前言

何謂資料平台？為何需要它？建置資料與機器學習（ML）平台涉及哪些？為何應該在雲端建立資料平台？本書一開始會解答處理資料與 ML 專案時產生的這類常見問題，接著展示在商業營運下建置資料與 ML 功能，建議的策略，這段旅程會展示執行此策略的每一步，將所有概念包裝在模型資料現代化案例裡。

為何需要雲端資料平台？

想像公司技術長（CTO）計畫建置一個行動裝置友善的電子商務網站。他宣稱：「我們的業務正在流失，因為網站沒有針對手機最佳化，尤其是亞洲語言的使用體驗。」執行長（CEO）認同 CTO 所說的現行網站行動使用者體驗不佳的看法，但她也想確認，透過手機訪問平台的用戶是否屬於具盈利潛力的族群，因此致電亞洲營運主管詢問：「使用手機接觸我們電子商務網站的用戶，營收與利潤率為何？若用手機購買的人數增加，對明年的整體營收將有什麼改變？」

亞洲區營運主管該如何回答這個問題？這會需要能夠將客戶訪問資料（確認 HTTP 要求來源）、客戶購買行為（了解他們買了什麼）與採購資訊（掌握商品的成本結構）建立關聯的能力，還需要能夠預測市場不同地區的成長。

地區主管是否必須聯絡 IT 部門，要求整合所有不同來源的必要資訊，並開發一套程式計算這些統計資料？IT 部門是否有時間精力回答這個問題，是不是具備預測分析的能力？

若企業已建立完善的資料平台，這些問題將迎刃而解。在這個案例，所有資料都已收集並整頓完成，可供整個企業組織分析與合成。資料分析團隊僅需執行互動式特定查詢，即可取得所需資訊。他們也能輕鬆利用內建人工智慧（AI）功能，建立或檢索收入與流量封包模式的預測，協助 CTO 評估投資新行動裝置友善網站的要求，做出資料驅動的決策。

回應 CEO 問題的其中一種可行方式，就是取得並部署真實使用者監控（RUM）[譯註]工具。可供使用的專業工具很多，像這樣的一次性決策都有工具可用。擁有資料平台讓企業組織得以回答許多諸如此類的一次性問題，無須取得並安裝一堆特定解決方案。

現代企業組織漸漸走向能根據資料做決策。我們的範例重點在一次性決策，然而許多案例裡，企業組織想要的是自動針對每筆交易重複做決策。例如，企業組織可能想確認購物車是否有被棄置的風險，進而馬上推薦低成本品項消費項目，讓消費者加入購物車滿足免運門檻。這些推薦品項要對個別購買者有吸引力，因此需要可靠的分析與 ML 能力。

為了依據資料做決策，企業組織需要簡化下列內容的資料與 ML 平台：

- 存取資料
- 執行互動、特定查詢
- 產生報告
- 依據資料自動做決策
- 業務服務個人化

在本書中，你將發現雲端為基礎的資料平台可降低上述所有功能的技術門檻：能從任何地方存取資料、執行迅速、即便在邊緣裝置上也能執行大規模查詢，還提供許多分析與 AI 能力服務的優勢。然而，能夠將所有建構區塊準備就緒達成目標，有時會是一場複雜的旅程。

本書旨在協助讀者更瞭解建置現代雲端資料平台的主要概念、架構模式與可用工具，對自身所處企業資料能有更佳能見度與控制，進而做出更有意義與自動化的商業決策。

身為本書作者，我們都是擁有多年經驗的工程師，曾協助各行各業、遍布全球的企業建置資料與 ML 平台。我們發現，許多企業渴望從自身的資料中獲取洞察，卻無法快速擷取、整合各類資料，因而最終體認到，他們必須自行建置現代化的資料與 ML 平台。

譯註　根據索引與維基百科，RUM 全名為 real user monitoring，並非這裡原文的 real-time user monitoring。

這本書寫給誰？

本書是為了想透過公有雲技術建立資料與 ML 平台，支援所處商業組織以資料驅動決策的建構師所寫。資料工程師、資料分析師、資料科學家與 ML 工程師會發現，這本書有助於瞭解他們想實施之系統的概念設計。

數位原生公司已如此運作多年。

早在 2016 年，Twitter 曾解釋（*https://oreil.ly/OwTy4*）其資料平台團隊主張「支援與管理各種商業目的資料生產與消費系統，包括公開報告指標、建議、A/B 測試、廣告定位等。」在 2016 年，這涉及維護全球最大 Hadoop 叢集之一，而在 2019 年，這已含括支援使用雲端原生資料倉儲解決方案（*https://oreil.ly/xeud3*）。

另一例則是：Etsy 表示（*https://oreil.ly/4vckj*）其 ML 平台「透過開發並維護 Etsy ML 從業人員賴以大規模建立原型、訓練與部署 ML 模型的技術性基礎架構，支援 ML 實驗。」

Twitter 與 Etsy 皆建置了現代資料與 ML 平台。兩家公司的平台各異，支援不同型態資料、人員與平台需支援的商業使用案例，但底層手法相當類似。本書，將展示如何建置現代資料與 ML 平台，賦予工程師在所處商業組織具備下列能力：

