本書的目標讀者與必要的先備知識

本書主要對 AI 工程師與系統工程師介紹在特定商業場合應用機器學習所需的工作流程與系統，以及建立開發與維護團體的方法。

本書會說明機器學習系統的雲端架構，以及利用 Python 建立機器學習系統的實例，說明應用機器學習的方法，維護與改善機器學習系統的重點。

為了確認程式碼可以執行，本書使用的平台為 Docker 與 Kubernetes。

程式設計語言則以 Python 為主。在開發部分 Android 應用程式的時候會使用 Kotlin。

此外，使用 Poetry 管理 Python 函式庫。

本書使用的機器學習函式庫主要為 TensorFlow、scikit-learn、LightGBM。資料處理的函式庫則使用 pandas、Numpy、pandera 這些函式庫。

在建置系統時，會使用 Argo Workflows 這套工作流程引擎，搜尋引擎則使用 Elasticsearch，機器學習管理則使用 MLflow，網頁畫面則使用 Streamlit 製作，至於 Web API 的部分則使用 FastAPI 以及其他軟體製作。

本書也會使用各種函式庫或軟體建置系統，其中較重要的函式庫或軟體則會於內文進一步說明。

INTRODUCTION 本書的編排方式

本書總共分成四個章節。

第 1 章「課題、團體、系統」會說明如何挑出該以機器學習解決的商業課題，以及解決該課題所需的工作流程與系統的建置方法，最後還會說明應用機器學習的團隊該如何組成。

第 2 章「建立需求預測系統」則是以虛擬的全國 AI 商店為前提，說明如何利用機器學習預測飲料需求的流程。

第 3 章「利用動物圖片應用程式建置違規內容偵測系統」則會說明以虛擬的動畫圖片分享應用程式「AIAnimals」偵測使用者違規行為，以及禁止使用者違規的工作流程，以及建置機器學習系統與評價系統的方法。

第 4 章「於動物圖片應用程式的搜尋功能使用機器學習」則會說明在「AIAnimals」的搜尋系統使用機器學習，改善搜尋使用者體驗的方法。

OPERATING ENVIRONMENT 本書範例檔的執行環境

本書於 GitHub 發表的範例檔已於 表 1 的環境確認可正常執行。

表 1 執行環境

Linux	
項目	內容
Ubuntu	22.04 LTS
處理器	3.60GHz 4 核 Intel Core i3-10100F
記憶體	32GB
GPU	NVIDIA GeForce RTX 3060 Ti 8GB

1.2 設計利用機器學習解決課題的腳本

「設定以機器學習解決的課題」這句話讓人覺得公司內部已經有可以透過機器學習解決的課題，也可以立刻為了解決該課題而開發機器學習。不過，在各種商業應用場景之中，有許多有待解決的課題，而我們必須從中找出可利用機器學習解決的課題，以及評估利用機器學習解決該課題的優點是否高於導入機器學習的缺點。

比方說，有許多企業都在網站使用了聊天機器人，但真的有必要利用機器學習讓這個聊天機器人學會對話嗎？（ 圖 1.4 ）雖然造訪網站的使用者會因為不知道該怎麼使用商品而詢問聊天機器人，但不代表在這種聊天機器人使用機器學習是件合理的事情，因為聊天機器人不過是種工具。想利用聊天機器人這種工具解決的課題，通常是以 Q&A 的方式一步步釐清使用者的問題再提出解決方案。

其實就算不使用機器學習，也可以將這種聊天機器人設計成選項式問卷，讓使用者透過選項找到理想的解決方案，這種方式一樣能夠解決課題。這種問卷方式在技術層面上，遠比機器學習來得簡單，有時甚至能創造更優質的使用者體驗，而且問卷方式的開發工序也比較簡單，只需要先列出問題與選項，之後再利用多個 if-else 條件式打造一問一答的流程，就能解決問題。

