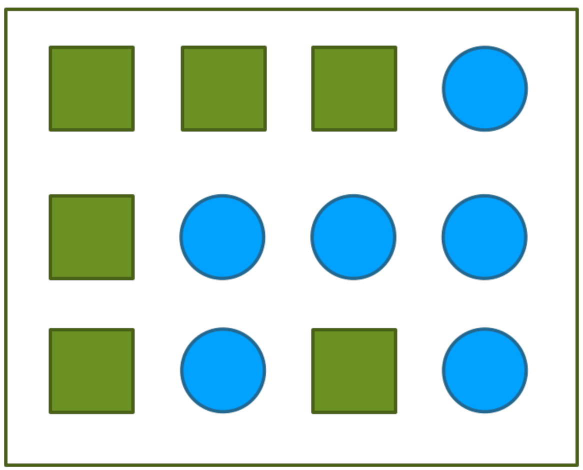
* **Higher** Gini Impurity values means **high** impurity

Gini Impurity

Intuition

If we pick two randomly selected instances from a population, they must belong to the same class

What if they **don't belong** to the **same class**?



$$G = 1 - \sum P(i)^2$$

Where $P(i)$ is the proportion of instances in the node that belong to class *i*

$$G = 1 - \left[(0.5)^2 + (0.5)^2\right] = 0.5$$

* Higher Gini Impurity values means less impurity

Information Entropy* **

Uses the *Entropy* to measure how much information can be obtained from a set of instances

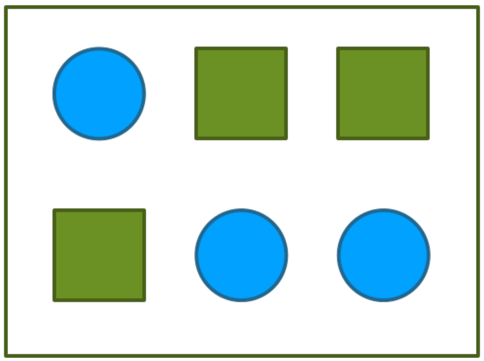
$$H = -\sum_{i=1}^{c} P(i) \cdot \log_2(P(i))$$

Where $P(i)$ is the proportion of instances in the node that belong to class *i, and c is the total number of classes*

* Information entropy was proposed in: Shannon, Claude E. "A mathematical theory of communication." *The Bell system technical journal* 27.3 (1948): 379-423.

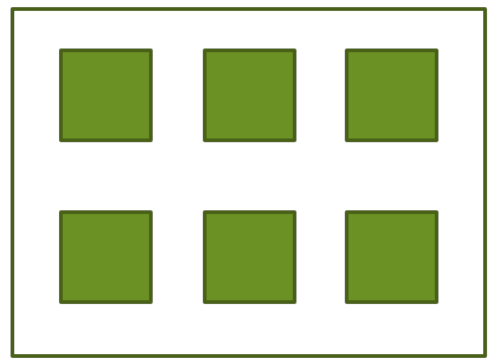
** Used in ID3: Quinlan, J. Ross. "Induction of decision trees." Machine learning 1 (1986): 81-106.