- 從各種來源收集資料，例如作業資料庫、用戶點擊串流、物聯網（IoT）裝置、軟體即服務（SaaS）應用等等。
- 瓦解企業組織不同部門的孤島。
- 匯入資料或載入後處理資料，同時確保適當處理資料品質與治理。
- 定期或非計劃性分析資料。
- 使用預先建置 AI 模型強化資料。
- 建置 ML 模型執行預測分析。
- 定期依資料採取行動，或回應觸發事件或門檻。
- 傳達洞見並嵌入分析。

若處理的是企業資料的 ML 模型，這本書是架構考量的絕佳介紹，因為你會需要在由資料或 ML 平台團隊所建置的平台上處理工作。因此，身為資料工程師、資料分析師、資料科學家或是 ML 工程師，就會發現這本書有助於獲得高階系統設計觀點。

儘管主要經驗在 Google Cloud，但我們盡力維持服務雲端中立的願景，引進但不限於三大雲端供應商（Amazon Web Services[AWS]、Microsoft Azure 與 Google Cloud）的範例建立基礎架構。

本書大綱

本書由 12 個章節組成，第 2 章將詳細解說以資料進行創新對應的策略性步驟。最後以模型使用案例場景，說明企業組織著手處理現代化歷程的方式。

本書流程視覺呈現如圖 P-1 所示。

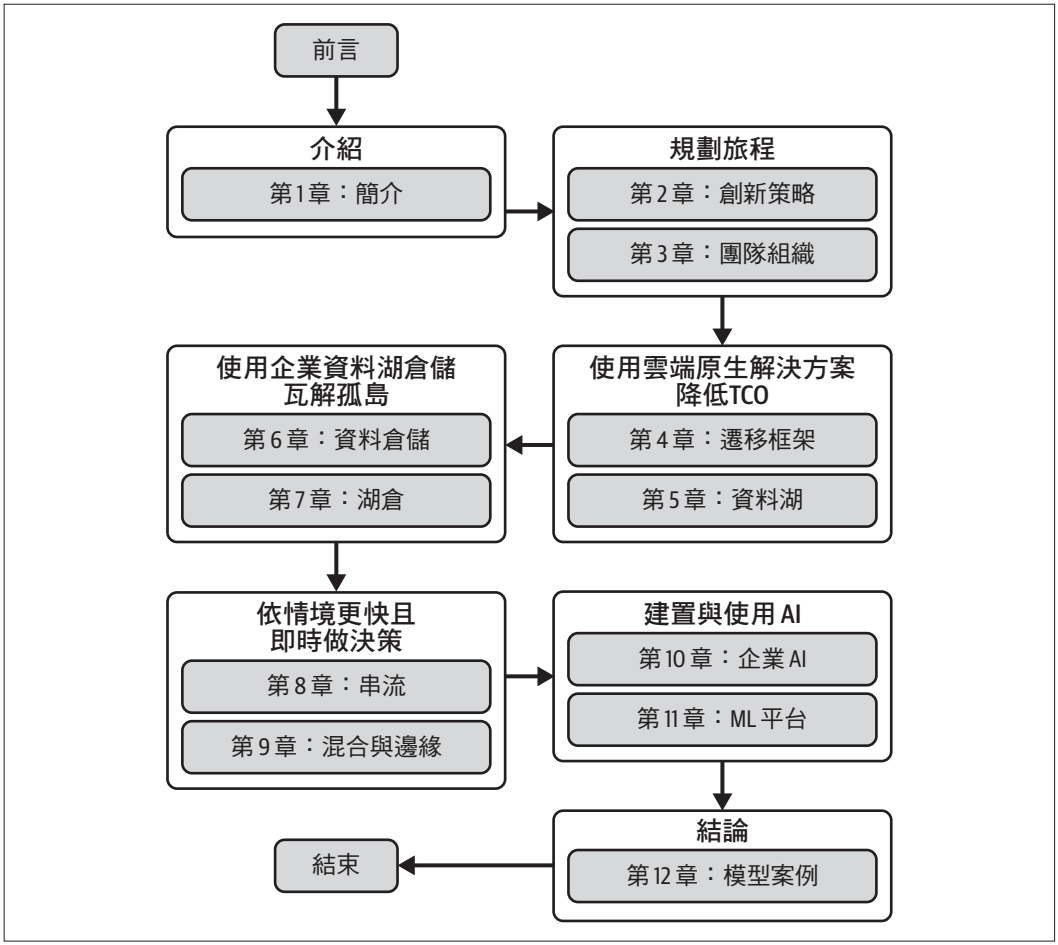

圖 P-1　本書流程圖

第 1 章，探討為何企業組織應建置資料平台。內容還會涵蓋資料平台處理方法、技術趨勢與核心原則。

第 2 章與第 3 章，會更深究如何規劃這趟旅程、確認創新的策略性步驟與如何影響變革。這部分會探討降低所有權總成本（TOC）、除去資料孤島與如何充分利用 AI 解鎖創新這類概念。還會分析資料生命週期的建置區塊、探討如何設計資料團隊，與推薦採用規劃。第 4 章，將這一切匯整到遷移框架。

第 5、6 與 7 章，探討三個資料平台最常見架構：資料湖（第 5 章）、資料倉儲（第 6 章）與資料湖倉（第 7 章）。我們示範了可以兩種方式擇一建置湖倉，從資料湖或資料倉儲擇一開始進化到這個架構，再探討這兩種方式如何抉擇。