如果利用機器學習打造聊天機器人，就必須先將使用者對商品的疑問轉換成文章，再匯入聊天機器人，然後讓聊天機器人透過機器學習理解這些文章與找出解決方案，然後再將解決方案輸入聊天機器人。要讓聊天機器人透過機器學習了解文章，就必須先準備大量的問題以及對應的解決方案，然後再開發機器學習模型，以及將該模型植入正式的系統。要準備如此大量又完全正確的資料集是件非常困難的事，而且當商品的使用方法改變時，就得另外準備新的資料集。雖然商品的使用方式不一定都這麼複雜，更新的頻率也不一定很高，但是問卷式聊天機器人的開發與維護流程，應該會比機器學習式聊天機器人簡單十倍，而且問卷式聊天機器人的題目還可以翻譯成英文或中文，支援不同的語言。若是以機器學習開發聊天機器人，就必須為了支援多國語言而準備各種語言的文章，再為了這些語言的文章建立不同的機器學習模型。

我的意思不是利用機器學習開發聊天機器人絕對不符合效率，而是在使用機器學習之前，一定要先思考，該課題是否真的非使用機器學習解決不可。

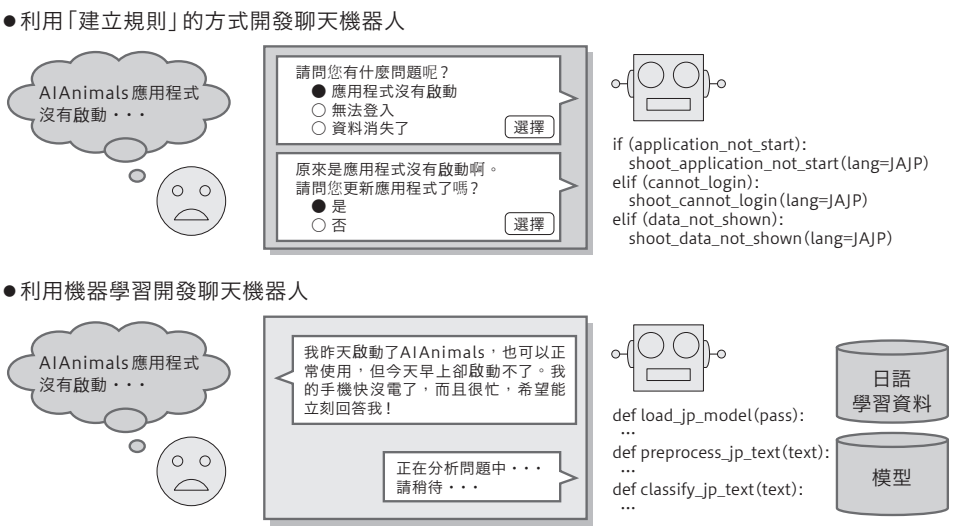

圖 1.4 利用「建立規則」模式開發的聊天機器人與利用機器學習開發的聊天機器人

必須利用機器學習解決的課題，是那些可以用其他方法解決但品質和成本不如機器學習的課題。本書先前說明的網頁佈告欄規則違反偵測的圖像辨識就是其中一例。近年來，機器學習辨識圖像的精確度已遠遠高於人眼，而且都能自動辨識。如果是能以圖像辨識解決的課題，那麼使用機器學習當然也是合情合理的選擇。不過，圖像辨識的課題不一定都得利用機器學習解決。以網頁佈告欄一天最多只有十篇圖像貼文，上傳人臉視為違規的情況來看，員工一下子就能從十篇圖像貼文之中找出違規的文章，判斷該文章的圖像是否為人臉也不會超過 30 秒，所以 10 篇文章 ×30 秒 =300 秒，也就是 5 分鐘，換言之，要判斷上傳的圖像貼文是否違規只需要 5 分鐘。要不要將這 5 分鐘的作業交由機器學習處理端看公司的情況，但就大部分的情況來說，讓機器學習解決更重要的課題會是比較好的選擇。

