

關於本書



本書教給你兩件事：機器學習模型以及如何使用它們。機器學習模型有不同的類型，其中一些會回傳確定的答案，例如是或否，而另一些則以機率的形式回傳答案。它們之中有些使用方程式，另一些使用 if 語句，它們的一個共同點是都會回傳一個答案或一個預測。機器學習的分支，包括回傳預測的模型，被恰當地命名為預測性機器學習，這就是我們在本書中專注的機器學習類型。

本書架構

章節類別

本書有兩種類型的章節。其中大部分（第 3、5、6、8、9、10、11 和 12 章）各包含一種機器學習模型的類型。每章中對應的模型都有詳細的研究，包括範例、公式、程式碼和供你解決的練習。其他章節（第 4、7 和 13 章）包含了用於訓練、評估和改進機器學習模型的有用技術。特別是，第 13 章包含一個真實資料集上的端到端範例，在這個範例中，你將能夠應用你在前面幾章中獲得的所有知識。

建議的學習順序

你有兩種使用這本書的方式。我推薦的是逐章線性地閱讀，因為你會發現，學習模型和學習訓練模型的技術之間的交替是很有意義的。然而，另一種學習途徑是首先學習所有的模型（第 3、5、6、8、9、10、11 和 12 章），然後學習訓練它們的技術（第 4、7 和 13 章）。當然，由於我們都有不同的學習方式，你可以創造你自己的學習路徑！

附錄

本書有三個附錄。附錄 A 包含每章練習的解答。附錄 B 包含一些有用的正式數學推導，但比本書的其餘部分更具技術性。附錄 C 包含我推薦的參考資料和資源清單，如果你想進一步了解的話。

要求與學習目標

本書為你提供了一個可靠的預測性機器學習框架。要充分利用這本書，你應該有一個視覺思維，並對初級數學有良好的理解，例如線圖、方程式和基本機率。如果你知道如何寫程式，特別是 Python，這將很有幫助（儘管不是強制性的），因為在整本書中，你有機會在真實的資料集中實作和應用幾個模型。閱讀本書後，你將能夠做到以下幾點：

- 描述預測性機器學習中最重要的模型及其工作原理，包括線性迴歸和邏輯迴歸、單純貝氏分類、決策樹、神經網路、支援向量機和集成方法。
- 識別它們的優點和缺點，以及它們使用的參數。
- 識別這些模型在現實世界中的使用方式，並制訂潛在的方法，將機器學習應用於你想解決的任何特定問題。
- 學習如何最佳化這些模型，比較並改進它們，以建立我們可以建立的最佳機器學習模型。
- 手動或使用現有的套裝軟體來編寫模型，並使用它們對真實的資料集進行預測。

如果你有一個特定的資料集或問題，我邀請你思考如何將你在本書中學到的東西應用於其中，並將其作為一個起點來實現和實驗自己的模型。

我非常高興能和你一起開始這段旅程，希望你也一樣興奮！

其他學習資源

這本書自成一體，這意味著除了前面所述的要求之外，我們需要的每個概念都在書中介紹了。然而，本書包含了許多參考資料，如果你想更深入理解這些概念，或者你想探索更多主題，我建議你查看在附錄 C 和這個連結中的參考資料：

<http://serrano.academy/grokking-machine-learning>。



本章包含

- 什麼是機器學習
- 機器學習很難嗎（劇透：一點也不難）
- 我們可以從本書學到什麼
- 什麼是人工智慧，以及它和機器學習有何不同
- 人們是如何思考的，以及我們如何將這些想法加諸於機器
- 一些現實生活中基本的機器學習範例