第 8 章與第 9 章，探討基本湖倉模式的兩種常見擴充。我們會展示如何透過引進串流模式依情境更快且即時做決策，以及如何擴展至邊緣支援混合架構。

第 10 章與第 11 章介紹在企業環境下建置與使用 AI/ML 的方式，以及如何設計架構以利創新模組的設計、建置、服務與編排。兩章皆涵蓋預測性與生成式 ML 模型。

最後，第 12 章，將走一遍傳統資料現代化案例之旅，重點會放在如何從舊有架構遷移至新架構，藉由企業組織選擇的特定解決方案，解釋處理程序。

若你是為企業建置資料與 ML 平台的雲端架構師，請依序閱讀本書所有章節。

若你是資料分析師，任務是產生報告、儀表板與嵌入分析，請讀第 1 章、第 4 至 7 章與第 10 章。

若你是資料工程師，負責建立資料管道，請讀第 5 至 9 章。其他章節可以略過但視為參考，遇到特定應用程式型態需求時或能派上用場。

若你是資料科學家，被交付建置 ML 模型的任務，請讀第 7、8、10 與 11 章。

若你是 ML 工程師，有興趣瞭解操作化 ML 模型，第 1 至 9 章可以略過，請詳讀第 10 章與第 11 章。

技術框架

資料工程團隊已要求備有廣泛各種技能，不要期望他們還要維護運作資料管道的基礎架構，讓他們的工作更難。資料工程師應著眼於如何清理、轉換、強化與準備資料，而不是解決方案可能需要多少記憶體還是多少核心。

參考架構

匯入層由即時資料使用的 Kinesis、Event Hubs 或 Pub/Sub，與批次資料的 S3、Azure Data Lake 或 Cloud Storage 組成（見圖 3-5、3-6 與 3-7），無須任何預先配置的基礎架構，這裡所有解決方案在各類使用案例皆適用，因為能自動化擴充，滿足輸入工作負載需求。