Entropy Examples



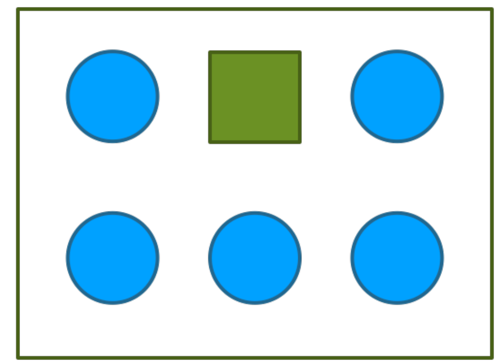
Blue = 0.5

Green = 0.5



Blue = 0

Green = 1.0

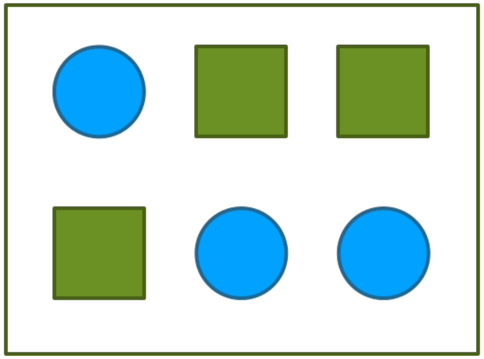


Blue = 0.83

Green = 0.17

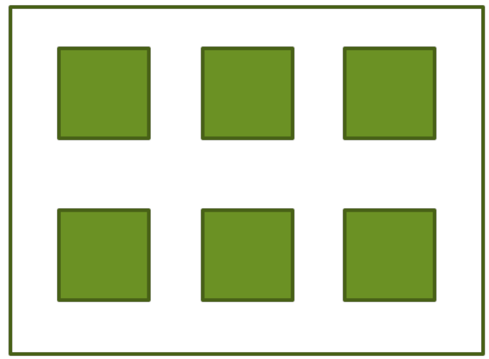
Entropy Examples

$$entropy = -(0.5 * \log(0.5) + 0.5 * \log(0.5)) = 1$$



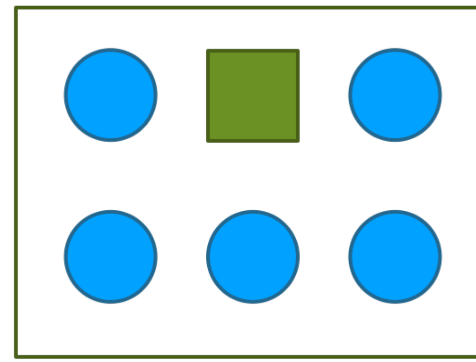
Blue = 0.5

Green = 0.5



Blue = 0

Green = 1.0

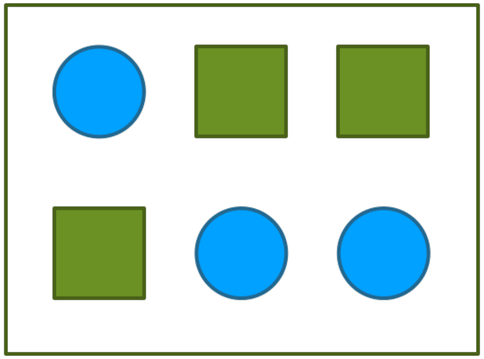


Blue = 0.83

Green = 0.17

Entropy Examples

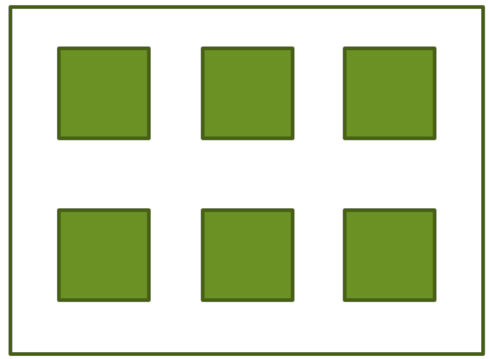
$entropy = -(0.5 * \log(0.5) + 0.5 * \log(0.5))$ = 1