USER FRIENDLY by Illiad



Copyright (c) 2000 Illiad <http://www.userfriendly.org>



我超級高興能加入你的學習旅程！

歡迎你來到本書！我超級高興能和你一起參加這趟了解機器學習的旅程。縱觀來看，機器學習是一個電腦用和人類差不多的方式來解決問題、並做出決策的過程。

在本書中，我想告訴你的是，機器學習其實很簡單！你不需要有深厚的數理和程式背景來了解機器學習。的確，你會需要一些基本的數理知識，但主要是要有常識、好的直覺、及一顆好學的心，並且期望將學習到的方法應用於你所熱衷的事物，以及用來改善世上任何事情。

我在寫這本書的過程是很愉快的，因為能加深我對這個主題的了解，我也希望你在閱讀本書並深入了解機器學習的同時也能快樂盡興！

機器學習無所不在！

「機器學習無所不在！」這句話是一天比一天更真實。我很難想到生活中有哪個方面是不能透過機器學習以某種方式來改善的，對於任何重複的或需要查看資料並收集結論的工作，機器學習都可以提供協助。過去幾年來，由於計算能力的進步和資料收集的普遍，機器學習有巨大的進展。舉幾個機器學習應用的例子：推薦系統、圖像辨識、文字處理、自駕車、垃圾郵件辨識、醫療診斷……這清單持續增加中。或許你有一個目標或是有一個想要產生影響的領域（或者你可能已經在實現它了），機器學習很可能可以應用於該領域，這也許就是你閱讀本書的原因。讓我們一起來了解一下吧！

我需要有深厚的數理和程式背景才能了解機器學習嗎？

不需要！機器學習需要的是想像力、創造力、和視覺思維。機器學習是關於從世界上出現的現象中找出模式，並應用這些模式來對未來進行預測。如果你喜歡尋找模式和發現事物之間的關聯性，那麼你就可以進行機器學習。如果我告訴你，我停止抽菸、吃了很多的蔬菜還有開始運動，那麼你預測在未來一年內我的健康會如何呢？或許我的身體健康會有改善。那如果我跟你說，我原本穿紅色毛衣現在改穿綠色毛衣，你預測我的健康在未來一年內會發生什麼變化？（可能會有改變，但不是基於我所提供的資訊）。找出這些關聯性和模式就是機器學習的目的；唯一的不同是，在機器學習中，我們需要將公式和數字加進這些模式之中，讓機器來找出其中的關聯。

雖然你需要具備一些數學和程式語言的知識，但你並不需要成為專家；如果你只是其中一方面的專家，或者兩個方面都是專家，那你肯定會發現你的技能非常有用；但倘若你不是，你仍可以學習機器學習，並在過程中學會數學和程式語言。在本書中，我們會在需要的時候介紹所有的數學概念。而關於程式語言，你想在機器學習中寫多少程式碼完全取決於你。有的人整天都在寫程式，有的人根本不寫；有很多現成的套件、API 和工具可以幫助我們用最少的程式碼來進行機器學習。每一天，這世上的每個人都有更多機會使用機器學習，我很高興你已經在這行列中了！

當公式和程式碼被視為一種語言時，它們是很有趣的

在大部分的機器學習書籍中，演算法是以公式、導數等數學方式來解釋。雖然這些精確的說明方式在實務上的效果很好，但一個單獨的公式可能會更令人困惑。然而，就像樂譜一樣，一個公式可能在混亂的背後隱藏著美妙的旋律。例如，讓我們看一下這個公式： $\sum_{i=1}^4 i$ ，乍看之下很醜，但它表示一個非常簡單的加總，就是 $1 + 2 + 3 + 4$ 。那麼 $\sum_{i=1}^n w_i$ 呢？這只是許多 (n) 個數字的總和。但是當我想到許多數字的總和時，我寧願想像是 $3 + 2 + 4 + 27$ 這樣的東西，而不是 $\sum_{i=1}^n w_i$ 。每當我看到一個公式的時候，我馬上要想像一個小例子，然後腦海中的畫面就會更加清晰了。而當我看到像 $P(A|B)$ 這樣的東西時，會想到什麼呢？那是一個條件機率，所以我會想到一些類似「在一個事件 B 已經發生的情況下，事件 A 發生的機率」的句子。例如，如果 A 代表今天下雨，B 代表生活在 Amazon 雨林中，那麼公式 $P(A|B) = 0.8$ ，表示「假設我們生活在 Amazon 雨林中，今天下雨的機率是 80%」。

