藉由 ChatGPT Plus 與 Gemini Pro 以 Vibe Coding 的方式，完成 Fantasy Baseball 預測模型

112011208 劉威志

一、 前言

近年來，生成式人工智慧的快速發展，使程式設計、資料分析與專題實作的學習方式產生明顯改變。過去若要完成一個完整的資料科學或機器學習專案，通常需要具備程式語言、資料前處理、統計分析、模型訓練與模型評估等多方面能力。然而，ChatGPT、Gemini 等生成式 AI 工具出現後，使用者可以透過自然語言描述需求，讓 AI 協助產生程式碼、解釋錯誤訊息、修改模型流程，甚至一步一步建立可執行的分析系統。這種以自然語言與 AI 協作完成程式開發的方式，常被稱為 Vibe Coding。本文所稱之 Vibe Coding，並不是指使用者完全不需要理解程式，而是指使用者透過與生成式 AI 反覆對話，將原本抽象的想法逐步轉化為可執行程式與可檢驗結果的開發模式。在這個過程中，AI 可以協助產生程式碼、提供除錯方向與整理分析流程，但使用者仍必須負責檢查模型邏輯、資料處理方式與結果合理性。因此，Vibe Coding 的核心不是「讓 AI 代替人完成所有工作」，而是形成一種人機協作的學習與實作過程。

本專題選擇以 Fantasy Baseball 作為實作主題是因為它同時結合個人興趣、棒球數據與機器學習應用。Fantasy Baseball 是一種以真實 MLB 球員表現為基礎的虛擬球隊經營遊戲。玩家會建立自己的球隊，而球員在真實比賽中的全壘打、打點、盜壘、打擊率、勝投、防禦率、三振數等表現，會影響玩家在遊戲中的積分與排名。也就是說，Fantasy Baseball 的核心問題，其實就是如何根據球員過去的成績，判斷其未來可能的表現與價值。然而，運動預測並不是單純將過去成績平均後就能得到準確答案。球員表現會受到年齡、傷病、出賽機會、球隊環境、打線位置、球場因素與角色變化等多種因素影響。以打者為例，OPS 較接近球員本身的攻擊能力，但 RBI 會受到隊友是否上壘與打線位置影響；以投手為例，ERA 與 WHIP 較能反映投手壓制能力，但勝投與救援成功數則高度依賴球隊戰力與投手角色。因此，本專題不只是要建立一個球員排行榜，更希望透過模型結果分析：哪些數據較容易從歷史表現中預測，哪些數據則受到現實環境高度干擾。

本報告的目標，是透過 ChatGPT Plus 與 Gemini Pro 的協助，以 Vibe Coding 的方式建立 MLB Fantasy Baseball 預測模型。模型使用 2015 至 2024 年 MLB 球員年度統計資料，並以過去三年的球員表現預測下一年的成績。打者模型主要預測 HR、RBI、SB、AVG、OPS；投手模型主要預測 W、SO、ERA、WHIP、SV。模型方法採用 XGBoost，並透過 time-based split、walk-forward validation、baseline comparison、MAE 與 Top-K Accuracy 檢驗模型表現。

透過這個專題，我希望呈現生成式 AI 不只是文字生成工具，也能成為資料科學學習與實作的輔助工具。然而，AI 的產出並不必然正確，使用者仍必須具備判斷、驗證與反思能力，才能避免模型表面上看似準確，實際上卻存在資料洩漏或方法錯誤的問題。

二、 運動機器學習模型與專有名詞說明

運動數據分析的目的，是透過球員或球隊的歷史表現，找出影響未來成績的可能因素。傳統上，球迷或分析者可能會直接比較打擊率、全壘打、防禦率、勝投等統計數據；但隨著資料量增加，單純依靠人工判斷或單一數據比較，已不容易完整掌握球員價值。因此，機器學習逐漸被應用在運動預測、球員評估與比賽分析中。機器學習是指讓電腦從歷史資料中學習規律，並利用這些規律預測未來結果的方法。以本專題為例，輸入資料可以是球員過去三年的 OPS、HR、PA、ERA、WHIP 等成績，預測目標則是下一年的 HR、RBI、OPS、ERA、WHIP 等表現。模型透過大量「過去表現對應下一年結果」的案例，學習哪些特徵組合通常會對應到較好的未來表現。

本專題建立的是監督式學習原型系統。所謂監督式學習是指模型訓練時同時具有輸入資料與答案資料。例如，模型以球員前三年的成績作為輸入，以下一年的真實成績作為答案。模型訓練完成後，再使用未見過的年份資料測試預測效果。這種方法適合本專題，因為 MLB 球員歷史資料具有明確年份順序，也具有明確的真實成績可供驗證。在模型選擇上，本專題使用 XGBoost 作為主要演算法。XGBoost，全名為 Extreme Gradient Boosting，是一種常用於表格型資料的機器學習方法。它能夠處理多個變數共同影響結果的情況，也能學習非線性關係。例如，打者下一年的全壘打數不一定只由過去全壘打數決定，也可能同時受到打席數、長

打能力、年齡與近期趨勢影響。XGBoost 的優點在於，它能從這些特徵中學習較複雜的規律，因此適合作為本專題的主要模型。

不過，機器學習模型能輸出預測結果，並不代表模型一定有效。尤其是運動預測這類時間序問題，若資料切分方式不正確，模型可能在訓練階段接觸到未來年份資料，進而產生資料洩漏。為避免此問題，本專題在後續模型驗證中採用依年份順序切分資料的方法，使模型評估更接近真實世界中「用過去預測未來」的情境。

