資料分析工程師該具備的技術與知識

資料分析工程師至少該具備下列四種技術。

表 1.1 資料分析工程師該具備的技術

必要的技術	介紹	於本書對應的章節
收集與加工資料的這類處理	從資料庫、檔案取得資料，再視情況加工的技術	第 4 章第 1 節 Numpy 第 4 章第 2 節 pandas
讓資料視覺化	將資料畫成圖表或其他視覺設計，以便了解資料特性的技術	第 4 章第 3 節 Matplotlib 第 4 章第 2 節 pandas 的部分內容
程式設計	Python 這類程式設計技術	第 2 章 Python
基礎建設	建置環境、伺服器技術、資料架構這類技術	第 2 章 Python 的部分內容

下列是資料分析工程師應該具備的附加技術，總共有三項。

表 1.2 資料分析工程師該具備的附加技術

必要的技術	介紹	於本書對應的章節
機器學習	了解機器學習流程，執行機器學習的技術。比起演算法的相關知識，更需要知道執行機器學習的方法	第 4 章第 4 節 scikit-learn
數學	高中到大學初階數學的數學知識	第 3 章 數學
專業知識（產業知識）	對資料分析領域的了解	—

1.1.5 資料處理（前置處理）的重要性

在資料分析的世界裡，**資料處理**是非常重要的一環。

機器學習的世界裡，資料處理大概佔了八成至九成的部分，而被稱為**前置處理**的資料處理指的是收集資料、加工資料、合併資料與呈現資料這些部分，而這些部分會於分析資料之際反覆操作。如果資料不足就必須尋找其他的資料來源，有些機器學習還必須先標準化資料。

1.2 機器學習的定位與流程

接下來要解說的是，在資料分析領域備受注目的機器學習於目前的定位與流程。

1.2.1 何謂機器學習

機器學習是根據機器學習演算法從大量的資料找出資料的特性，再預測結果的程式區塊，而這個區塊又稱為**模型**。建立以機器學習預測結果的模型之後，即可分類資料或是預測數值。要建立這個模型，必須輸入資料，也必須具有處理資料的演算法。

根據資料與演算法依序更新內部的參數，一步一步完成機器學習的模型。之後可利用這個模型以及未知的資料預測數值，或是分類資料，藉此了解輸入資料的特性。

1.2.2 機器學習之外的選項

也有跳過機器學習，直接分類資料或預測數值的方法。

第一種方法稱為建構規則模式。

這種方法就是程式語言的條件分歧寫成的 if 陳述式。

以預測門市的單日業績而言，大概就是當明天為「星期天」，而且是「晴天」，就預測業績達到五萬元，而後天是「星期一」又是「雨天」所以業績只達三萬元。這個範例利只利用「星期」與「天氣」這兩個變數預測。由於參數只有兩個，所以很容易替之前的資料建構規則，可是當參數一多，就很難寫出顧全大局的規則。