如果你真的喜歡公式，別擔心——這本書還是有這些公式，但它們會出現在說明的例子之後。

程式碼也是如此。當我們從遠處看程式碼，它看起來可能很複雜，我們可能很難想像有人能把所有的程式碼放進他們的腦袋中。但是，程式碼其實只是一系列的步驟，通常每個步驟都很簡單。在本書中，我們將會寫程式，但程式會被分解成簡單的步驟，並且每一步都會透過範例和插圖來仔細地說明。在前面幾章中，我們將從頭開始編寫模型，來了解它們是如何運作的；而在後面的章節中，模型會變得更加複雜，我們將使用 Scikit-Learn、Turi Create 或 Keras 等套件，它們能以非常清楚且強大的方式來實現大部分的機器學習演算法。

好的，所以機器學習到底是什麼？

要定義機器學習，首先讓我們定義一個更通用的術語：人工智慧。

什麼是人工智慧？

人工智慧 (*artificial intelligence*, AI) 是一個通用術語，其定義如下：

人工智慧 (artificial intelligence, AI) 電腦用來做決策的所有任務之組合。

在許多情況下，電腦透過模仿人類做決定的方式來做出這些決策；在某些情況下，它們可會模仿進化過程、遺傳過程、或物理過程。但總括來說，當我們看到電腦自行解決問題時，不論是駕駛汽車、尋找兩地點之間的路線、診斷病患還是推薦電影，我們都在關注人工智慧。

什麼是機器學習？

機器學習和人工智慧很相似，而且它們的定義經常被混淆。機器學習 (*machine learning*, ML) 是人工智慧的一部分，我們將其定義如下：

機器學習 (machine learning, ML) 電腦可以根據資料來做決策的所有任務之組合。

這是什麼意思？讓我用圖 1.1 的圖表來說明。

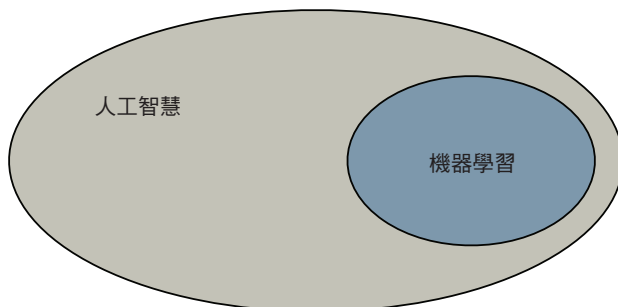


圖 1.1 機器學習是人工智慧的一部分。

讓我們回過頭來看看人類是如何做出決策的。一般而言，我們透過以下兩種方式做出決策：

- 透過使用邏輯和推理
- 透過利用我們的經驗

例如，假設我們正試圖決定購買什麼汽車，我們可以仔細查看汽車的特性，例如價格、油耗和導航，並嘗試找出適合我們預算的最佳組合。這就是使用邏輯和推理。相反地，如果我們問我們所有的朋友他們擁有什麼汽車，以及他們喜歡和不喜歡什麼，我們形成一個資訊清單並使用該清單來決定，那麼我們正在使用經驗（在本例中，是我們朋友的經驗）。

機器學習代表第二種方法：使用我們的經驗做出決策。在電腦術語中，經驗就是資料，因此，在機器學習中，電腦根據資料做出決策。所以，每當我們讓電腦解決問題或僅使用資料做出決定時，我們就是在進行機器學習。通俗地說，我們可以用以下方式描述機器學習：

