D24：決策樹演算法(Decision Tree)

課程閱讀 本日作業 問題討論 學習心得(完成)

- 重要知識點 >
- 分類與回歸 >
- 什麼是決策樹？ >

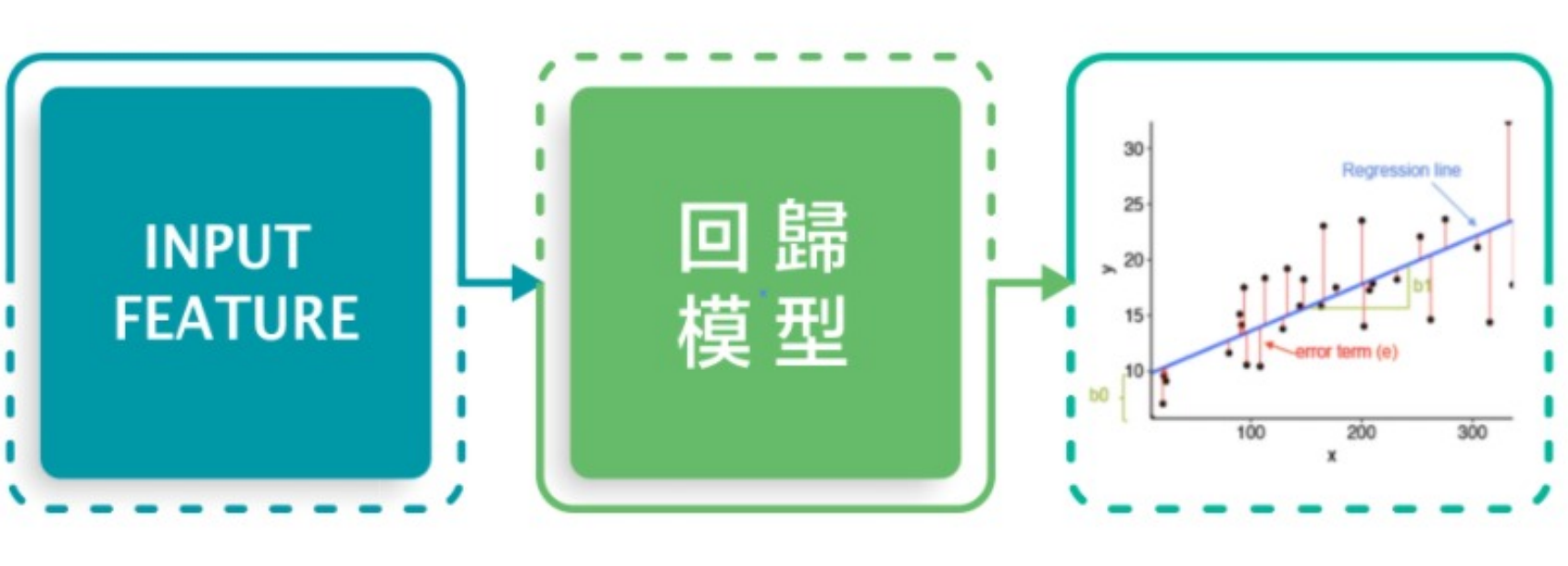
重要知識點

- 了解何謂決策樹模型
- 了解何謂決策樹 Feature Importance

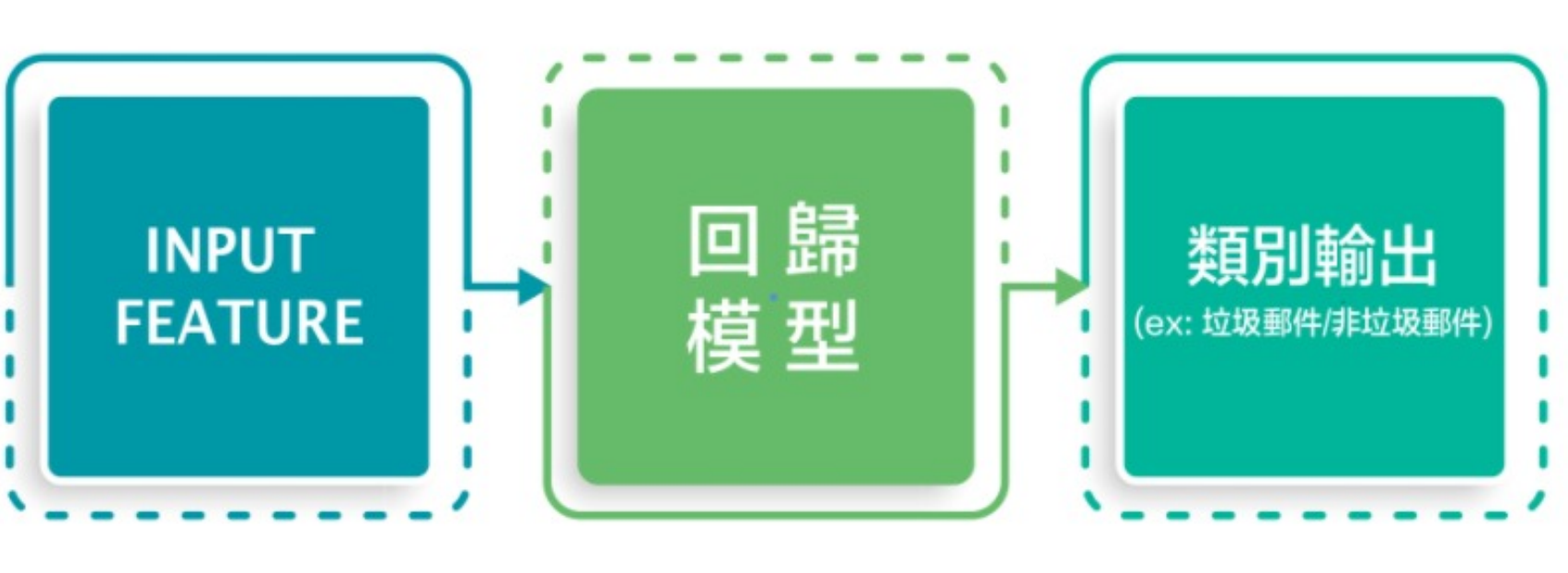
分類與回歸

開始介紹決策樹模型前，先讓學員瞭解在機器學習 Supervised Learning 中的分類(Classification) 與回歸(Regression) 的概念，而決策樹也是在這兩種問題中常用的機器學習模型。

