Decision Tree Version 1

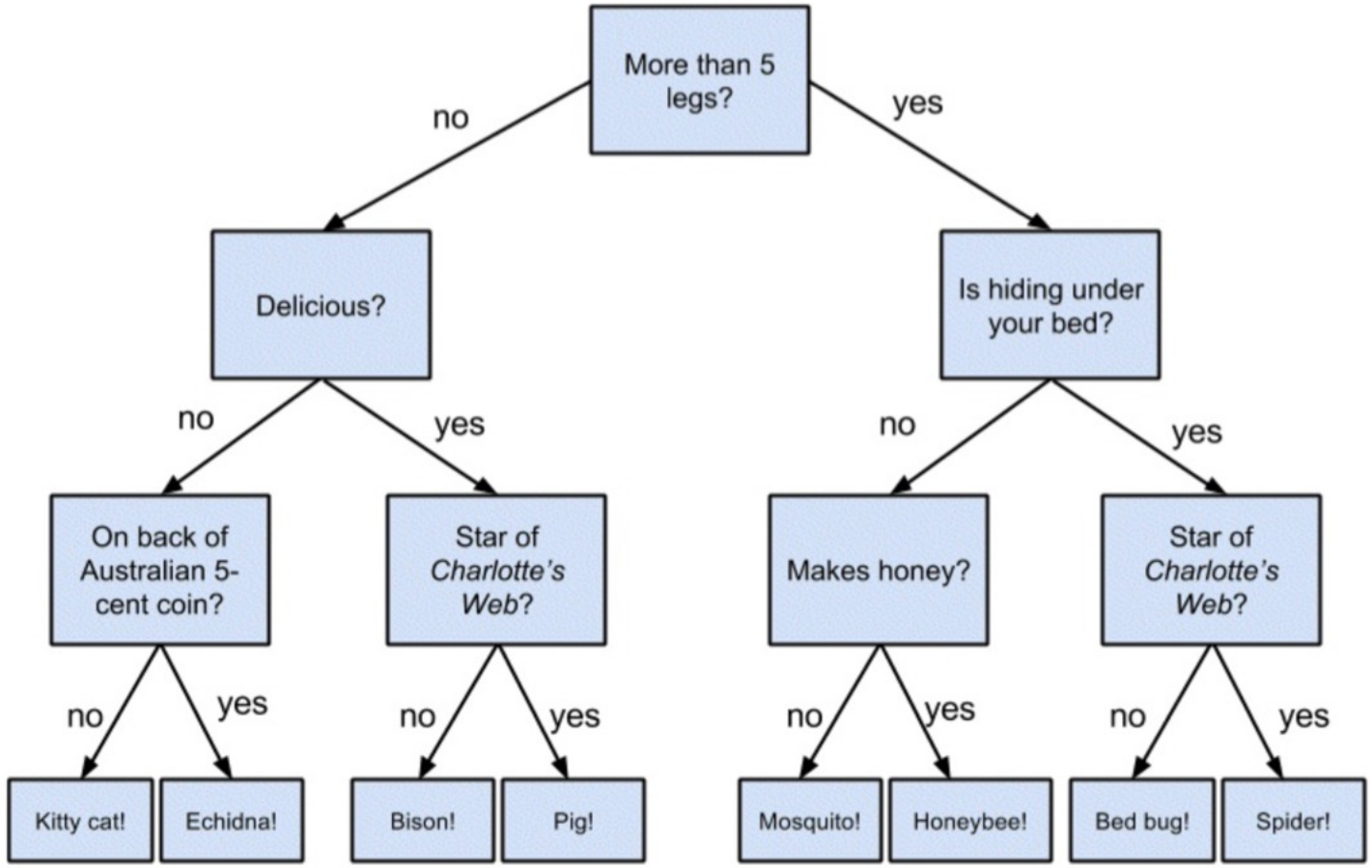
Contribute to 高宏宇教授、朱威達教授

這份資料是整合我修過課程中一些較為雷同的教材

為了閱讀方便，有以下註解可以參考

名詞：就是值得記住的名詞，以後會常常用到

Example: 就是可以快速理解要學什麼的示意圖



"*Guess the animal*" decision tree

Example

可以發現，decision tree就是一種結構

而這個結構，可以把獲得的新資料做分類

只要從root開始往下巡訪就行了

當到達leaf時，就可以獲得分類結果

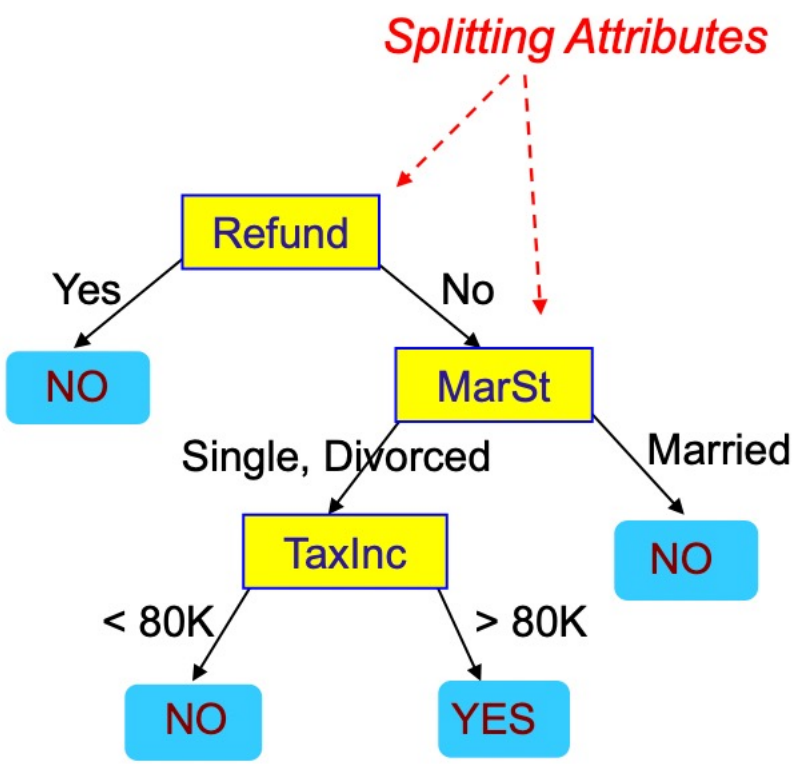
名詞

categorical categorical continuous class

Tid	Refund	Marital Status	Taxable Income	Cheat
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

Training Data

名詞



Model: Decision Tree

名詞 Test data

Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?

已經知道Decision Tree是什麼了

接下來就要知道如何建構這棵樹

我們知道一棵樹要形成，一定要有Node 和edge，

Node 在這邊扮演的角色是 Condition，意即巡訪至此，會問測試資料某個問題、條件，然後決定往哪走

Edge 在這邊扮演的角色是 Node 與 Node 之間的關係，也就是問題的先後順序

為了簡便說明

Node就是 Spliting attribute

而spliting attribute的選擇就是靠 Training Data決定

當樹建構完成，就可以喂測試資料進去，也就是 Testing Data，

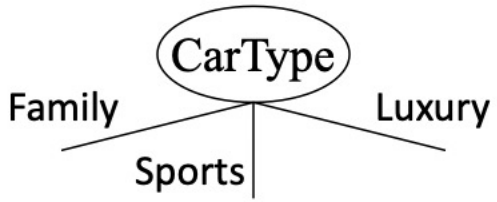
要注意的是Spliting attribute的資料型態有兩種，

一種就是一眼就看出label意義的categorical，

另一種就是看了也沒感覺的數值資料，也就是continuous連續資料 (但很有分析價值)