機器學習是常識，只是由電腦所完成。

從使用任何必要的手段解決問題到僅使用資料來解決問題，這對電腦來說可能是一小步，但對人類來說卻是一大步（圖 1.2）。曾幾何時，如果我們想讓一台電腦執行一項任務，我們必須編寫一個程式，即一整套指令供電腦執行。這個過程適用於簡單的任務，但有些任務對於這個框架來說太複雜了；例如，識別圖像是否包含蘋果的任務，如果我們開始編寫一個電腦程式來開發這個任務，我們很快就會發現它很難。



圖 1.2 機器學習涵蓋了電腦根據資料做出決策的所有任務。就像人類根據以前的經驗做決定一樣，電腦也可以根據以前的資料做出決策。

讓我們退後一步思考以下問題。作為人類，我們是如何了解蘋果的外觀的？我們學習大多數詞彙的方式，不是透過某人向我們解釋它們的意思，而是從不斷重複中學會這些詞彙。我們小時候見過很多東西，大人會告訴我們這些東西是什麼。為了解蘋果是什麼，多年來我們在聽到蘋果這個詞時看到了很多蘋果，直到有一天我們才知道蘋果是什麼。在機器學習中，這就是我們讓電腦去做的事情。我們向電腦展示了許多圖像，並告訴它哪些包含一個蘋果（構成我們的資料）；我們重複這個過程，直到電腦捕捉到構成蘋果的正確模式和屬性。在這個過程結束時，當我們向電腦輸入新圖像時，它可以使用這些模式來確定圖像中是否包含蘋果。當然，我們仍然需要對電腦進行程式設計，使其能夠捕捉到這些模式，為此，我們有幾種技術將在本書中學習。

既然我們已經做到了，那什麼是深度學習呢？

就像機器學習是人工智慧的一部分一樣，深度學習也是機器學習的一部分。在上一段中，我們了解到我們有幾種技術可以讓電腦從資料中學習，其中一種技術表現非常出色，因此它有自己的研究領域，稱為深度學習（DL），我們將其定義如下，如圖 1.3 所示：

深度學習（deep learning，DL） 使用某些稱為神經網路（*neural networks*）之對象的機器學習領域。

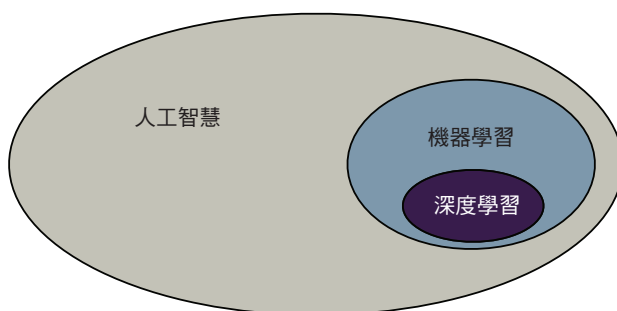


圖 1.3 深度學習是機器學習的一部分。

什麼是神經網路？我們將在第 10 章說明介紹。深度學習可以說是最常用的機器學習類型，因為它運行得非常好。如果我們正在研究任何尖端應用，例如圖像辨識、文字生成、下圍棋或自駕車等，我們很可能正在以某種方式研究深度學習。

換句話說，深度學習是機器學習的一部分，而機器學習又是人工智慧的一部分。如果這本書是關於交通工具，那麼 AI 就是車輛，ML 就是汽車，DL 就是法拉利。

我們如何讓機器根據資料做出決策？ 記得 - 制定 - 預測框架

在上一節中，我們討論了機器學習由一組技術組成，我們使用這些技術讓電腦根據資料做出決策。在本節中，我們將了解以資料為基礎做出決策的含義，以及其中一些技術的工作原理。為此，讓我們再次分析人類根據經驗做出決策的過程。這就是所謂的記得 - 制定 - 預測框架，如圖 1.4 所示。機器學習的目標是教電腦如何以相同的方式思考，遵循相同的框架。