回歸模型：給定輸入資料特徵，模型輸出**連續預測值** **(ex：房價、股價預測)**

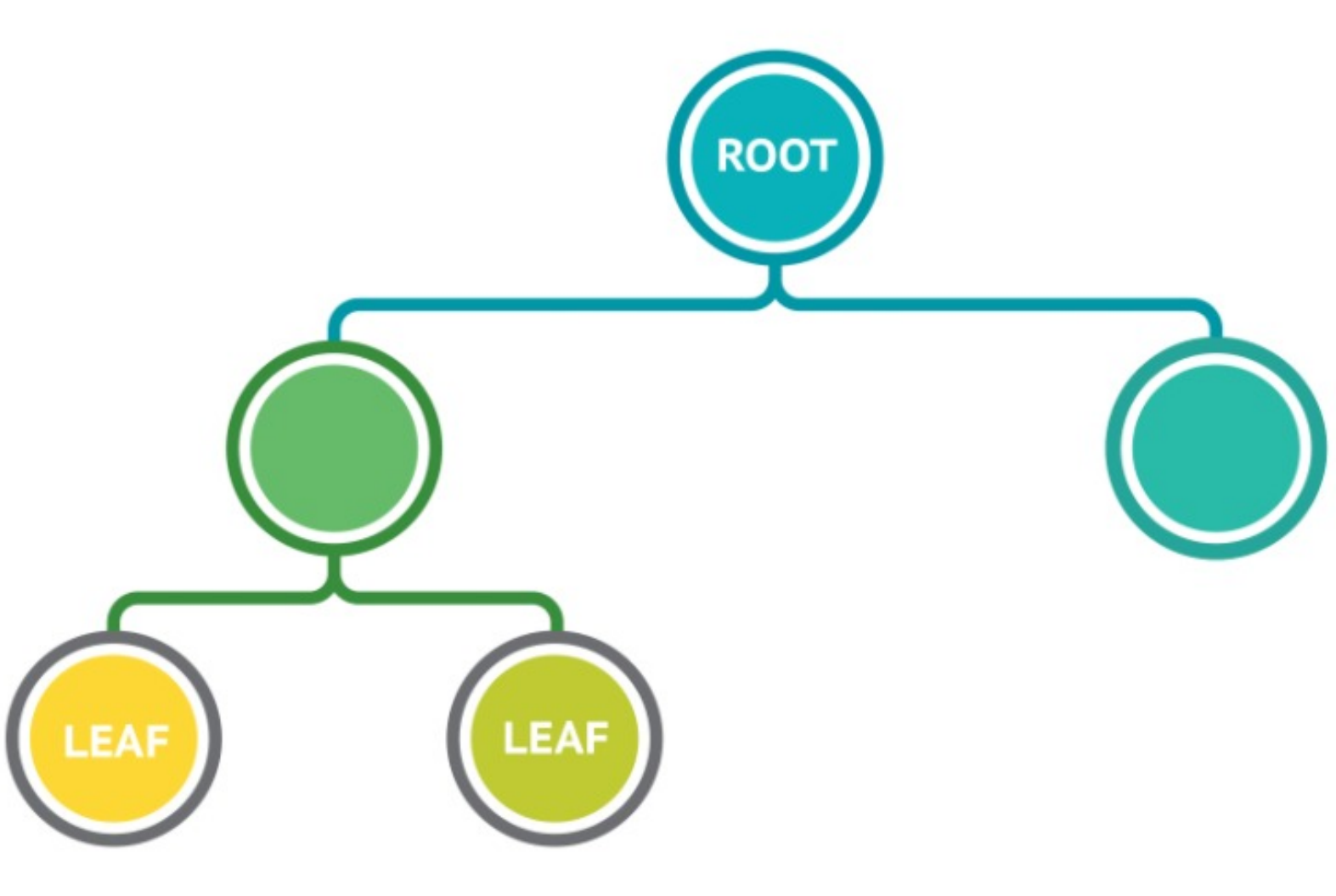


分類模型：給定輸入資料特徵，模型輸出**離散類別預測** **(ex：文章情緒分類、垃圾郵件分類)**

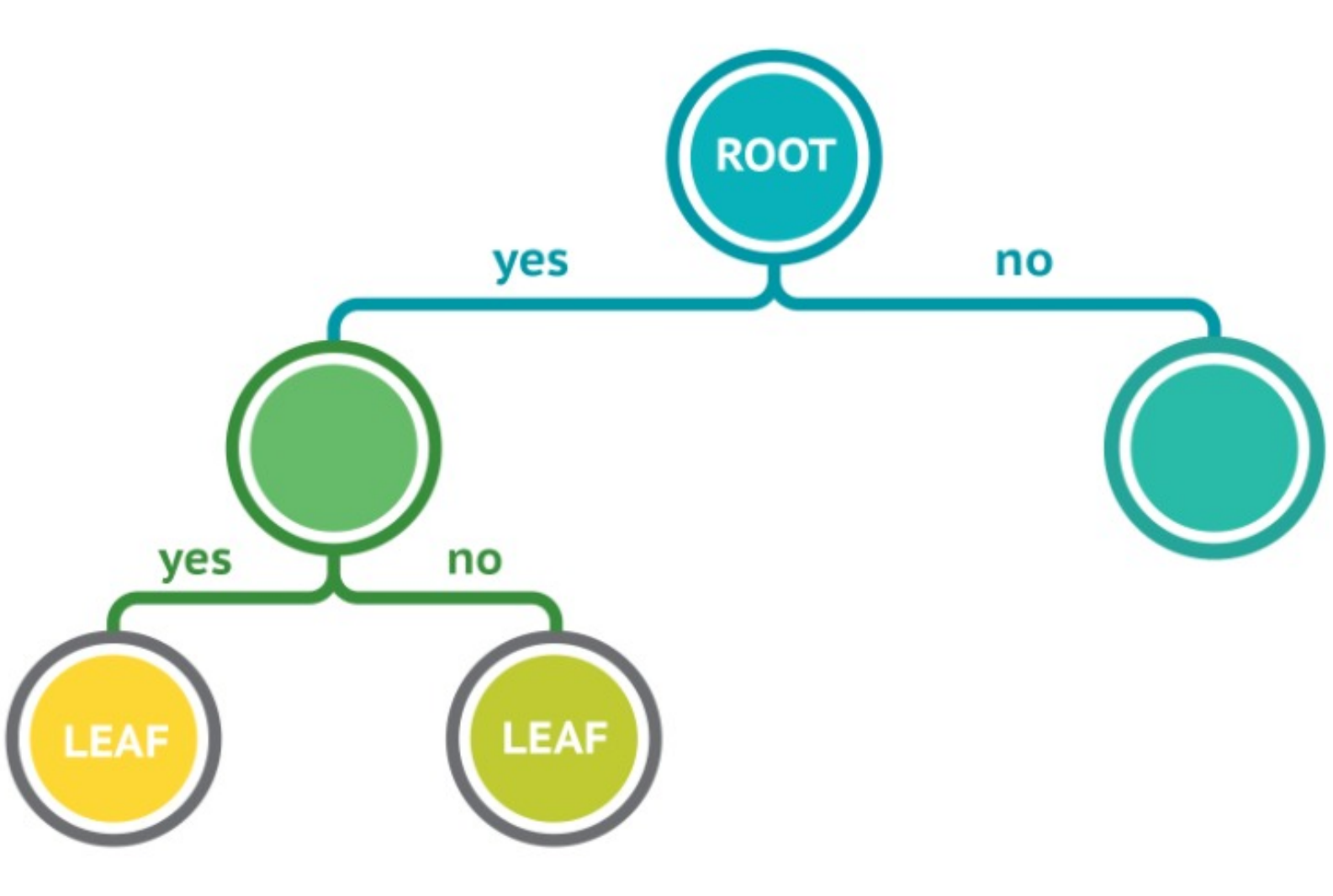


什麼是決策樹？

決策樹(Decision Tree) 為機器學習中 Supervised Learning 底下的演算法，是一種過程直覺單純，且執行效率較高的模型，可以用來處理 classification 與 regression 的資料類型的問題。