話說回來，如果網頁佈告欄一天有一百萬篇圖像貼文又當如何？就算辨識一張圖片只需要 30 秒，一百萬篇貼文就得耗費 100 萬篇 ×30 秒 =3,000 萬秒，如果全由一個人來做，相當於需要 347 天才能完成，如果 1,000 個人來做，大

概可在九個小時之內完成（但是不能休息）。光是為了辨識圖像就雇用一千名員工，絕對不符合經營成本。此時就該利用機器學習自動辨識圖像，才能刪減一千名員工的人事成本，使用機器學習也變成合理的選擇。

要利用機器學習解決課題就必須先定義利用機器學習解決課題的狀態，若以上述偵測違規貼文為例，就是能夠刪減人事費用，維持與改善服務品質。所謂的「刪減人事費用」是指，維護機器學習的人事費用低於以人力篩檢違規貼文的人事費用的狀態。至於維持服務品質則是指，利用機器學習偵測違規貼文的疏漏率，低於服務得以正常營運（避免使用者瀏覽多餘的圖片或是惡質內容，留住使用者的意思）的疏漏率的意思。雖然是為了達成這兩個目標而導入機器學習，但使用機器學習不代表沒有任何風險，也不代表能零成本解決課題。要讓機器學習能付諸實用，需要聘請工程師，當工程師的人事費用超過以人力辨識圖像所需的人事費用時，就無法達成刪減人事費這個目標。此外，就算機器學習能 100% 正確辨識人臉，卻有 50% 的機率誤判動物的臉，此時就必須修正機器學習的規則，將動物表情的貼文視為合格貼文。換言之，必須另外解決這種誤判的問題（或是判斷這個問題是否不需要解決）。

所謂利用機器學習解決課題的腳本就是選擇課題、釐清解決課題的方式、定義以機器學習解決課題的狀態，以及在解決課題之後，找出解決新課題的方法。幾乎沒有任何方法或技術可以完美地解決課題，而且社會、企業與個人所面對的課題通常很複雜也很難解決。本書會在**第 2 章**、**第 3 章**、**第 4 章**定義需求預測、違規預測、搜尋引擎的課題，以及介紹利用機器學習解決這些課題的方法。雖然無法 100% 解決這些課題，但至少能解決 50% 以上的問題。盡管本書介紹的課題與讀者遇到的課題不盡相同，但如果能為大家的課題帶來一線曙光，那將是作者的榮幸。

1.5 新的機器學習系統設計模式

前面已經提過，機器學習系統設計模式可根據運用機器學習的系統架構以及維護方式分成不同模式。機器學習系統設計模式的內容當然不是全部，但新的設計模式往往會因為應用各種機器學習的專案而誕生。

接下來要在機器學習系統設計模式追加新模式。於本節追加的模式會是後續使用的系統設計。

1.5.1 評估儀表板模式

機器學習是透過數據進行評估，評估指標也有非常多種，例如正確率、平均誤差都是其中一種。當推論對象的資料增加，就得針對每筆資料進行評估，得確認的評估結果也會跟著增加。全國連鎖門市或是企業銷售的商品種類多達數百種，利用機器學習針對每間門市或是商品種類進行評估，再確認這些評估結果的業務也非常複雜，光是要將這些評估結果整理成清單或是表單的數據，再判讀這些數據，就得耗費不少時間。本節介紹的評估儀表板模式可將這些評估結果轉換成儀表板之中的圖表，讓使用者一眼讀懂評估結果。