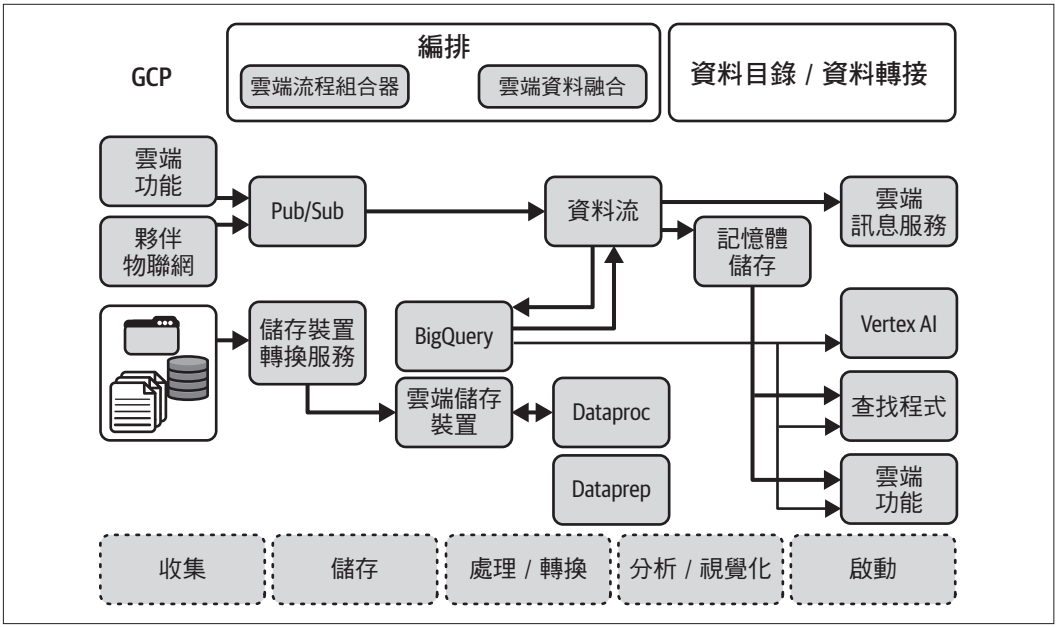

圖 3-5　Google Cloud 資料工程驅動型架構範例

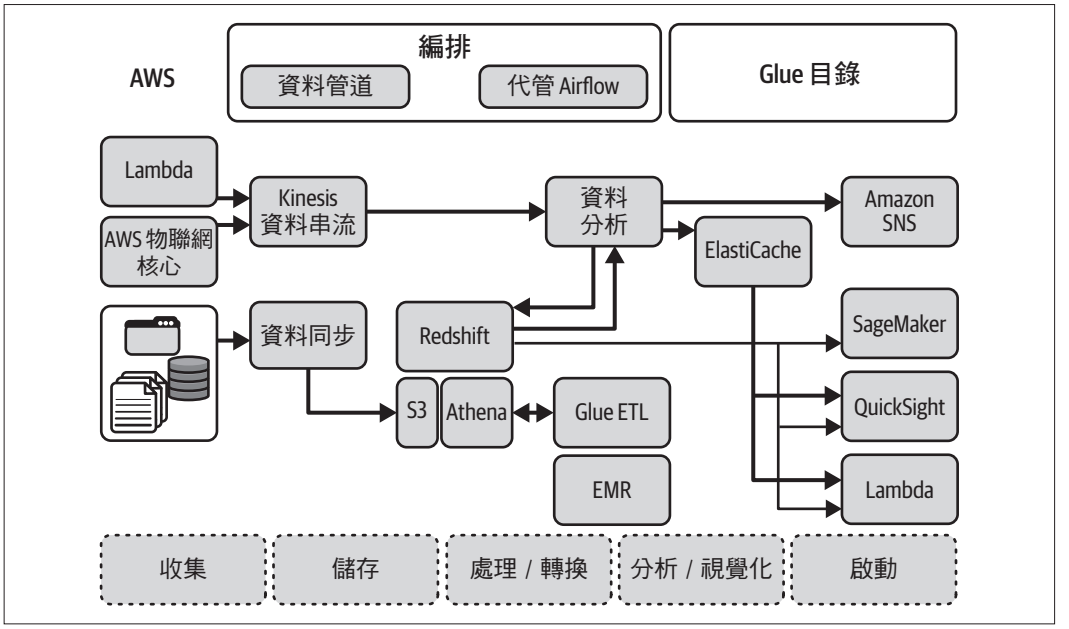

圖 3-6　AWS 資料工程驅動型架構範例

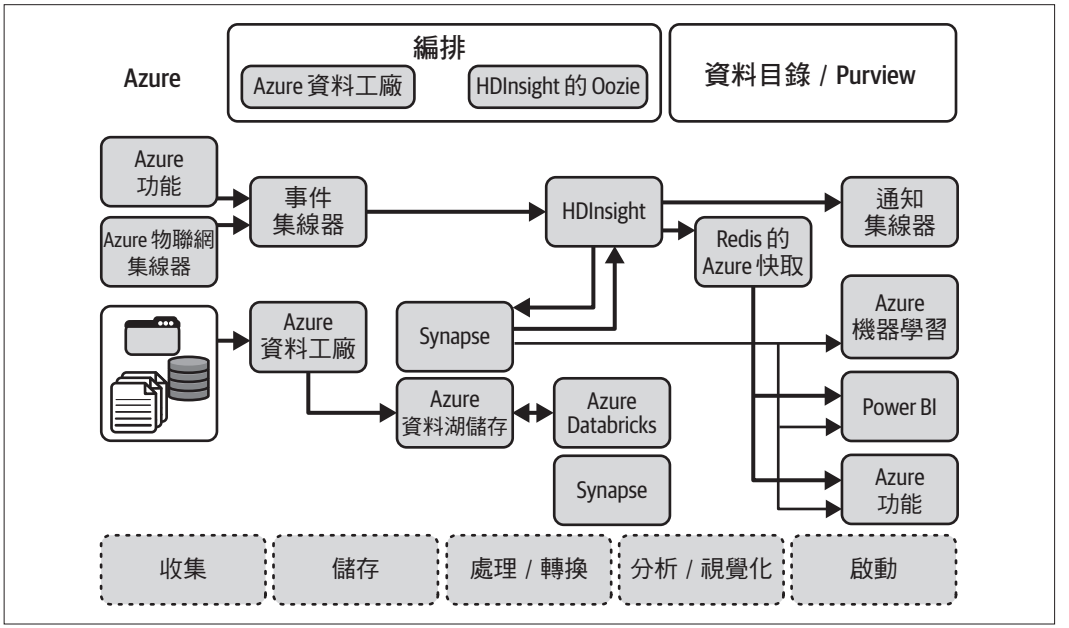

圖 3-7　Azure 資料工程驅動型架構範例

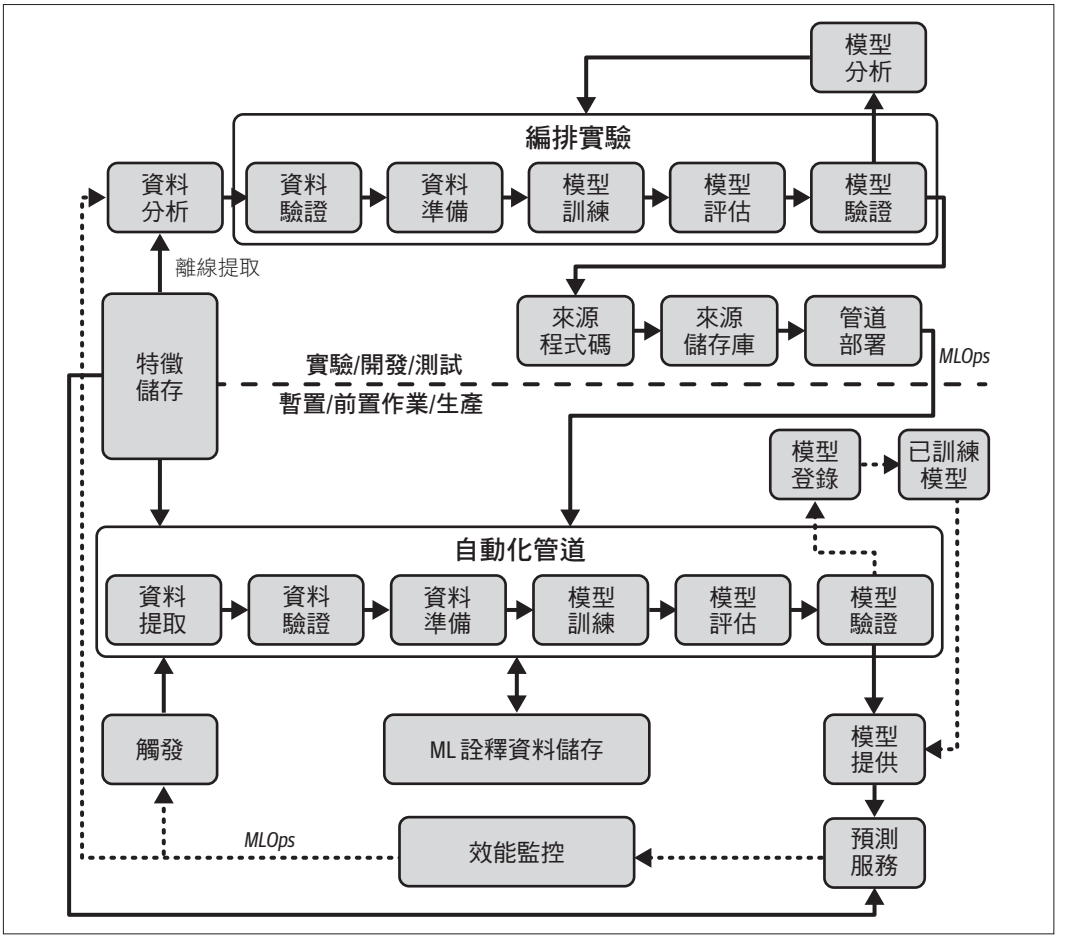

圖 3-8　使用公有雲的 ML 管道產品，可重複資料科學實驗並穩健部署

總結

本章，已瞭解確保所處企業組織類型成功，設計資料團隊的各種方式。最佳方式是找到正確技能與經驗的組合，補充現有團隊不足並協助達成商業目標。關鍵要點如下：

- 雲端技術使新作業方式存取更便利。所有既定身分的潛在可發揮空間擴充了，資料工作者如今能夠用手邊的資料做更多事，且更有效率執行。

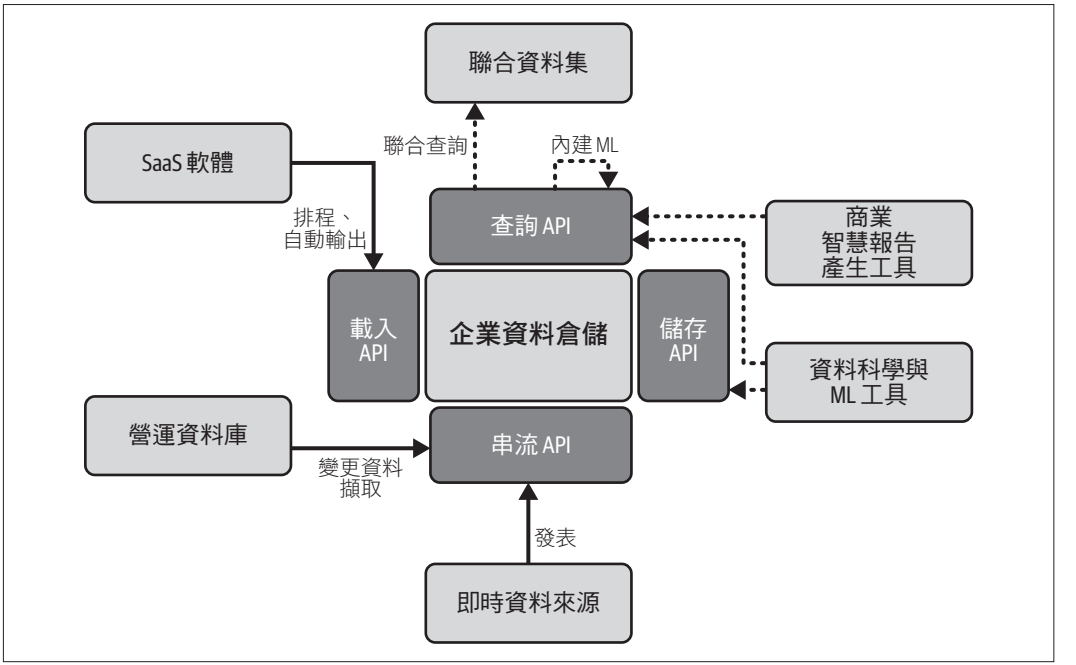

圖 6-1　軸輻式架構；現代企業級 DWH 成為中軸

分析為先的資料與 ML 能力，涉及到將原始資料載入 DWH（*企業級 DWH*），接著執行支援各種需求的必要轉換。由於運算與儲存隔離，只會有單一版本資料。查詢（*Query API*）、產生報告 / 互動式儀表板（*商業智慧報告產生工具*）及 ML（*資料科學與 ML 工具*）這類不同的運算工作負載，在座落於 DWH 的資料上運作。由於可以充分利用 SQL 進行所有轉換，所以可以使用一般檢視與具體化檢視加工，無須使用 ETL 管道。這些檢視可以呼叫外部功能，從而能夠利用 ML API 與模型強化 DWH 的資料。某些情況下，甚至能只用 SQL 語法訓練 ML 模型，或透過簡單 SQL 命令排程複雜的批次管道。現代 DWH 還支援直接匯入（*Load API*）甚至串流資料（*Streaming API*），因此只有必須在資料進來時執行低延遲、視窗彙總分析時，才借助串流管道的能力。要一直記得評估解決方案最終成本，將之與能得到的好處比較。有時一種策略（批次）可能優於其他（串流），反之亦然。