Blue = 0.5

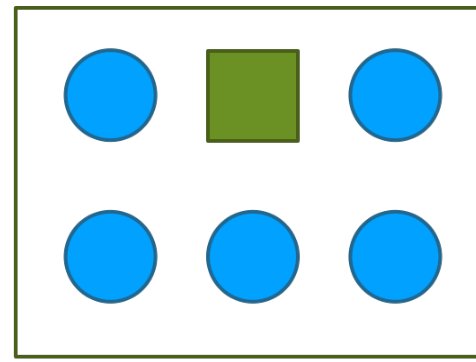
Green = 0.5

$entropy = -(0 * \log(0) + 1.0 * \log(1.0))$ = 0



Blue = 0

Green = 1.0

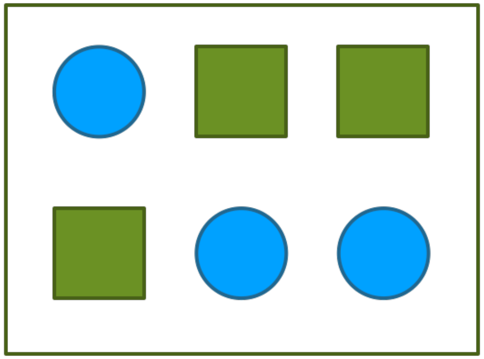


Blue = 0.83

Green = 0.17

Entropy Examples

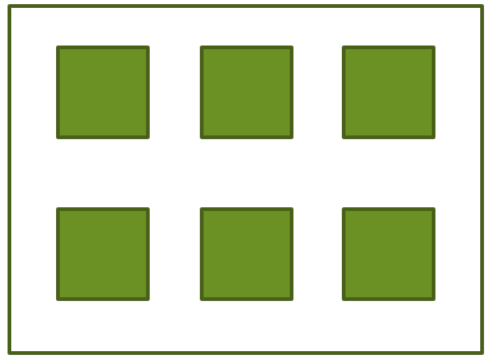
$entropy = -(0.5 * \log(0.5) + 0.5 * \log(0.5)) = 1$



Blue = 0.5

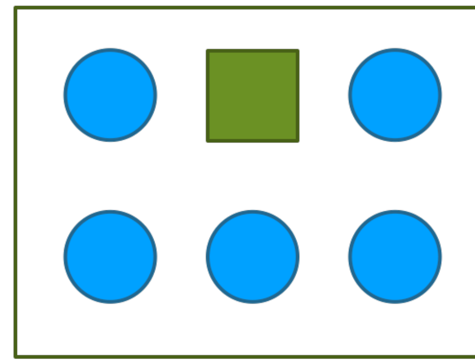
Green = 0.5

$entropy = -(0 * \log(0) + 1.0 * \log(1.0)) = 0$



Blue = 0

Green = 1.0



Blue = 0.83

Green = 0.17

$entropy = -(0.83 * \log(0.83) + 0.17 * \log(0.17)) = 0.65$

Information Gain (IG)*

The Information Gain is simply how much we reduce the entropy of a node by splitting it on a particular feature

In other words, the entropy of the node P minus the weighted entropy of the children nodes L and R that we obtain as we split on a given feature F

$$IG(F) = H(P) - \left[\left(\frac{N_L}{N_P}\right) \cdot H(L) + \left(\frac{N_R}{N_P}\right) \cdot H(R)\right]$$

More generally…where k is total number of children

$$IG(F) = H(P) - \sum_{l=1}^{k} \left(\frac{N_l}{N_P}\right) \cdot H(l)$$

* Same procedure when using Gini Impurity

General DT algorithm

Input: a set of instances with features and class labels (X and y)

Output: a decision tree classifier which performs classification

1. For each leaf node, compute if the set of instances is pure as possible
2. If a set is not pure, select the best **(unused in that path)** feature as the next node (**lowest impurity**)
3. Split the training data into sub-sets according to the chosen feature possible values
4. Recurse on each of the sub-sets

Considerations about DT

- **Training** and **Predicting**
  - In most ML algorithms training is costly, but predicting is efficient
  - What about DT?
- **Nominal features** are easy to handle, continuous features not so much
  - Requires discretizing the features or choosing a split point (or multiple split points)
- Decision trees can be adapted for **regression**
  - The main change is on the impurity measure (example: reduction in variance)
- Fully grown decision trees are prone to **overfitting**
  - It is doable to prune fully grown trees or stop spitting earlier (careful not to stop too early: underfit)
- **Gini index and Entropy** are both suitable impurity measures, but what is the difference between using one or the other?

Wrap up

- **DT** is a powerful **supervised learning** method
  - It is interpretable and serve as base learner for more advanced algo.

- Implementing a basic Decision Tree (DT) algorithm becomes easier if you spend sufficient time understanding the **role of Information Gain (IG) and entropy**, and if you're familiar with **recursive algorithms**.

Coming up next…

- ML examples (Tutorial this week)

- Ensembles (next week)