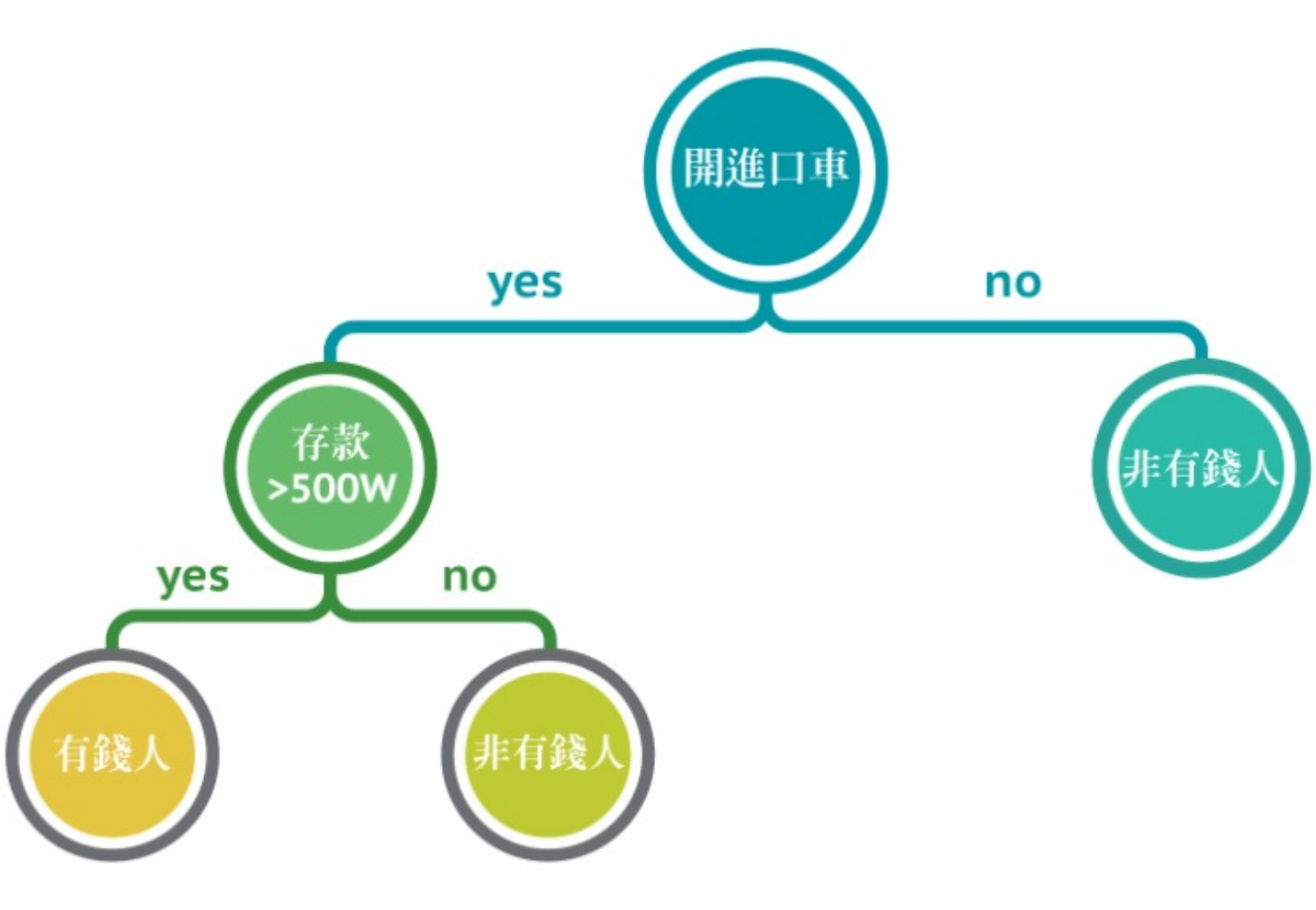


相較於其他分類模型(Logistic Regression, SVM) 的分類依據，決策樹(Decision Tree)的每個決策階段都相當明確清楚(Yes or No)。可以透過每個決策階段了解模型的判別依據並了解資料中的重要參數。



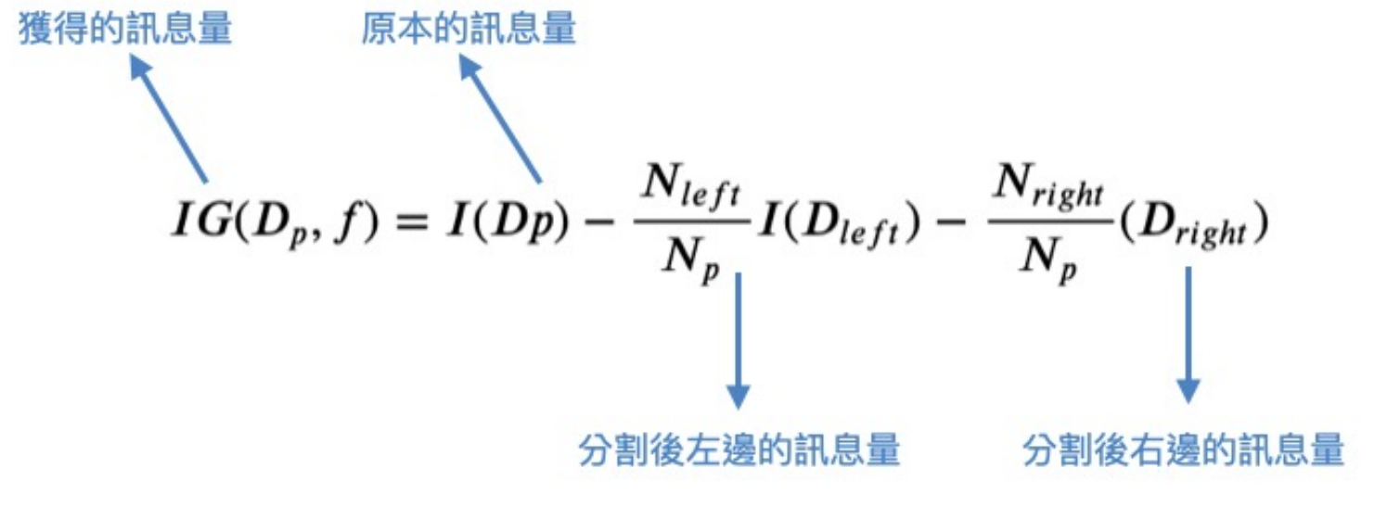
如何進行決策？

- 在了解決策樹架構後，決策樹是如何決定像是以下問題的決策？