人類是怎麼思考的？

當我們作為人類需要根據我們的經驗做出決定時，我們通常使用以下框架：

1. 我們**記得**（**remember**）過去的類似情況。
2. 我們**制定**（**formulate**）一個一般規則。
3. 我們使用這個規則來**預測**（**predict**）未來可能發生的事情。

舉例說明，如果問題是「今天會下雨嗎？」，猜測的過程如下：

1. 我們**記得**上週大部分時間都在下雨。
2. 我們將其**制定**，在這個地方，大部分時間都在下雨。
3. 我們**預測**今天會下雨。

我們可能是對的，也可能是錯的，但至少我們正試圖根據我們所掌握的資訊做出最準確的預測。



圖 1.4 記得 - 制定 - 預測的框架是我們在本書中使用的主要框架。它包括三個步驟：(1) 我們記得以前的資料；(2) 我們制定一個通則；(3) 我們使用該規則來預測未來。

一些機器學習術語——模型和演算法

在我們深入研究更多說明機器學習中使用技術的案例之前，讓我們定義一些本書中使用的有用術語。我們知道，在機器學習中，我們讓電腦學習如何使用資料解決問題，電腦解決問題的方式是利用資料建構模型。什麼是模型？我們定義一個模型如下：

模型 (model) 一組代表我們的資料並可用於進行預測的規則。

我們可以將模型視為使用一組盡可能接近現有資料的模擬規則來表示現實。在上一節下雨的例子中，模型是我們對現實的表示，這是一個大部分時間都在下雨的世界。這是一個簡單的世界，有一個規則：大部分時間都在下雨。這種表示可能準確，也可能不準確，但根據我們的資料，它是我們所能制定出最準確的現實之表示方法。我們稍後會使用此規則對看不見的資料進行預測。

演算法是我們用來建構模型的過程。在同個下雨的例子中，過程很簡單：我們查看了下雨的天數並發現這是大多數。當然，機器學習演算法可能比這複雜得多，但最終，它們總是由一組步驟組成。我們對演算法的定義如下：

演算法 (algorithm) 是用於解決問題或執行計算的過程或一組步驟。在本書中，演算法的目標是建構模型。

簡而言之，模型是我們用來進行預測的東西，而演算法是我們用來建構模型的東西。這兩個定義很容易混淆並且經常互換，但為了使它們清晰，我們來看幾個案例。

人類所使用的一些模型案例

在本節中，我們關注機器學習的一個常見應用：垃圾郵件偵測。在下方的例子中，我們將檢測垃圾郵件（spam）和正常郵件。非垃圾郵件也稱為正常郵件（ham）。

spam 與 ham *spam* 是垃圾郵件或不想要的電子郵件的常用術語，例如連鎖信、促銷等。該術語來自 1972 年的 Monty Python 草圖，其中餐廳菜單中的每個項目都包含垃圾郵件作為成分。在軟體開發人員中，*ham* 一詞用於指代正常郵件。

案例 1：煩人的電子郵件朋友

在此案例中，我們的朋友 Bob 喜歡向我們發送電子郵件。他寄的很多電子郵件都是以連鎖信形式的垃圾郵件，我們開始對他有點生氣。現在是星期六，我們剛收到 Bob 的電子郵件通知，我們可以不看信就猜出這封電子郵件是 spam 還是 ham 嗎？

為了解決這個問題，我們使用了記得 - 制定 - 預測。首先，讓我們**記得**，比如說，我們從 Bob 那裡收到的最後 10 封電子郵件，這是我們的資料。我們記得其中六個是垃圾郵件，另外四個是正常郵件。根據這些資訊，我們可以**制定**以下模型：

模型 1：Bob 發送給我們的每 10 封電子郵件中有 6 封是垃圾郵件。

這條規則將成為我們的模型。請注意，此規則不必為真；這也可能錯得很離譜，但有鑑於我們的資料，這是我們能想到最好的，所以我們會接受它。在本書的後面部分，我們將學習如何評估模型並在需要時改進它。