第二種方法則是**統計**。

這種方法是根據資料算出統計數值，再依照這些數值預測。

1.2.4 機器學習的處理步驟

接下來要以監督式學習為例，依序介紹機器學習的處理步驟。

依照目的分割處理流程，可分割成以下八個項目：

- 收集資料
- 加工資料
- 具體呈現資料
- 選擇演算法
- 學習程序
- 評估精確度
- 實驗應用
- 運用結果

這些流程的順序可參考下列的流程圖，同時也附註了主要使用的工具與技術。

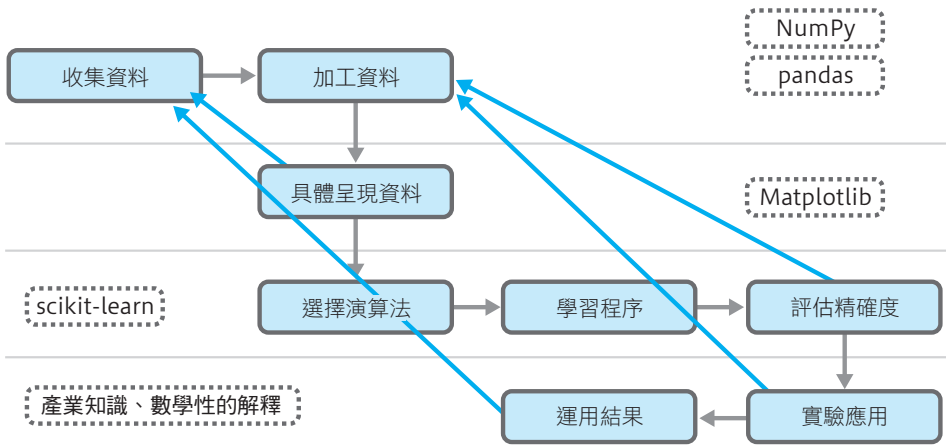

圖 1.1 機器學習的處理流程

2.3 Jupyter Notebook

這節要介紹互動型程式執行環境 Jupyter Notebook。主要介紹 Jupyter Notebook 的安裝方法、基本的使用方法與方便的使用方法，最後還要建立本書使用的資料分析 Python 執行環境。

2.3.1 何謂 Jupyter Notebook

Jupyter Notebook 是資料分析、視覺化資料、機器學習常用的開放原始碼工具，主要是以網路應用程式的方式使用，可在瀏覽器的環境下執行各種程式、預覽執行結果以及建立文件。

Jupyter Notebook 原本的名字是 IPython Notebook，主要是為了在網路瀏覽器執行前一節介紹的 IPython 所開發的工具，但隨著用途越來越廣，除了支援 Python，也開始支援 Julia、R 語言或是其他程式設計語言，後來就改名為 Jupyter Notebook，而 Jupyter 這個名稱也源自 Julia、Python 與 R。

Jupyter Notebook 常於資料分析、機器學習這些領域使用的理由在於可將程式碼（例如 Python）、程式碼執行結果與文件（Markdown 語法）統整於單一 Notebook 文件，而且執行結果不只是單純的字串，還能以 pandas 的 DataFrame 這種方便瀏覽的表格格式顯示，也能利用 Matplotlib 這類視覺呈現工具製作成圖表。

2.3.2 安裝

接下來要安裝包含 Jupyter Notebook 的套件。安裝方法與其他的第三方套件一樣，都是先利用 venv 建立名為 pydataenv 的虛擬環境，再利用 pip 命令安裝。

```
$ python3 -m venv pydataenv
$ source pydataenv/bin/activate
(pydataenv) $ pip install jupyter==1.0.0
```

2.3.3 基本的使用方法

Jupyter Notebook 安裝完畢之後，為大家說明基本的使用方法。第一步，從終端機啟動 Jupyter Notebook。

第一次啟動 Jupyter Notebook 時，必須透過網頁瀏覽器驗證 Token。複製於結尾顯示的 http://localhost:8888/?token=39b39ad…的部分，再於瀏覽器的網址列輸入，就會驗證 Token，並於網頁瀏覽器顯示 Jupyter Notebook 的 Home 畫面。第二次啟動就會自動執行 jupyter notebook 命令以及在網頁瀏覽器顯示 Home 畫面（圖 2.2）。

```
(pydataenv) $ jupyter notebook
( 省略 )
[C 17:22:33.998 NotebookApp]

    Copy/paste this URL into your browser when you ➡
connect for the first time,
    to login with a token:
        http://localhost:8888/?token=39b39ad37085efb995 ➡
da95a15fad047ac32e( 以下省略 )
```