- 分成幾個節點？
- 那一個節點適合作為啟始根部？
- 節點的判斷依據及數值的認定為何？
  (如以下分類是否為有錢人的決策樹)

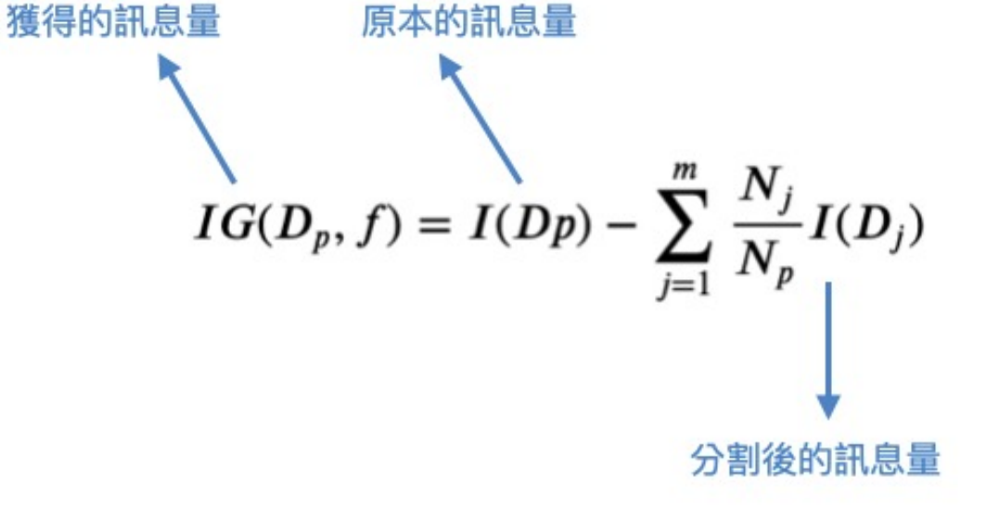


決策樹是依據何種決定選擇了開進口車與否，以及存款是否> 500W，來作為分類依據？

決策樹會透過訓練資料，從最上方的根節點開始找出規則將資料依據特徵分割到節點兩側。分割時的原則是將較高同性質的資料放置於相同側以得到最大的訊息增益(Information Gain, IG)。