AI 架構

大致來說，目前成功的有七種 AI 應用：

瞭解非結構化資料

例如，有一張設備照片，想要 ML 模型識別設備模型編號。理解影像、影片、語音與自然語言文字的內容，皆屬於這類應用。

產生非結構化資料

訴訟期間可能會收到一份「初步披露」的文件，會需要摘要收到的資訊。或者，想依據消費者最後五次購買行為建立個人化廣告，交織成一個故事。現在已經能用 AI 產生文字、影像、音樂與影片。

預測結果

例如，擁有購物車一籃品項與購物者購買記錄的資訊，想確認是否購物車沒有購買行為被放棄。預測事件的可能性與重要程度，落在此種應用。

預估值

例如，想預估下週指定商店將販售的特定類別品項的數字。預估隨時間變動的數量，像是未來時間點庫存量，就落入這個種類。預測結果與預估值的差異不一定明確。預測結果時，其他屬性會主導事件是否發生；預估時，一段時間的變動模式是最重要因子。有些問題可能得兩種方法都嘗試。

異常偵測

有很多案例是為了識別不尋常的事件或行為。例如，在遊戲場景中，異常偵測可用來識別玩家可能作弊的狀況。

個人化

想要依使用者與其他相似使用者過去喜歡的為基礎，推薦產品或設計體驗。

自動化

針對當下無效率執行的程序，試圖利用 ML 部分或全部取代之。

本節，將檢視每項正規架構與技術選擇。這裡所有架構，都將建置在下一章探討的平台上。

指令調整，可以教模型新任務而不是新知識（撰文此時，這必須從頭開始在新資料上重新訓練模型）。為了低成本教導 LLM 新知識，常見做法是利用 LLM 能力從新範例解決新任務（少樣本學習）、利用填入提示的資訊（檢索增強生成），或產生可傳遞至外部 API（代理）的結構化資料。例如，為了回答問題，嵌入問題或假設性答案，針對文件區塊的向量資料庫搜尋找出相關文件，接著要求 LLM 摘要這些文件區塊，回答提出的問題（見圖 10-3）。開源抽象層（例如 LangChain）逐漸成為實作這些模式的熱門方式。以這種方式架構正式上線使用案例，鏈的每一步都是 ML 管道的容器。容器化 ML 管道將於下一章說明。

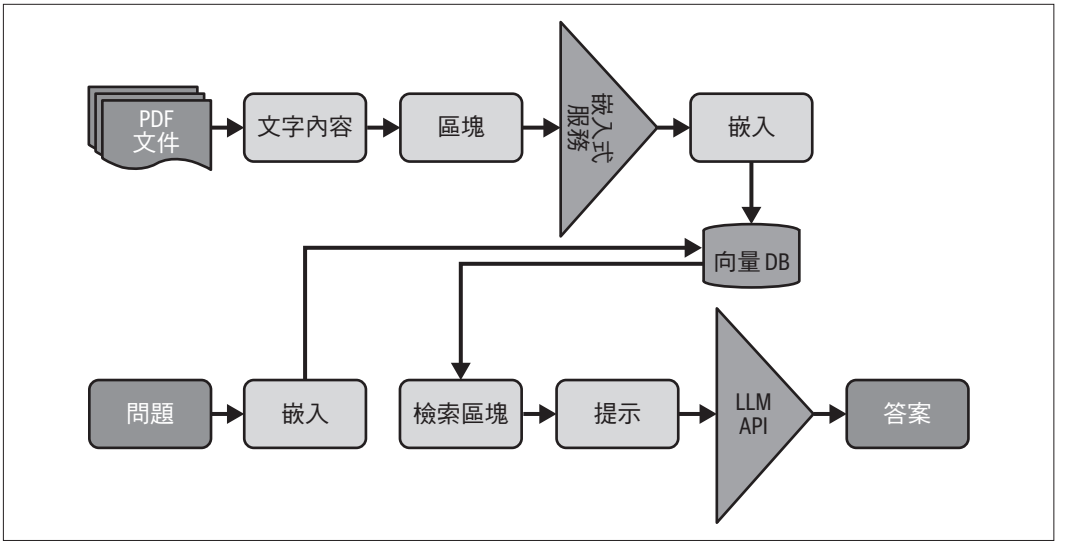

圖 10-3　檢索增強生成架構：利用 LLM 的問答模式

預測結果

作為訓練資料預測結果使用的歷史資料，多半存放在事件紀錄資料與交易資料庫裡。舉例來說，若你想預測購物車是否會被放棄，就必須取得所有網站活動的歷史資訊，多半在事件紀錄資料裡。可能選擇包含品項價格、有效折扣與客戶是否實際購買該品項的資訊。雖然這些資料部分或全部或許能從網站事件紀錄檢索，但從資料庫查詢可能更輕鬆。因此，這種情況下歷史資料也包含標籤（見圖 10-4）。