使用情況

- 在機器學習的學習結果或是推論結果，需要與訓練資料或是實際資料比較或評估的時候使用。
- 在資料量過多，無法只憑數值一眼看出評估結果的時候使用。

要解決的課題

在應用機器學習的場景之中，很少出現只以所有資料的正確率或是誤差評估機器學習的推論結果。大部分的資料都包含五花八門的項目（例如地區、門市、

時段、商品種類、使用者類別），針對這些項目確認機器學習的評估結果，再針對這些項目研擬對策是非常重要的業務。機器學習的模型評估也會將資料拆解成不同的項目，再針對這些項目分析正確率與誤差。假設機器學習執行的業務負責商品是食品，其會根據食品資料（或是讓食品資料與其他資料比較）研擬提升業績或是降低成本的對策，而且大部分的資料都有很多面向。換言之，資料往往包含多個元素，光是某個食品的銷售成績資料，就可能包含銷路較佳的地區、門市名稱、熱銷時段、製造商名稱、產地名稱、有無折扣、是否為生鮮食品這些元素。除了這些項目資料之外，也會將價格、份量、營養價值這類數值資料統整至某個範圍（群組化）再進行分類。一般來說，會使用 BI（Business Intelligence）或是資料分析的手法根據上述這些元素分析評估結果。為了能一眼讀懂分析結果，就會使用 Redash、Looker 或 Tableau 這類 BI 儀表板。所謂的評估儀表板就是使用 BI 儀表板，讓使用者一眼就能讀懂評估結果的設計模式。

架構

BI 工具有些是免費的，有些則需要付費才能使用，而且種類也非常多。選擇使用哪種工具之後，取得資料的方式、製作圖表的方式、具體呈現的手法、向利害關係人報告的方式都會不一樣。一般來說，BI 工具包含下列這些功能（圖 1.6）。

- 資料連線：與資料庫或是資料倉儲連線，取得以其他功能操作的資料。
- 報表功能：在畫面顯示資料、圖表或是表單，說明形狀與釐清課題。
- 儀表板功能：將資料、圖表與表單轉換成元件，再於畫面顯示。使用者可操作期間、地區這類參數，調整元件的可視化範圍。
- 分析功能：透過線上分析處理或是資料探勘手法，從不同的角度分析資料。可執行切片或是向下鑽研（Drill down）這類資料分析。可分析時序資料的規則與傾向，再預測未來的趨勢。

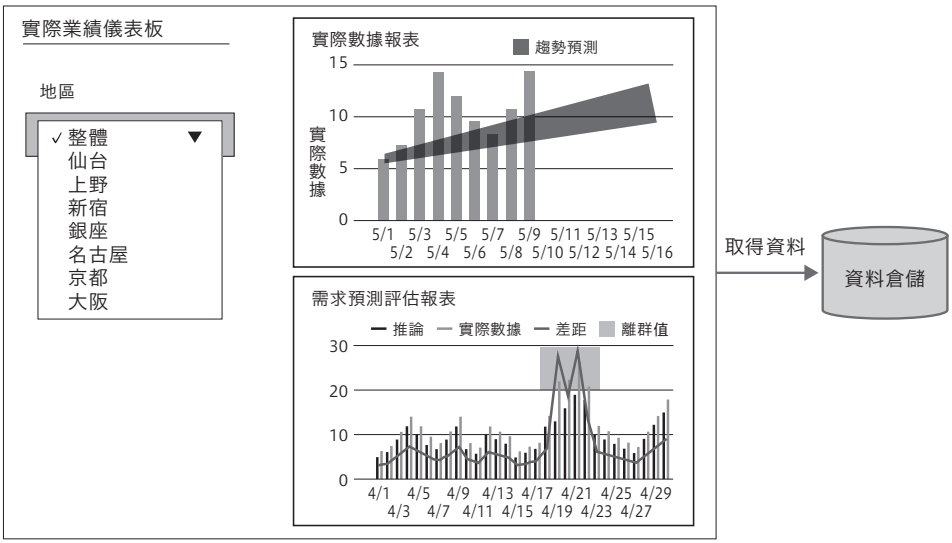

圖 1.6 BI 工具的儀表板範例

透過 BI 工具可視化資訊的方式也能在其他用途使用。比方說，在開發機器學習模型時，就會需要讓學習曲線、ROC 曲線、時序資料推論曲線的評估結果以不同的方式可視化。當推論值的變化越大，需要推論的資料越多，以圖表或是其他可視化手法呈現評估結果，會比只有數據的評估結果來得更有效率，也更能透過直覺理解評估結果。評估儀表板設計模式會將評估結果轉換成圖表，讓使用者從不同的角度了解機器學習模型的有效性（ 圖 1.7 ）。