現在我們有了規則，我們可以使用它來**預測**電子郵件是否是垃圾郵件。如果 Bob 的 10 封電子郵件中有 6 封是垃圾郵件，那麼我們可以假設這封新郵件有 60% 的可能性是垃圾郵件，40% 的可能性是正常郵件。從這條規則來看，認為電子郵件是垃圾郵件更安全一些。因此，我們預測該電子郵件是垃圾郵件（圖 1.5）。

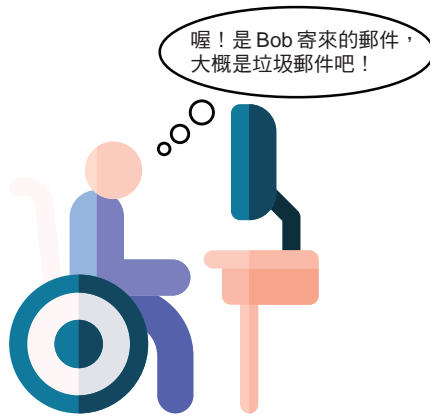


圖 1.5 一個非常簡單的機器學習模型。

同樣地，我們的預測可能有錯。我們可能會打開電子郵件之後才發現它其實是正常郵件，但我們已經盡可能在我們的知識範圍中做出了預測。這就是機器學習的全部意義所在。

你可能會想，我們能做得更好嗎？我們似乎以同樣的方式判斷來自 Bob 的每封電子郵件，但可能有更多資訊可以幫助我們區分垃圾郵件和正常郵件。讓我們嘗試更多地分析電子郵件。舉例來說，讓我們看看 Bob 發送電子郵件的時間，看看我們是否可以從中找到一些線索。

案例 2：一個週期性煩人的電子郵件朋友

讓我們更仔細地看看 Bob 在上個月發給我們的電子郵件，更具體地說，我們將看看他是哪一天寄出來的。以下是有關垃圾郵件或正常郵件的日期和資訊：

- 星期一：正常郵件
- 星期二：正常郵件
- 星期六：垃圾郵件
- 星期日：垃圾郵件
- 星期日：垃圾郵件
- 星期三：正常郵件

- 星期五：正常郵件
- 星期六：垃圾郵件
- 星期二：正常郵件
- 星期四：正常郵件

現在情況不同了。你能看到一個模式嗎？Bob 在週間發送的每封電子郵件都是正常的，而他在週末發送的每封電子郵件則都是垃圾郵件。這是有道理的，也許他在週間向我們發送的都是跟工作相關的郵件，而在週末，他有時間自由自在地發送垃圾郵件。因此，我們可以**制定**一個更有根據的規則或模型，如下所示：

模型 2：Bob 在週間發送的每封電子郵件都是正常的，而他在週末發送的都是垃圾郵件。

現在讓我們看看今天是星期幾。如果是星期天，那麼我們可以非常有把握地**預測**他發送的電子郵件是垃圾郵件（如圖 1.6）。我們可以不點開郵件查看就直接做出這個預測，接著繼續我們的一天。



圖 1.6 稍微複雜一點的機器學習模型。

案例 3：事情變複雜了！

現在，假設我們繼續這條規則，有一天我們在街上看到 Bob，他問：「你為什麼不來參加我的生日聚會？」我們完全不知道他在說什麼。原來上週日他給我們發了他生日聚會的邀請，而我們錯過了！為什麼我們會錯過？因為他是在週末發送的，而我們認為它是垃圾郵件。看來我們需要一個更好的模型。讓我們回過頭來看看 Bob 的電子郵件，回到我們**記得**的這個步驟。讓我們看看能不能找到新的模式：