Home 畫面會顯示檔案一覽表、結束 Jupyter Notebook 的 Quit 按鈕以及各種選單。

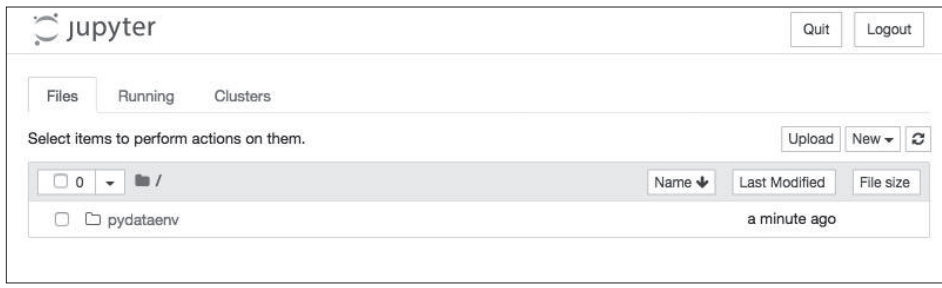

圖 2.2 Jupyter Notebook 的 Home 畫面

從 Home 畫面的 New 選單點選 Python 3（圖 2.3），就能新增撰寫 Python 程式碼的畫面。這個畫面稱為 Notebook，程式碼或圖表會以 Notebook 檔案格式（副檔名為 .ipynb）儲存。此外，若安裝其他種類的程式設計語言，也能建立 Julia 或 R 語言的 Notebook。

圖 2.3 新增 Notebook

建立 Notebook 之後，會顯示 圖 2.4 的畫面。Notebook 可另外加上標題，該標題也將當成檔案名稱使用。預設的標題為 Untitled，點選標題，即可自行命名。

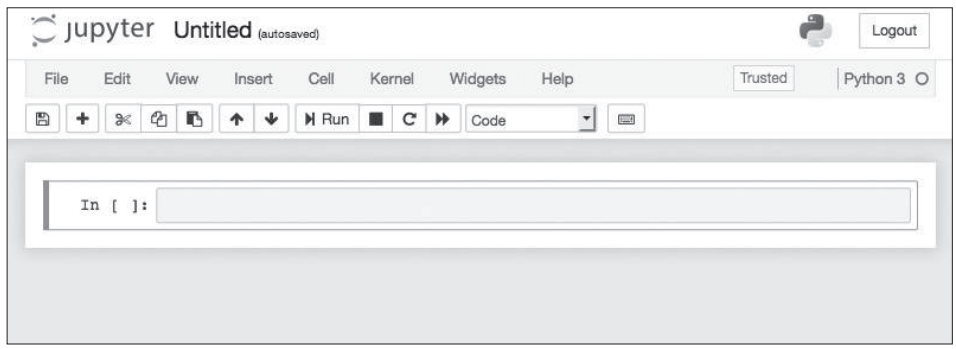

圖 2.4 Untitled 的 Notebook

Notebook 是於區塊（Cell）這個範圍撰寫 Python 程式碼，按下「Shift」+「Enter」鍵即可執行該程式碼。在以 In 為字首的區塊撰寫程式之後，會在以 Out 為字首的區塊輸出執行結果。

下列的 Notebook 要執行幾個前一節所介紹的 Python 程式。最上方的區塊是 Markdown 格式的區塊，能將說明內容、程式碼與程式碼執行結果統整於單一 Notebook 是 Jupyter Notebook 的優點。此外，於區塊撰寫 Python 程式碼時，會以 IPython 為雛型，所以可利用前一節說明的 TAB 鍵提示後續的程式碼，也能使用魔法命令與 Shell 命令。

3.1 閱讀公式所需的基礎知識

本節要介紹常用的數學符號，如果原本就不擅長數學，不妨就將這些數學符號當成普通的符號或簡寫，放鬆心情地讀完本節即可。

3.1.1 公式與符號

希臘字母

數學符號或是公式常出現非英文字母的希臘字母，若不懂這些希臘字母，恐怕有時會產生困擾，所以先讓我們確認字母的形狀與讀法，之後若遇到這些字母，也可隨時翻回 表 3.1，重新確認一次字形與讀音。