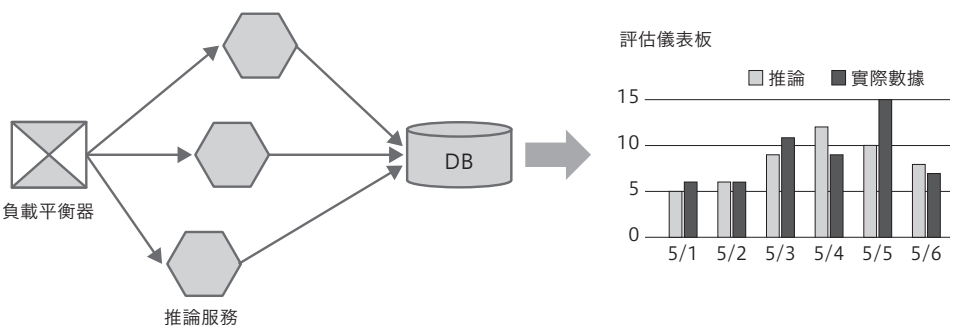

圖 1.7 於機器學習模型開發使用儀表板的範例

使用評估儀表板設計模式的情況大致分成兩種（ 圖 1.8 ）。

1. 機器學習模型發佈之前的評估：在機器學習模型發佈為正式上線的系統之前，比較評估資料與訓練資料時，就會使用這種設計模式。利用個別資料或是群組化資料比較推論結果與訓練資料，分析學習模型如何推論這些資料。在群組化資料的時候，可根據月份、地區、商品種類建立群組，釐清各群組的推論趨勢與評估結果。完成這類分析之後，機器學習的團隊或是負責下決策的高層即可確定機器學習模型是否可發佈為正式上線的系統。
2. 機器學習模型發佈之後的評估：這種設計模式也可在機器學習模型發佈之後，用於評估推論結果與實際數據。比方說，可在儀表板顯示推論結果與實際數據之間的誤差，以及源自這些誤差的課題或是相關的對策。在同一個儀表板顯示機器學習模型的效果以及推論結果與實際數據之間的誤差，可讓團隊成員達成共識。

於評估儀表板可視化資料的方法取決於資料的種類。以時序資料為例，可讓推論結果與訓練資料（或是實際數據）沿著時間軸排列，再利用折線圖或是長條圖整理資料，讓使用者一眼看出這些資料在時間軸上的變化。如果是各地區資料，就可替各地區的資料製作圖表，或是直接將資料植入地圖。如果是商品類型資料或是使用者類型資料，可依照商品類型或是使用者類型整理資料，再根據整理完成的資料繪製圖表。推論結果與實際數據的評估結果可根據資料的種類選擇適當的呈現方式。

假設公司內部已經導入了 BI 工具，可利用現有的 BI 工具建置機器學習的評估儀表板。如果希望獨立管理評估儀表板，或是利用程式繪製圖表，可使用 Streamlit 或 Plotly Dash 這種內建資料分析功能的網頁應用程式。Streamlit 與 Plotly Dash 都可以利用 Python 繪製圖表，也可當成網頁應用程式啟動，在網頁瀏覽器顯示圖表。由於可利用 Python 撰寫程式，所以可使用 Python 內建的資料分析函式庫或是機器學習函式庫。