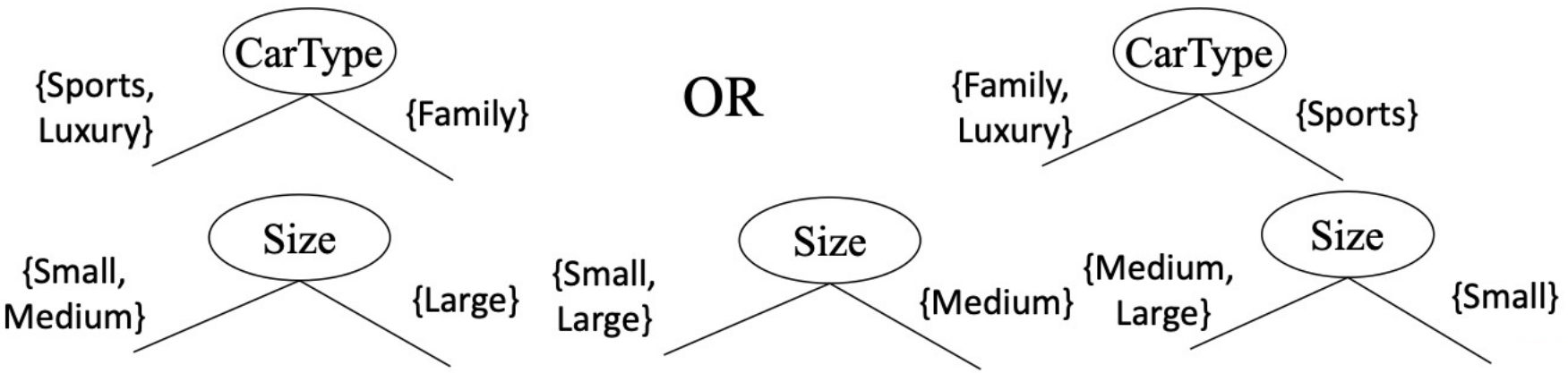
名詞

- Multi-way split: Use as many partitions as *distinct values*.



名詞

- Binary split: Divides values into two subsets. Need to find optimal partitioning.



Node 裡面有分支條件

要怎麼去分可以用幾句話概括

如果是一看就有label意義的attribute

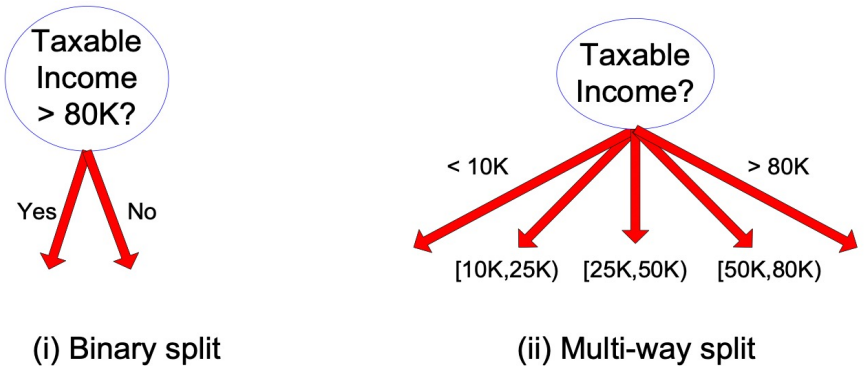
可以選擇multi-way的分法，一次分很多 Node

也可以選擇binary-way的分法，先分兩大堆，再做細分（有一點Quick sort的概念）

如果是連續型資料，就需要門檻，邊緣條件幫忙

可以選擇multi-way的分法，一次設立幾個區間(Interval)