表 3.1 希臘字母

大寫	小寫	讀音	大寫	小寫	讀音
A	α	Alpha	N	ν	Nu
B	β	Beta	Ξ	ξ	Xi
Γ	γ	Gamma	O	o	Omicron
Δ	δ	Delta	Π	π	Pi
E	ϵ	Epsilon	P	ρ	Rho
Z	ζ	Zeta	Σ	σ	Sigma
H	η	Eta	T	τ	Tau
Θ	θ	Theta	Υ	υ	Upsilon
I	ι	Iota	Φ	ϕ (φ)	Phi
K	κ	Kappa	X	χ	Chi
Λ	λ	Lambda	Ψ	ψ	Psi
M	μ	Mu	Ω	ω	Omega

集合

數學很適合將事物轉換成抽象的概念，所以有時能用來呈現沒有順序，只是某些數字的群體，而這種數字群體就稱為**集合**，剛好與 Python 的集合類型（set 類型）相同。某個元素 x 屬於集合 S 時，會以下列的公式呈現。

3.1.2 數學符號

Python 與許多程式設計語言都有 for 陳述式，而這個陳述式可用於撰寫重複處理，至於數學也有表現重複處理的符號。

連續加總

從x_1加總至x_n的數字可利用下列的公式代表。

$$\sum_{i=1}^{n} x_i \tag{3.8}$$

公式（3.8）使用的符號是希臘字母大寫的 sigma。迴圈的第一個數字寫在下面，最後一個數字寫在上面，有時候會省略這些數字。此外，公式也可呈現加總至無限大這類無法以程式加總的內容。

$$\sum_{n=1}^{\infty} \frac{1}{4^n} = \frac{1}{3} \tag{3.9}$$

連續加乘

一如加法，也有連乘的符號。從x_1連乘至x_n的運算可透過下列的公式表現。

$$\prod_{i=1}^{n} x_i \tag{3.10}$$

公式（3.10）使用的符號是希臘字母大寫的 Π。迴圈的第一個數字與最後一個數字與 Σ 相同。

特殊常數

假設眼前有一個直徑為 1 的圓形，其圓周會有多長呢？答案就是我們熟知的**圓周率**，也就是 3.1415……的無限小數，數學則以 π 這個符號代表圓周率。

另一個常用的常數為 e 代表的**納皮爾數（自然對數底）**。這個常數或許不比圓周率出名，但在處理函數的微分與積分的分析學領域之中，卻是舉足輕重的常數。e 可利用公式（3.11）定義。

$$e = \sum_{n=0}^{\infty} \frac{1}{n!} \tag{3.11}$$

3.4 機率與統計

要想統整充斥全世界的資料或是根據手邊的資料降低未來的不確定性，機率與統計的理論的確幫了很大的忙。

3.4.1 統計的基礎

現代已是資訊爆炸，資料多到不知該如何篩選的時代，所以將資料整理成方便閱讀的格式，或是計算平均值、觀察資料的特性，都是分析資料之際，不可或缺的過程。接下來要利用樣本資料介紹統計學的基本理論。

代表值

表 3.2 是從日本總務省統計局[※5]下載的家庭全年納豆平均購買金額資料，這份資料的統計期間為 2015 年到 2017 年，總共是三年份的資料。日本都道府縣的相關資料都可於這個網站下載。

表 3.2 家庭全年納豆平均購買金額（根據附註的網址製表）

都道府縣	金額	順位（升冪）
和歌山縣	1795	1
沖繩縣	2782	8
東京都	4009	29
神奈川縣	4153	31
福島縣	6092	47

這些資料若能進一步加工，就能算出不同的統計數據，想必大家常在報紙或網站的報導看到這類能一窺資料全貌的代表值吧？讓我們先整理常使用的代表值。

※5 出處：家計調查（二人以上家庭）品目別都道府縣廳所在市及政命令指定都市（*）排行（平成 27（2015 年）～29 年（2017 年）平均）。從網站下載的資料已根據都道府縣整理完畢。