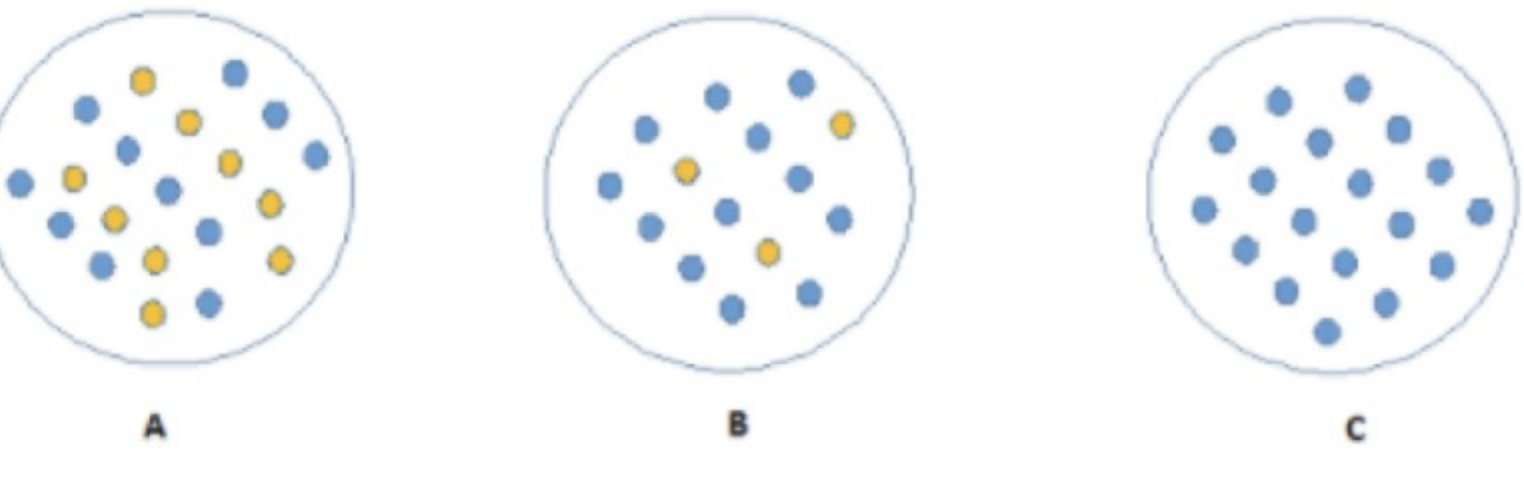


$$IG(D_p, f) = I(Dp) - \frac{N_{left}}{N_p} I(D_{left}) - \frac{N_{right}}{N_p} (D_{right})$$



$$IG(D_p, f) = I(Dp) - \sum_{j=1}^{m} \frac{N_j}{N_p} I(D_j)$$

何謂訊息量？



C: 資料皆為同類別，不需而外資訊即可描述
B: 需要少量訊息量來描述其中不同類別的三個資料點
A: 需要較多的訊息量來描述資料(因為資料較雜亂)

資料來源：決策樹 Decision trees

衡量訊息增益

常見資訊量有兩種：熵(Entropy) 與 Gini 不純度（Gini Impurity)

熵：

$$Entropy = -\sum_{i=1}^{c} p(i) log_2 p(i)$$

Gini：

$$Gini = \sum_{i=1}^{c} p(i)(1 - p(i)) = 1 - \sum_{i=1}^{c} p(i)^2$$

熵(Entropy) 與 Gini 不純度：

- 都在衡量一個序列中的混亂程度，值越高越混亂
- 數值都在 0 ~ 1之間，0 代表序列純度高，皆為同樣的值(類別)

試著考量下列兩種情況的熵(Entropy) 與 Gini 不純度值

皆為藍球(相同類別)

熵： **Entropy = -1 * log 1 = 0**
Gini： **Gini = 1 - 1^2 = 0**

藍球與黃球各半(相同類別)

熵： **Entropy = -0.5*log0.5 - 0.5*log0.5 = 1**
Gini： **Gini = 1 - 0.5^2 - 0.5^2= 0.5**

訊息增益

假設我們為決策樹，根據下列分割計算訊息增益(Entropy)

熵：

$$Entropy = -\sum_{i=1}^{c} p(i) log_2 p(i)$$

$$I_H(D_p) = -(0.5 * log_2(0.5) + 0.5 * log_2(0.5)) = 1$$
$$I_H(D_{left}) = -(\frac{3}{4} log_2(\frac{3}{4}) + \frac{1}{4} log_2(\frac{1}{4})) = 0.81$$
$$I_H(D_{right}) = -(\frac{1}{4} log_2(\frac{1}{4}) + \frac{3}{4} log_2(\frac{3}{4})) = 0.81$$
$$IG_H = 1 - \frac{2}{4} * 0.81 - \frac{2}{4} * 0.81 = 0.19$$