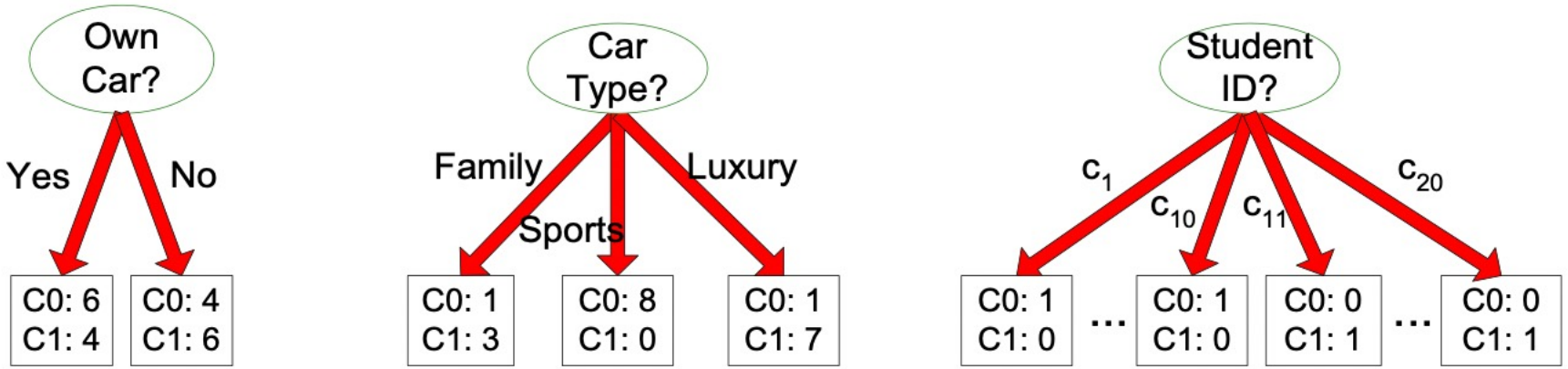
可以選擇binary-way的分法，只設一個Threshold，去分兩類



再來看一下，decision tree 在不同的attribute下的分類結果

C0:是一個欲分類結果（例如：可以看成沒病）

C1:是另一個結果（例如： 可以看成有病）



分得很爛

不管我有沒有車

分完兩派人馬還是的很均勻

分得很好

C0大多數人都是跑車

C1則有一些人是豪華車

所以用車種可以分得很開

有分跟沒分一樣

我用個體來分，每個人都有他的種類

在一開始就知道了

要怎麼去分是decision Tree最大的議題，也是分析的要點

從直觀去想，我就是要選擇一個attribute可以把資料分得越開越好，

巴不得是那種attribute 0就代表class 0, attribute 1就代表class 1

所以我們需要量化參數來評估 “什麼叫越開越好 “

有三個

1. Entropy
2. Gini Index
3. Misclassification error

會依照上述順序講解

- Greedy approach: 名詞
  - Nodes with homogeneous class distribution are preferred
- Need a measure of node impurity: 名詞

C0: 5 C1: 5	C0: 9 C1: 1
Non-homogeneous, High degree of impurity	Homogeneous, Low degree of impurity 名詞

Entropy 熵

我們一定會想到一個方法：比比看分支前，分支後的差異性，

如果差異越大（原本很混雜），代表我分得還不賴。

「我分得還不賴」是一個資訊，而且是一個很不錯的資訊量（大）

因此我們導入可以算「資訊量」的數值 Entropy，「機率」就摻和進來了

越大的資訊量反應出「Entropy減少很多」，因此我們要得到的就是Entropy的差值

$$\text{Entropy:} \quad H(V) = \sum_k P(v_k) \log_2 \frac{1}{P(v_k)} = -\sum_k P(v_k) \log_2 P(v_k)$$

題外話：如果有一枚出老千的硬幣，有很大的機率擲出人頭，若我拿著這枚硬幣擲出數字，那很意外（資訊量大）

$$\text{Entropy:} \quad H(V) = \sum_k P(v_k) \log_2 \frac{1}{P(v_k)} = -\sum_k P(v_k) \log_2 P(v_k)$$

$$H(Loaded) = -(0.99 \log_2 0.99 + 0.01 \log_2 0.01) \approx 0.08 \text{ bits.}$$

我擲了10次

出現了5 人頭 5 數字

$H\left(\frac{5}{10}\right) = -(0.5log_2 0.5 + 0.5log_2 0.5) = 1\ bits$

$\text{Information} = H\left(\frac{5}{10}\right) - H(loaded) = 1 - 0.08 = 0.92bits$（大）

資訊量大，是時候去買樂透了

我擲了10次

出現了9 人頭 1 數字

$\text{Information} = H\left(\frac{9}{10}\right) - H(loaded)$

$= 0.467\ bits - 0.08\ bits = 0.387bits$（相對小）

普通人

$$\text{Entropy:} \quad H(V) = \sum_k P(v_k) \log_2 \frac{1}{P(v_k)} = -\sum_k P(v_k) \log_2 P(v_k)$$

因此，當我們試用不同的attribute去分支時，每個分支都有它的entropy，

我們會把每個分支的entroy加總後與原本的entropy做相減，得到Information

在based on 這個information gain去選擇產生最大gain的attribute 作為node

$$Remainder(A) = \sum_{k=1}^{d} \frac{p_k+n_k}{p+n} B(\frac{p_k}{p_k+n_k})$$

$$Gain(A) = B(\frac{p}{p+n}) - Remainder(A)$$