URL http://www.stat.go.jp/data/kakei/5.html

決策樹

決策樹（decision tree）是如下圖所示，逐一建立分割資料的規則，再進行分類的演算法，也是具代表性的機器學習手法。由於模型的內容很容易理解，所以常於實務應用。

	使用次數	使用間隔	距離最後一次使用的天數	…	是否背離服務
A使用者	5次	10.1天	3天	…	會
B使用者	2次	1.5天	1天	…	不會
…	…	…	…	…	…

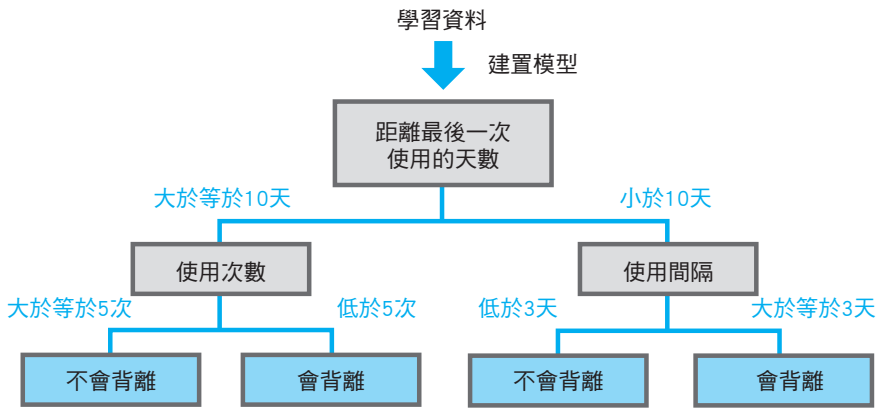

圖 4.56 決策樹

在說明決策樹之前，先說明之後會用到的術語。這些術語雖然不止用於決策樹，不過這種被稱為「樹」的資料結構是以頂點的「節點」以及與其連結的「有向邊」組成，以開頭的顧客是否繼續使用服務的範例而言，「距離最後一次使用的天數」、「使用次數」、「使用間隔」為節點，「距離最後一次使用的天數為大於小於 10 天」、「使用次數低於 5 次」的部分就是「有向邊」。若將樹木構造看成家譜，節點有時會有「子節點」，例如「距離最後一次使用的天數」的子節點就是「使用次數」與「使用間隔」這兩個節點，反向來看，節點也會有「父節點」，例如以「使用次數」為起點，「距離最後一次使用的天數」為父節點。