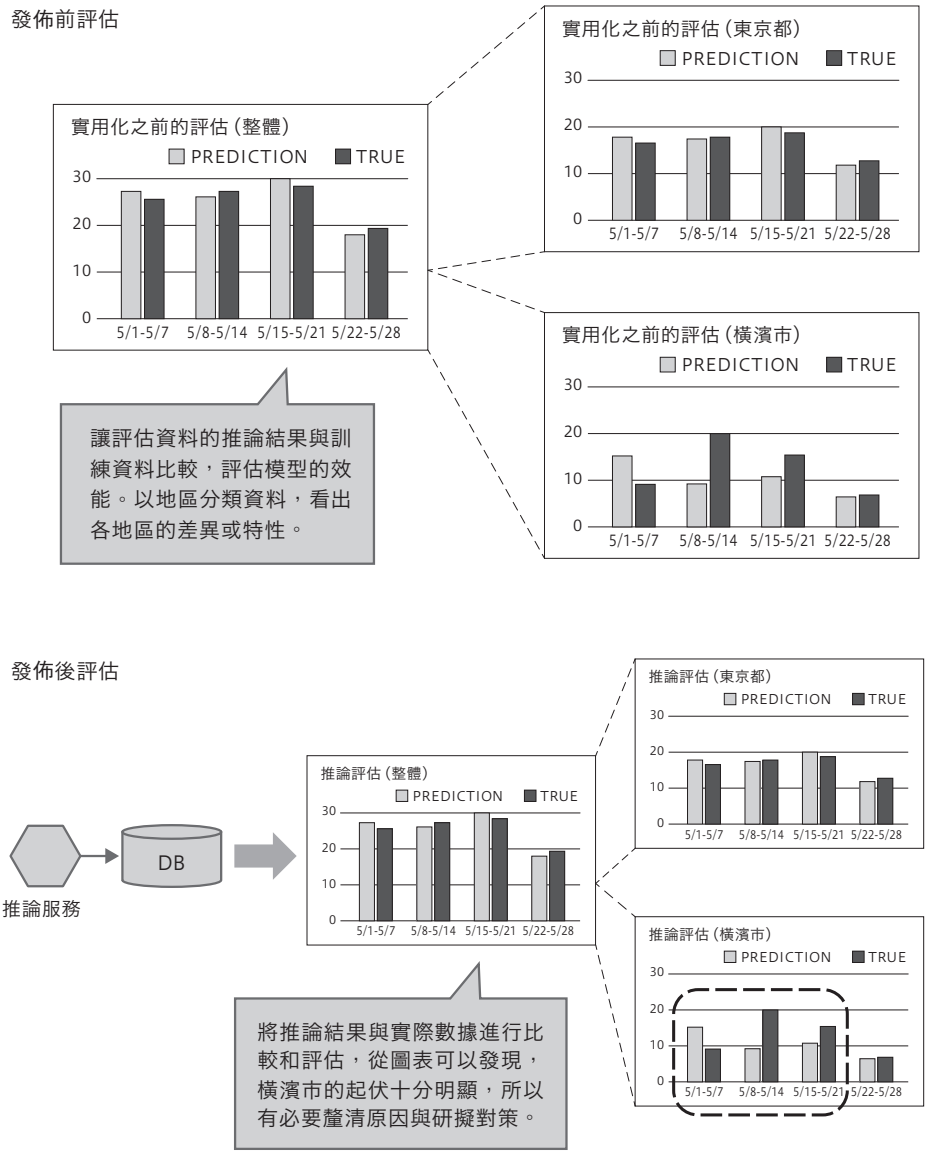

圖 1.8 發佈前評估（實用化之前的評估）與發佈後評估（推論評估）

- **Streamlit**

  URL https://streamlit.io/

- **Plotly Dash**

  URL https://plotly.com/dash/

本書的方針使盡可能使用免費的 OSS 工具建置系統，所以在介紹評估儀表板設計模式時，會介紹以 Streamlit 建置儀表板的例子。

建置

建置的部分將於**第 2 章**說明。

優點

- 就算推論結果的量非常多，也能以簡潔的方式顯示，讓使用者一眼讀懂推論結果。
- 可在網頁或是其他的介面顯示推論結果與實際數據，讓整個團隊達成共識。

檢討事項

評估儀表板設計模式是幫助機器學習團隊或是決策者根據畫面上的結果進行分析與做出決策的工具。換句話說，如果機器學習團隊或是決策者無法理解於儀表板顯示的結果，評估儀表板設計模式就毫無價值可言。最常發生的反面模式就是只在某段時間使用儀表板，而且不看儀表板上的資料就直接做出決策。如果本來就是不需要儀表板的業務，會將儀表板束之高閣也是無可奈何的事，但有時候明明知道儀表板的效果，卻會因為「太忙」或是「不好用」這些理由而不使用儀表板，此時就必須提高儀表板的方便性。在儀表板建置完成後，經常使用儀表板，根據儀表板上的結果研擬商業方針可說是最理想的狀態。此外，為了讓自己根據儀表板上的結果做出決策，可讓儀表板成為例行業務之一，直接於各項任務使用。最重要的是，要在公司內部或是團隊內部證明儀表板的實用性，讓大家認同儀表板的價值。為此，需要先釐清使用者課題，分析現行機器學習進行推論之際的課題，再透過儀表板共享課題。

2.4 需求預測系統與業務的工作流程

到目前為止，說明了透過機器學習預測需求，以及讓這個機器學習付諸實用的方法與程式，本節則要更進一步說明自動化系統的全貌與業務工作流程。

開發與執行系統或工作流程非常曠日廢時，而且在過程中，社會與事業也不斷地變化。要讓專案成功，就必須隨著大環境的改變不斷更新系統與工作流程。因此，接下來要將專案分成初期（2021 年 1 月～ 2021 年 6 月）與發展期（2021 年 7 月之後）這兩段期間，再分別說明這兩段期間的團隊、系統與工作流程。