- 1 KB：正常郵件
- 2 KB：正常郵件
- 16 KB：垃圾郵件
- 20 KB：垃圾郵件
- 18 KB：垃圾郵件
- 3 KB：正常郵件
- 5 KB：正常郵件
- 25 KB：垃圾郵件
- 1 KB：正常郵件
- 3 KB：正常郵件

我們發現了什麼？似乎較大的電子郵件往往是垃圾郵件，而較小的電子郵件往往是正常郵件。這是有道理的，因為垃圾郵件經常夾帶檔案較大的附件。

因此，我們可以**制定**以下規則：

模型 3：任何大小為 10 KB 或更大的電子郵件都是垃圾郵件，任何小於 10 KB 的電子郵件都是正常郵件。

現在我們已經制定了規則，我們可以做出**預測**。我們查看今天收到 Bob 的電子郵件，大小為 19 KB。因此，我們斷定它是垃圾郵件（如圖 1.7）。

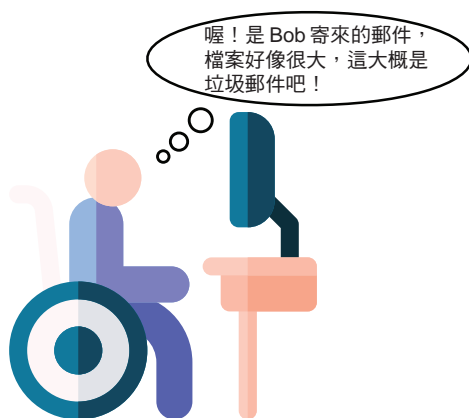


圖 1.7 另一個稍微複雜的機器學習模型。

這就是故事的結局了嗎？還差遠了呢！

但在我們繼續之前，請注意，為了做出預測，我們使用了星期時間和電子郵件的大小，而這些就是特徵 (*feature*) 的例子。特徵是本書中最重要的概念之一。

特徵 (feature) 模型可以用來進行預測之資料的任何屬性或特性。

你可以想像還有許多特徵可以標示出一封電子郵件是垃圾郵件還是正常郵件。你還想得到更多特徵嗎？在接下來的內容中，我們將看到更多特徵。

案例 4：更多？

我們的兩個分類器都很好，因為它們能排除大型電子郵件和週末發送的電子郵件，兩個分類器剛好分別使用這兩個特徵。但是，如果我們想要一個適用於這兩個特徵的規則呢？像下面這樣的規則可能可用：

模型 4：如果電子郵件大於 10 KB 或在週末發送，則將其歸類為垃圾郵件。否則，它被歸類為正常郵件。

模型 5：如果電子郵件是在週間發送的，那麼它必須大於 15 KB 才被歸類為垃圾郵件；如果它是在週末發送的，那麼它必須大於 5 KB 才被歸類為垃圾郵件。反之，則被歸類為正常郵件。

或者我們可以變得更複雜。

模型 6：考慮到星期幾，其中星期一是 0，星期二是 1，星期三是 2，星期四是 3，星期五是 4，星期六是 5，星期日是 6。若我們把星期幾和郵件大小（以 KB 為單位）加起來，結果大於等於 12 的話，則該郵件被歸類為垃圾郵件（如圖 1.8）；反之，它被歸類為正常郵件。