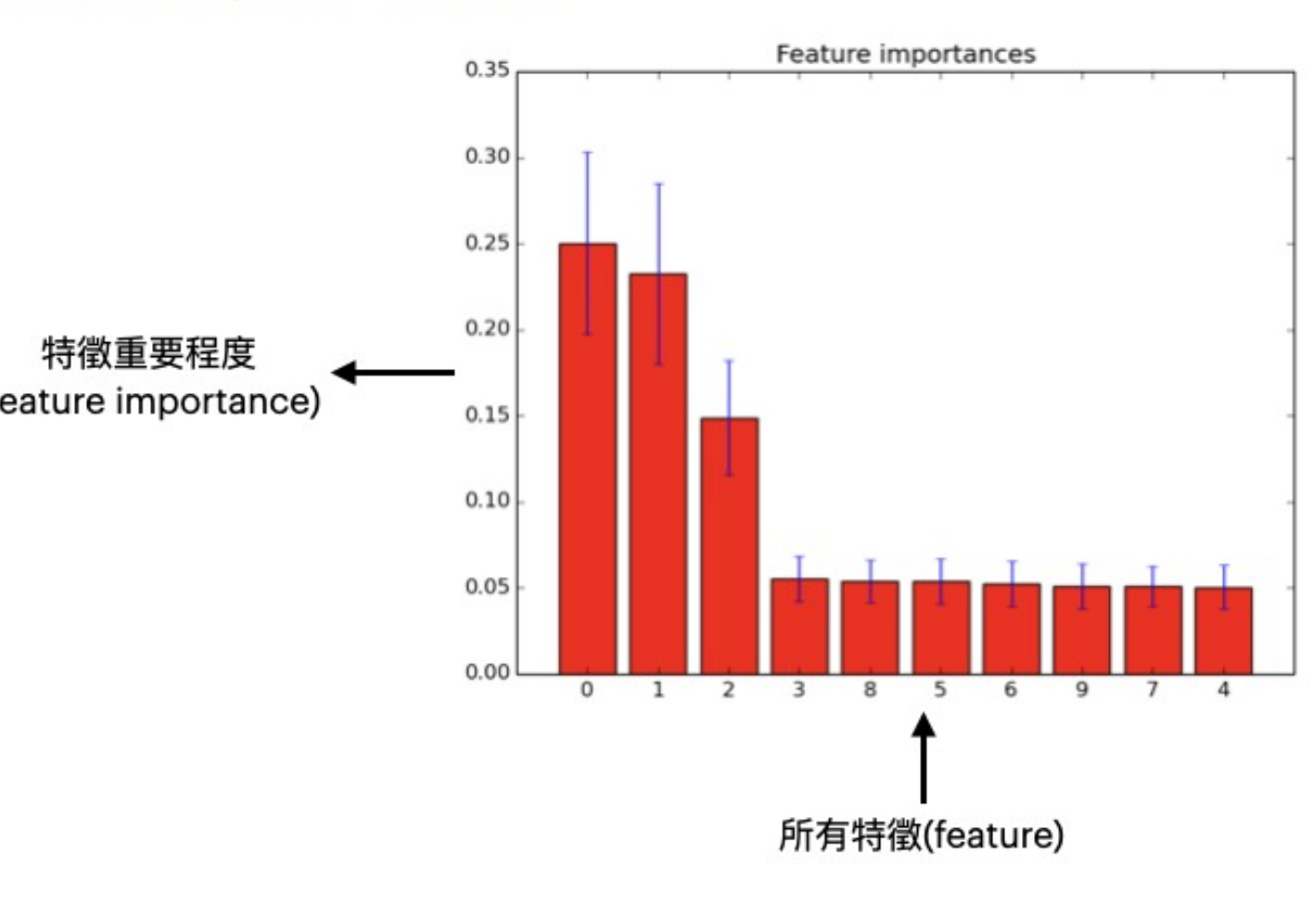
Gini：

$$Gini = \sum_{i=1}^{c} p(i)(1 - p(i)) = 1 - \sum_{i=1}^{c} p(i)^2$$

$$I_G(D_p) = 1 - (0.5^2 + 0.5^2) = 0.5$$
$$I_G(D_{left}) = 1 - ((\frac{3}{4})^2 + (\frac{1}{4})^2) = 0.375$$
$$I_G(D_{right}) = 1 - ((\frac{1}{4})^2 + (\frac{3}{4})^2) = 0.375$$
$$I_G = 0.5 - \frac{2}{4} * 0.375 - \frac{2}{4} * 0.375 = 0.125$$

決策樹 Feature Importance

- 透過建構決策樹，可以利用 feature 被用來切分的次數來得知哪些特徵(feature)是相對有用的，這樣就可以透過 feature importance 來排序特徵的重要性以及要選取使用的特徵
- 所有的 feature importance 的總和為 1



決策樹(Decision Tree)