2.4.1 專案初期的團隊、系統與工作流程（2021 年 1 月～ 2021 年 6 月）

專案初期的團隊只有 1 名機器學習工程師與 0.5 名軟體工程師，而帶領團隊的機器學習工程師負責收集與分析資料、與各門市聯絡、釐清工作流程、工作內容，以及將相關的工作指派給軟體工程師。軟體工程師則負責機器學習工程師不擅長的軟體開發（比方說，建置基礎建設或批次系統，以及檢視程式碼）。在專案的初期時，不需要將所有的系統做到完美的地步，而是要將心力放在真正需要的開發業務（打造精確度高於人類的需求預測系統，以及讓這套系統實用化），其他的系統（採用機器學習架構或是工作流程引擎）則可以在確定專案成功之後才行開發。就算建置了完善的機器學習架構，結果公司不打算使用機器學習的話，也只是白忙一場。於專案初期開發的機器學習模型的資料量或計算量不多，只需要使用筆記型電腦就能開發，所以當下可在機器學習工程師熟悉的開發環境開發即可。比方說，可以使用機器學習工程師的筆記型電腦開發，或是在雲端建置一台伺服器，以遠端存取的方式開發，唯獨開發的程式要利用儲存庫管理，避免只有機器學習工程師可以取得程式。

系統可以使用既有的系統，再準備要追加的東西。首先為大家列出必要的系統元件（ 表 2.7 ）。

表 2.7 必要的系統元件

系統名稱	既有、新增	說明
資料存儲	既有	儲存分析與開發模型所需的資料。目前是以 CSV 檔案的格式存放在公司內部分享空間
分析、學習系統	新增	分析資料、開發模型所需的系統。目前是使用機器學習工程師的筆記型電腦開發，所以要在機器學習工程師的筆記型電腦安裝需要的函式庫
推論值、評估管理系統	新增	管理學習完畢的模型、推論結果與實用化之際的評估結果的系統。目前會將這些內容轉存為 CSV 檔案，再存放至分享空間
儲存庫	新增	管理程式的儲存庫。AI 商店是利用 GitHub 管理，所以要在 GitHub 新增儲存庫
工作管理工具	新增	管理開發工作的系統。可在既有的工作管理系統建立工作空間
與門市聯繫的工具	既有	沿用公司內部聯絡工具 Slack