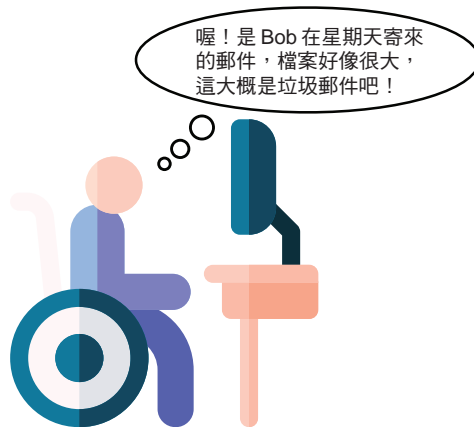


圖 1.8 更複雜的機器學習模型。

所有這些都是有效的模型，我們可以透過增加複雜層級，或查看更多特徵來繼續建立越來越多的模型。現在的問題是，哪個是最好的模型？這是我們開始需要電腦協助的地方。

一些機器使用的模型案例

我們的目標是讓電腦按照我們的思考方式來思考，也就是使用記得 - 制定 - 預測的框架。簡單來說，這是電腦在每個步驟中所做的事情：

記得：查看一個巨大的資料表。

制定：透過許多規則和公式來建構模型，並檢查哪個模型最適合資料。

預測：使用模型對未來資料進行預測。

這個過程與我們在上一節中所做的沒有太大區別，這裡最大的進步是電腦可以透過許多公式和規則組合起來快速建構模型，直到找到一個與現有資料非常吻合的模型。例如，我們可以建構一個垃圾郵件分類器，其特徵包括寄件者、星期幾和時間、詞彙數、拼寫錯誤次數以及出現某些關鍵詞（例如購買或贏）。一個模型很容易看起來像下面的假設句：

模型 7：

- 如果電子郵件有兩個或多個拼寫錯誤，則將其歸類為垃圾郵件。
- 如果附件大於 10 KB，則將其歸類為垃圾郵件。
- 如果寄件者不在我們的聯絡人清單中，則將其歸類為垃圾郵件。
- 如果它有出現購買或贏的詞彙，則被歸類為垃圾郵件。
- 若皆不符合上述條件，則被歸類為正常郵件。

它也可能類似於以下公式：

模型 8：如果（大小）+10（拼寫錯誤次數）-（「媽媽」一詞的出現次數）+ 4（「購買」一詞的出現次數）> 10，那麼我們將郵件分類為垃圾郵件（圖 1.9），反之，我們則將其歸類為正常郵件。

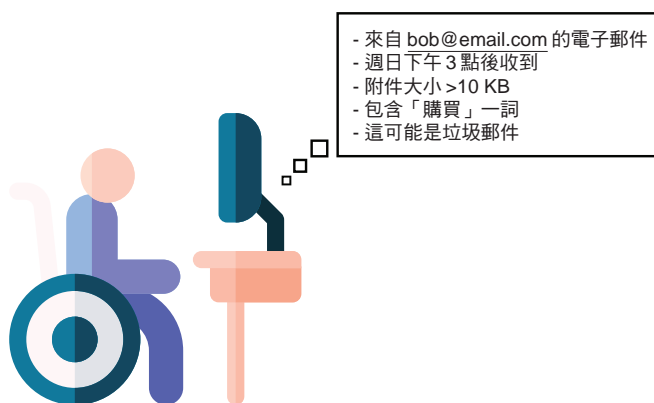


圖 1.9 由電腦所發現更複雜的機器學習模型。

現在的問題是，哪個才是最好的規則？快速答案是，最適合資料的規則就是最好的規則，儘管真正的答案應該是最適合新資料的規則。最終，我們可能會得到一個複雜的規則，但電腦可以制定它並使用它來快速做出預測。我們的下一個問題是，我們如何建構最好的模型？這正是本書所要說明的內容。

總結

- 機器學習很簡單！無論你的背景如何，任何人都可以學習和使用它。所需要的只是學習的意願和好的想法來實作！
- 機器學習非常有用，它被用於絕大多數的學科。從科學到技術再到社會問題和醫學，機器學習正在產生影響，並將持續產生影響。
- 機器學習是常識，由電腦所完成，它可以模仿人類快速準確地做出決策的思考方式。
- 就像人類根據經驗做出決策一樣，電腦也可以根據以前的資料做出決策，這就是機器學習的全部意義所在。

機器學習使用記得 - 制定 - 預測框架，如下：

- **記得**：查看以前的資料。
- **制定**：根據這些資料建構模型或規則。
- **預測**：使用模型對未來資料進行預測。