Out

```
True
```

執行上述的程式後，就會輸出繪有決策樹的 tree.png 圖檔。

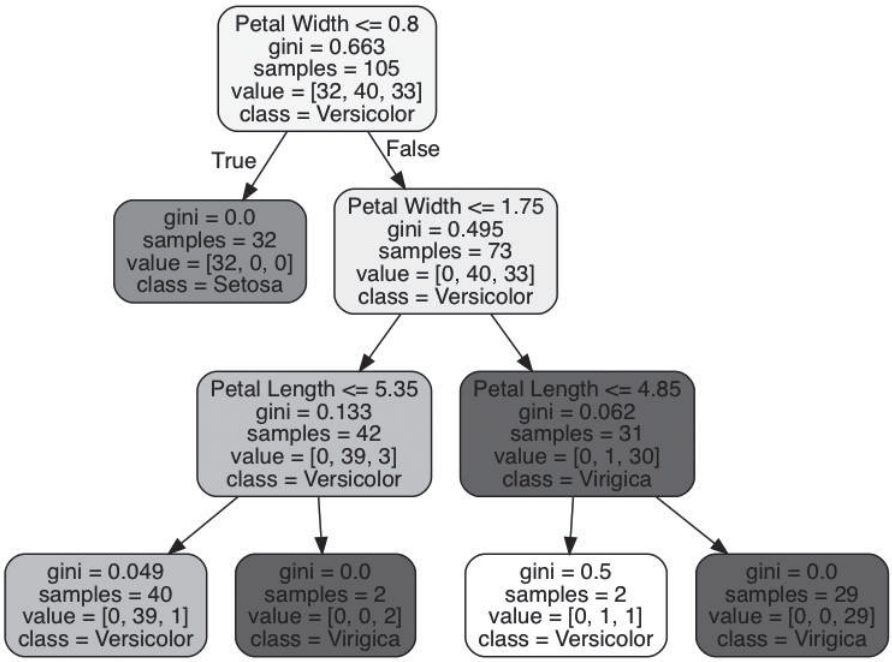

圖 4.57　具體繪製決策樹的構造

接著以一開始的節點分割為例，簡單地說明閱讀決策樹的方法。

- 最上方的節點會以 Petal Width 小於等於 0.8 或大於 0.8 的標準分割資料。資料分割之前，資料的類別數為「value = [32,40,33]」，這三個數字分別代表 Setosa、Versicolor、Virginica，而這三個英文單字代表是鳶尾花的種類（詳情請參考 P.225）。在多數決裡，Versicolor 最多，所以「class = Versicolor」的部分就是多數決的結果。此外，這個節點的基尼不純度指標為「gini = 0.663」。
- 當 Petal Width 小於等於 0.8 時，就往左下角的子節點移動，此時的類別數為 Setosa=32，其餘的 Versicolor 與 Virginica 為 0。由於只有類別 Setosa 的資料，所以基尼不純度指標為 0.0。

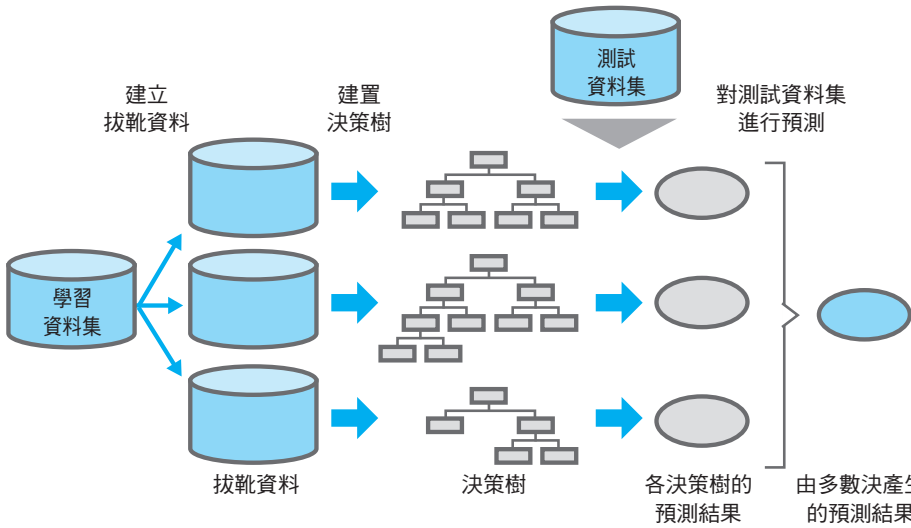

圖 4.58 隨機森林

要以 scikit-learn 執行隨機森林演算法，可使用 ensemble 模組的 RandomForestClassifier 類別。與之前說明的演算法一樣，都是利用 fit 方法學習，再利用 predict 方法預測未知的資料。建立 RandomForestClassifier 類別的實體時，可在參數 n_estimators 指定決策樹的個數。下列的範例指定產生 100 個決策樹。

In

```
from sklearn.ensemble import RandomForestClassifier
# 建立隨機森林的實體
forest = RandomForestClassifier(n_estimators=100, random_state=123)
# 學習
forest.fit(X_train, y_train)
# 預測
y_pred = forest.predict(X_test)
y_pred
```