了解決策樹的運作後，來看看決策樹的特性與優缺點：

優點：

- 算法簡易，容易理解與解釋
- 適合處理有缺失值屬性的樣本
- 能處理數值型與類別型的資料

缺點：

- 容易發生過擬合(於下章節詳細介紹)
- 為考慮數據間的相關聯性

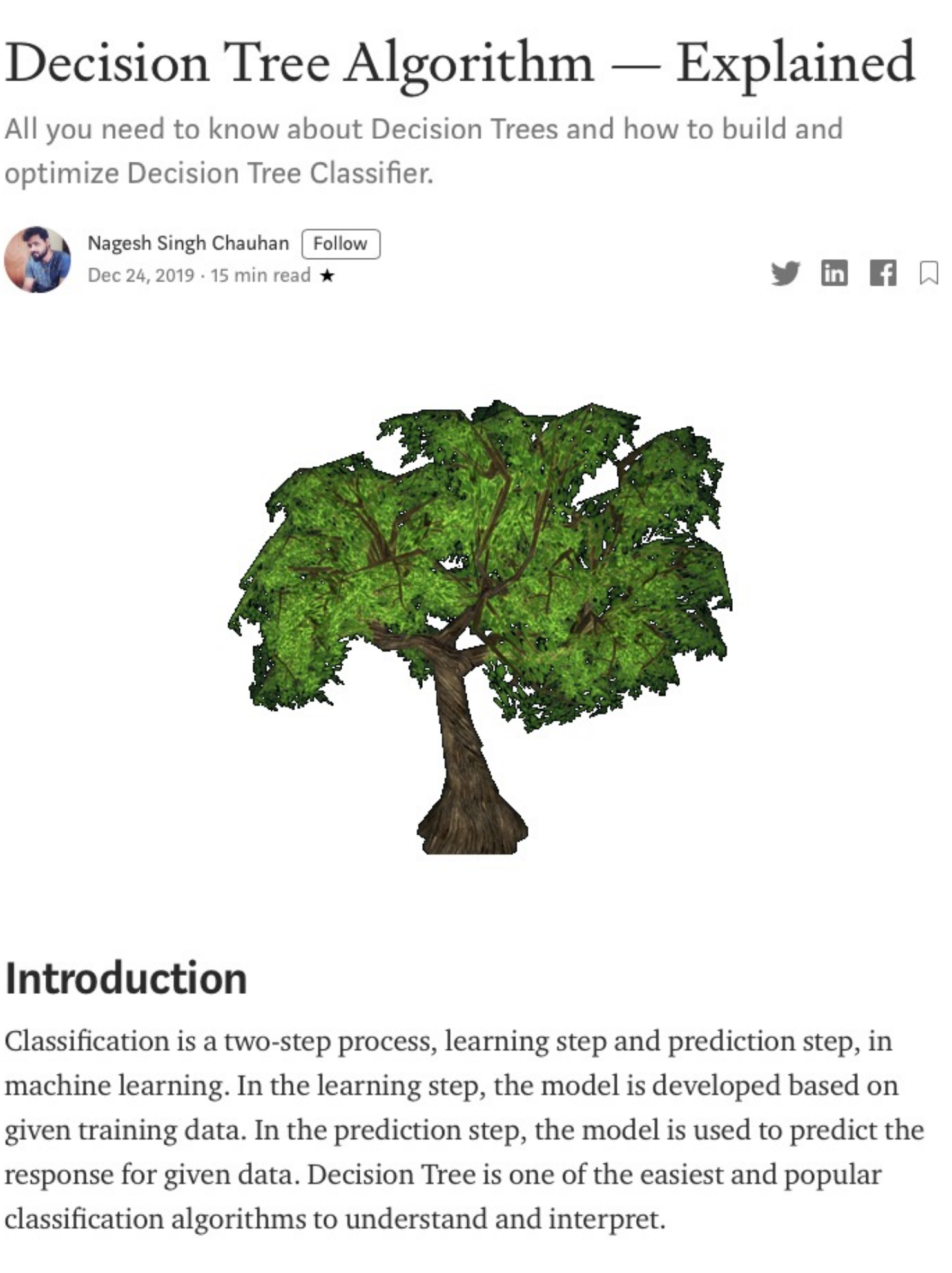
知識點回顧

在這章節我們學習到了

- 了解何謂決策樹模型與其運作原理
- 了解決策樹 Feature Importance 指標

延伸閱讀

網站：Decision Tree Algorithm
此文章對決策樹的演算法有詳細得介紹



Decision Tree Algorithm — Explained

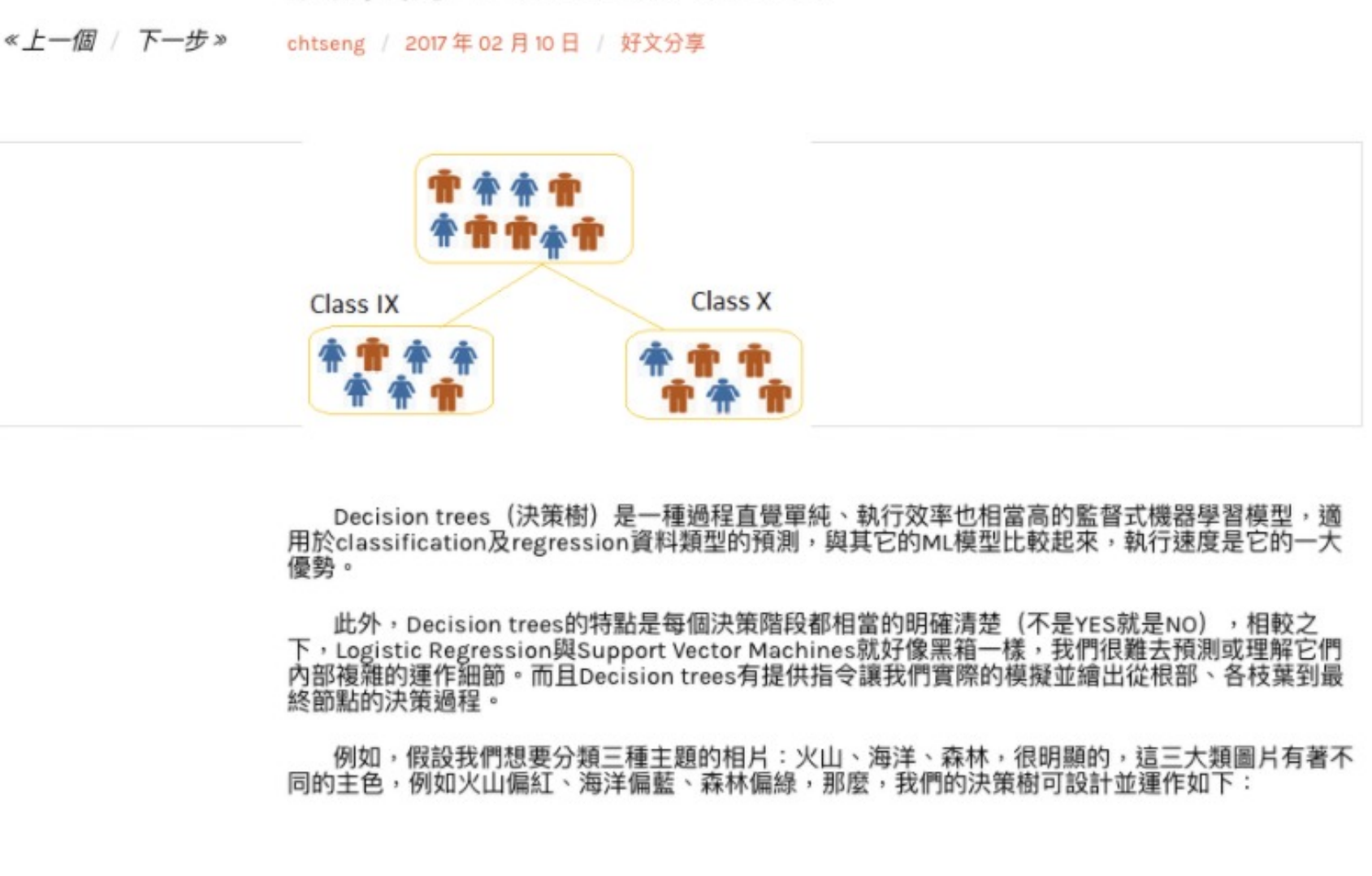
All you need to know about Decision Trees and how to build and optimize Decision Tree Classifier.