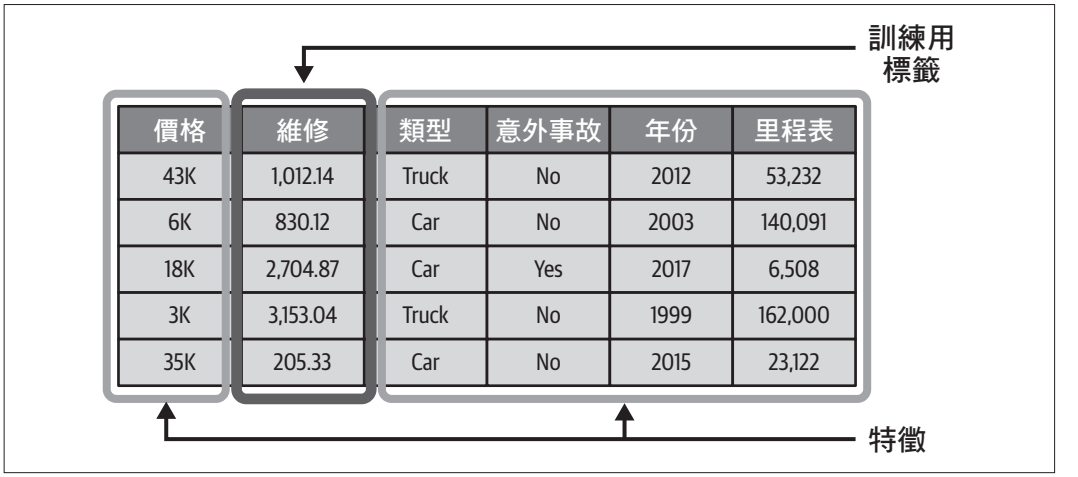

圖 10-4　預測結果模型通常涉及訓練模型，依其他欄值為基礎，預測 DWH 另一個欄值

因此，這個架構（見圖 10-5）包含使用複製機制與連結器，將所有必要資料放進 DWH。接著可以直接在 DWH 上訓練 ML 模型回歸或歸類，Amazon Redshift 與 Google BigQuery 都支援直接從 SQL 訓練模型，而較複雜模型則分別移交給 SageMaker 與 Vertex AI 處理。模型一旦訓練完成，便能匯出並部署至兩種雲端的代管預測服務。通常，所有預測或至少其中一例，會被存回 DWH，支持後續評估與飄移偵測。

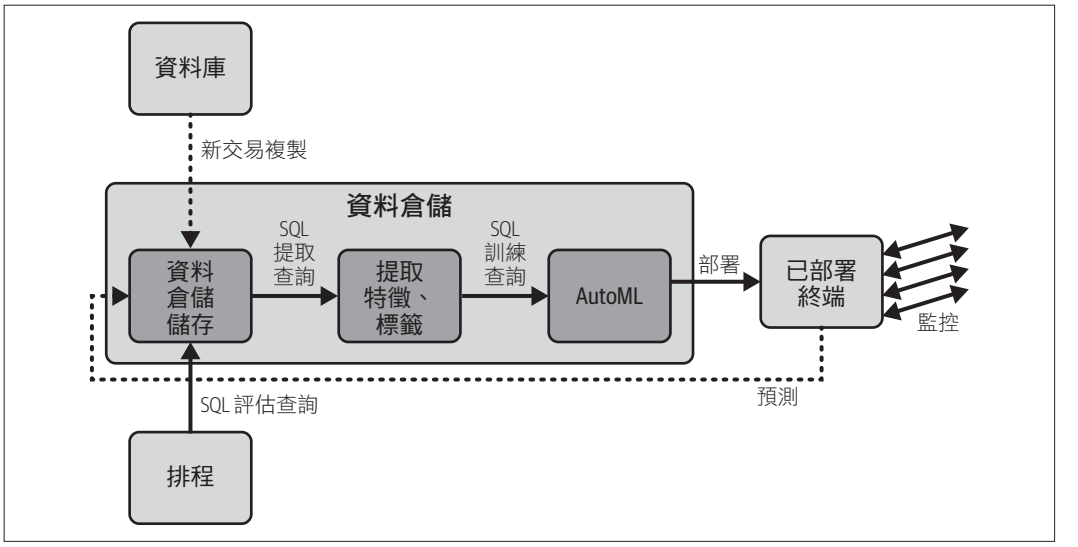

圖 10-5　訓練、部署與監控預測結果模型的架構

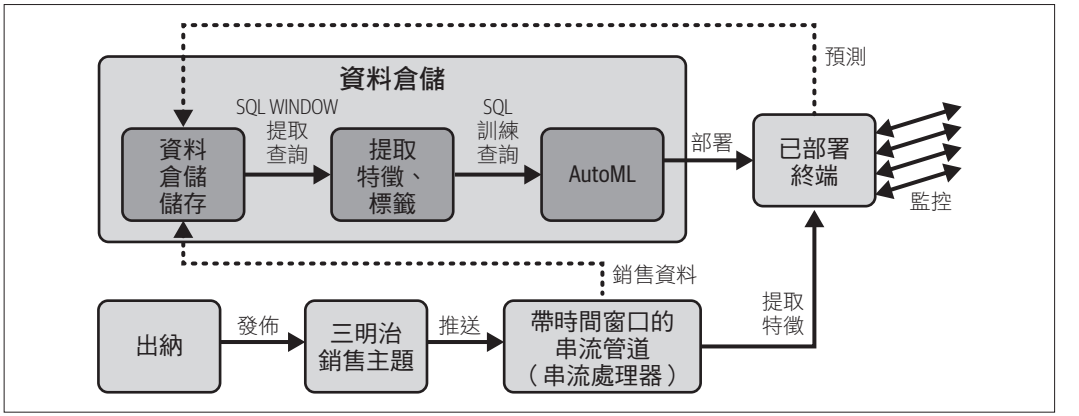

圖 10-7　批次與串流管道各自獨立的時間序列分析架構

然而，即時處理的情況下時間窗口必須在串流管道裡執行。串流管道將 28 天資料拉進來，再傳給預估模型進行預測，同時也負責保持 DWH 更新，讓下一個訓練可以在即時資料出現時執行。這種處理方式問題在於，必須維護兩個管道：批次管道依歷史資料訓練，串流管道依即時資料預估。