表 2.7 列出了開發需求預測模型以及將推論值分享給各門市所需的最低限度的系統。雖然有不少是「新增」的系統，但其實會沿用現有的工具或是基礎建設，所以不需要從零開始建置系統。如果想要打造高階自動化系統或是以團隊的方式開發系統，光是這張表格列出的系統可能稍嫌不足，但如果只是由 1.5 名工程師一邊與門市溝通，一邊進行開發的 PoC 專案，表格列出的系統應該已經綽綽有餘了。專案分期的目標是讓機器學習付諸實用，證明機器學習的效果，所以最重要的工作就是建立分析資料、開發模型、將推論結果分享給門市的工作流程，其他的系統或是工作都可以先緩一緩。

這次需要何種工作流程呢？從 **2.3.5 節**的說明便可知道，這次必須在限制時間之內預測飲料需求，以及將預測結果分享給門市，而這次的工作流程的課題在於在星期一之前，無法得到最新的資料，而且所有資料都是時序資料，所以當資料不夠新鮮，推論結果的精確度可能會下降。換言之，為了得到最精準的推論結果，最好能在星期一取得資料，並在星期一的時候預測下週的需求，所以此時必須思考該怎麼做，才能讓專案成員於星期一進行推論，再將推論結果分享給各門市，以及準備一個上述的工作流程無法維持時的備案。換句話說，當星期一為例假日或是機器學習工程師請病假的時候，上述的工作流程依舊得以

運作，門市還是能依照推論結果決定適當的進貨量。此外，上述的工作流程有可能因為某些意外而無法運作，例如因為某些緣故而無法在星期一取得實際銷售數據，或是現有的機器學習模型突然無法正確預測，抑或公司內部的聯絡工具突然故障，無法將預測結果分享給各門市的時候，甚至因為大地震或其他天災導致沒有餘力執行工作流程的時候，都有可能導致工作流程無法正常運作。需求預測系統或是 AI 商店若是發生問題，來買飲料的顧客就有可能因為在 AI 商店買不到想要的商品而去其他的店家購買。如果能夠避免這類問題發生，就能避免損失銷售機會。

雖然天有不測風雲，但只要能未雨綢繆，就能防患於未然。

- 年初就能知道星期一是否為例假日，所以能調整行程，改在隔天的星期二進行推論，再將推論結果分享給各門市。假設是黃金週或是新年這類連假，則可以在前一週多進一些貨，或是直接根據前一週的需求預測結果進貨。由於 AI 商店是零售業，所以可採用補休的方式，讓員工在例假日上班。唯一要注意的是，要事先確定製造商能在例假日的時候接受訂單。
- 可先將上述的工作流程整理成文件，軟體工程師就能在機器學習工程師請病假的時候接受。也可以透過「結對編程」（Pair Programming）的方式分享建置推論環境與進行推論的步驟，讓軟體工程師了解整個工作流程。
- 如果無法在星期一取得實際銷售數據，或是聯絡工具突然故障，若能在星期二修復，就於星期二分享推論值，不然就是只能請各門市自行預測。
- 假設現有的機器學習模型無法正確預測，代表資料的傾向改變（就是所謂的資料飄移或是概念飄移），此時有可能得追加資料、採用不同的前置處理、微調參數或是重新開發模型。要是這麼做還是無法得到正確的推論結果，有可能得在開發新的機器學習模型之前，先沿用舊的模型，或是根據上一週的預測結果決定進貨量。
- 如果發生了嚴重的天災，當然要以人命為優先，而不是關心進貨量或是門市能否正常運作這些事情。