Nagesh Singh Chauhan Follow
Dec 24, 2019 · 15 min read ★

Introduction

Classification is a two-step process, learning step and prediction step, in machine learning. In the learning step, the model is developed based on given training data. In the prediction step, the model is used to predict the response for given data. Decision Tree is one of the easiest and popular classification algorithms to understand and interpret.

網站：訊息增益-1



決策樹 Decision trees

«上一個 / 下一步»

chtseng / 2017 年 02 月 10 日 / 好文分享

Decision trees（決策樹）是一種過程直覺單純、執行效率也相當高的監督式機器學習模型，適用於classification及regression資料類型的預測，與其它的ML模型比較起來，執行速度是它的一大優勢。

此外，Decision trees的特點是每個決策階段都相當的明確清楚（不是YES就是NO），相較之下，Logistic Regression與Support Vector Machines就好像黑箱一樣，我們很難去預測或理解它們內部複雜的運作細節。而且Decision trees有提供指令讓我們實際的模擬並繪出從根部、各枝葉到最終節點的決策過程。

例如，假設我們想要分類三種主題的相片：火山、海洋、森林，很明顯的，這三大類圖片有著不同的主色，例如火山偏紅、海洋偏藍、森林偏綠，那麼，我們的決策樹可設計並運作如下：

網站：訊息增益-2

[資料分析&機器學習] 第3.5講：決策樹(Decision Tree)以及隨機森林(Random Forest)介紹

Yeh James Follow
Nov 5, 2017 · 7 min read

在前面的章節我們說明了如何使用Perceptron, Logistic Regression, SVM在平面中用一條線將資料分為兩類，並且Logistic Regression以及 SVM都可以知道這筆資料是A類還是B類的機率，更強大的SVM還可以透過將平面資料投影到空間中來做到非線性分類。前面提到這些知名的模型(Model)都有一個小缺點，想像一下你是一個披薩公司（像是必X客、達X樂）的資料科學家，你成功建立一個模型能預測披薩是美味的披薩還是難吃的披薩，假設使用在烤披薩的中間過程中的兩個量測數值：溫度以及濕度來建立的模型為：

*模型的決策邊界： -100 + 6*溫度 + 3*濕度 = 0*

-100 + 6*溫度 + 3*濕度 > 0 預測是一個美味的披薩

-100 + 6*溫度 + 3*濕度 < 0 預測是一個難吃的披薩

以上兩篇文章對訊息增益的計算有詳細的介紹，有興趣的讀者可以就此部分閱讀

網站：特徵重要性(Feature Importance)
文中對 Feature Importance 有詳細的計算與介紹

scikit-learn決策樹算法中特徵重要性的計算

DKY10 2018-12-05 21:42:12 8419 收藏 10

sklearn.tree.DecisionTreeClassifier類中的feature_importances_屬性返回的是特徵的重要性，feature_importances_越高代表特徵越重要，scikit-learn官方文檔[1]中的解釋如下：

> The importance of a feature is computed as the (normalized) total reduction of the criterion brought by that feature. It is also known as the Gini importance.

這是說，樹的構建中，每個特徵我們都會計算一個它的criterion（基尼指數[2]或者信息熵[3]），特徵重要性就是這個criterion減少量的歸一化值。在scikit-learn中的默認實現也就是我們常說的Gini importance。

stackover上的帖子[4]中答主Seljuk Gülcan指出如果每個特徵只被使用一次，那麼feature_importances_應當就是這個Gini importance：

```
N_t / N * (impurity - N_t_R / N_t * right_impurity
                    - N_t_L / N_t * left_impurity)
```

其中，N是樣本的總數，N_t是當前節點的樣本數目，N_t_L是結點左孩子的樣本數目，N_t_R是結點右孩子的樣本數目。impurity直譯為不純度（基尼指數或信息熵），這裡的實現的是基尼指數。
舉個例子，下面的決策樹

下一步：完成作業