較好的方式，是使用 Apache Beam 這類提供在批次與串流資料集兩者執行相同程式碼的資料管道框架。這可以限制訓練 - 應用偏差：因訓練與預測期間不同的預先處理步驟差異所導致。此類架構如圖 10-8 所示。

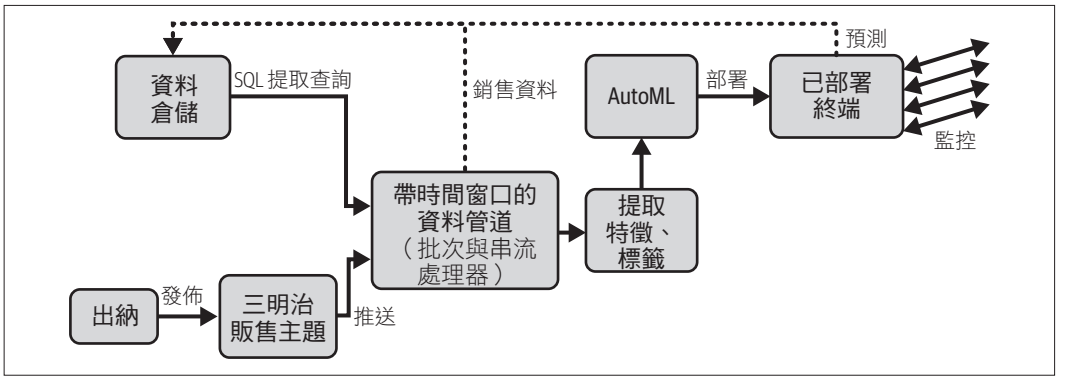

圖 10-8　使用 Apache Beam 這類支援批次與串流資料管道技術的架構

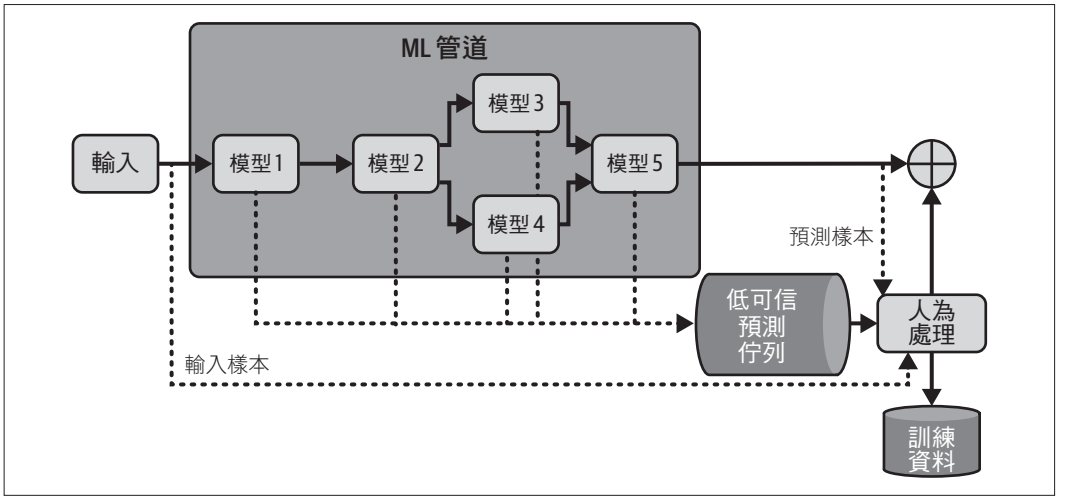

圖 10-11　眾多 ML 模型鏈結的自動化管道

ML 模型訓練與部署需視型態而定（例如 AutoML、預先建置 API 或客製化模型）。整組模型整合到管道裡，接著在公有雲現有代管服務 Kubeflow Pipeline 這類系統裡編寫。佇列可以是任何分散式任務排隊：AWS Simple Queue Service、 Google Cloud Pub/Sub 諸如此類。訓練資料將放在雲端物件儲存還是 DWH，需視非結構化或結構化而定。

負責任 AI

AI 功能強大，你必須小心不濫用，明瞭所處企業與政府的政策與法規。想像這個使用案例：想要降低人們花在等電梯的時間。所以，要自動識別人們走進建築物大廳，再利用對他們辦公室所在樓層的瞭解。這麼做，可以最佳化引導個人進入電梯，讓停留的次數最少。這個使用案例在很多地方管轄權上都違返隱私權法規，因為嚴禁拍照與追蹤人們。

ML 模型不可能完美，你必須對 ML 的使用更謹慎：那些自動決策無須人為介入的部分。有名的 Zillow 在終結建物過度庫存後，不得不叫停它們的自動化購屋模型（*https://oreil.ly/atiBv*）：房屋被低估的人不會賣給 Zillow，而被高估的人急著賣掉。

想進一步瞭解這些概念，我們推薦 Patrick Hall、Navdeep Gill 與 Benjamin Cox 所寫的《*Responsible Machine Learning*》（O'Reilly）。