三、 棒球統計指標說明

由於本專題的分析對象是 Fantasy Baseball 中常見的球員表現指標，因此在進入模型設計以前，有必要先說明各項棒球數據的意義。棒球數據大致可以分為「累積型數據」與「效率型數據」兩類。累積型數據主要反映球員在整個球季中累積出來的貢獻，通常會受到出賽場次與上場機會影響；效率型數據則較能反映球員在單位機會中的表現品質。

在打者數據方面：

- HR (Home Runs)：代表打者將球擊出全壘打牆外並直接得分的次數，通常用來衡量打者的長打火力。
- RBI (Runs Batted In)：代表打者透過安打、全壘打或其他進攻結果，使隊友或自己回到本壘得分的次數。RBI 雖然可以反映打者的得分貢獻，但它也會受到隊友是否上壘、打線位置與球隊整體攻擊能力影響。
- SB (Stolen Bases)：代表跑者成功盜上下一個壘包的次數，主要反映球員速度、跑壘能力與球隊戰術安排。
- AVG (Batting Average)：計算方式為安打數除以打數，用來衡量打者成功擊出安打的比例。
- OPS (On-base Plus Slugging)：由 OBP 上壘率與 SLG 長打率相加而成。相較於單純打擊率，OPS 同時考慮打者能否上壘與能否擊出長打，因此更能描述打者整體攻擊價值。

在投手數據方面：

- W (Wins)：代表投手在比賽中取得勝利紀錄的次數。然而，勝投不完全取決於投手自身能力，也會受到球隊打線支援、牛棚守成能力與比賽情境影響。
- SO (Strikeouts)：代表投手使打者三振出局的次數，通常可反映投手壓制打者與製造出局的能力。
- ERA (Earned Run Average)：表示投手每九局平均失去多少自責分，數值越低代表投手失分控制能力越好。
- WHIP (Walks plus Hits per Inning Pitched)：表示投手每投一局平均讓打者靠保送或安打上壘的次數，數值越低通常代表投手越能壓制對手上壘。
- SV (Saves)：代表後援投手在特定領先情境中成功守住勝利的次數。這項數據高度依賴投手是否擔任終結者，以及球隊是否經常進入可產生救援成功的比賽情境。

  除了上述傳統數據，本專題也加入若干衍生指標：
- ISO (Isolated Power)：由 SLG 減去 AVG 得出，用來衡量打者額外長打能力。
- BB% (Walk Rate)：代表打者每次打席中獲得保送的比例，可反映選球能力。
- K% (Strikeout Rate)：代表打者每次打席中被三振的比例，可用來觀察打者的揮棒風險與擊球穩定性。
- SO_per_IP：表示每局三振數，用來描述投手製造三振的能力。
- BB_per_IP：表示每局保送數，用來描述投手控制保送的能力。

這些數據之所以重要，是因為它們在 Fantasy Baseball 中扮演不同角色。HR、RBI、SB、W、SO、SV 等屬於累積型數據，通常與出賽機會、球隊環境與角色安排密切相關；AVG、OPS、ERA、WHIP、ISO、BB%、K% 等則較接近效率型或能力型指標，較能反映球員本身表現。因此，本專題後續分析模型結果時，也會特別區分「較接近球員自身能力的指標」與「較容易受到外部環境影響的指標」。這樣的區分有助於理解為什麼模型在 OPS、AVG、ERA、WHIP 等指標上較穩定，而在 RBI、W、SV 等指標上較難預測。

四、 資料來源與前處理

4.1 資料來源

這次專題中的數據使用 MLB 2015 年至 2024 年的球員年度統計資料，數據由 Baseball Reference 提供，並分成打者資料與投手資料兩部分。打者資料檔案命名為 hitters_2015.csv 至 hitters_2024.csv，投手資料檔案命名為 pitchers_2015.csv 至 pitchers_2024.csv。每一年都是獨立的 CSV 檔案，因此在訓練模型以前，需要先將不同年份資料合併成完整的歷史資料表。



圖 4-1：本專題的球員數據資料由 BASEBALL REFERENCE 提供[1]

打者資料主要包含 Player、Age、G、PA、HR、RBI、SB、BB、SO、BA、OBP、SLG、OPS 等欄位。其中 HR、RBI、SB 是 Fantasy Baseball 常見的累積型數據，AVG、OBP、SLG、OPS 則能反映打者的打擊品質。投手資料主要包含 Player、Age、W、ERA、G、GS、IP、H、ER、BB、SO、WHIP、SV 等欄位。其中 W、SO、SV 與 Fantasy Baseball 投手價值相關，而 ERA、WHIP 則較能反映投手的壓制能力。

類別	主要欄位	說明
打者資料	HR、RBI、SB、AVG、OPS、PA、G	火力、速度、打擊品質與出賽機會
投手資料	W、SO、ERA、WHIP、SV、IP、GS	勝場、三振、壓制能力與投手角色
衍生特徵	ISO、BB%、K%、SO/IP、BB/IP	用於描述效率與進階能力

表 4-1：資料欄位整理

[1] 圖片取自：https://www.baseball-reference.com。

4.2 資料合併流程

由於原始資料依年份分開儲存，模型不能直接使用單一年份資料。因此我先建立資料合併程式，將 2015 至 2024 年的打者資料合併成 historical_hitters.csv，並將投手資料合併成 historical_pitchers.csv。

合併流程大致如下：

1. 依年份讀取 CSV 檔案。
2. 選取模型需要的欄位。
3. 加入 Season 欄位標記年份。
4. 統一欄位名稱，例如將 BA 改為 AVG。
5. 合併所有年份資料。
6. 建立衍生特徵並輸出為新的歷史資料表。

4.3 衍生特徵

在資料前處理階段，本專題加入多項衍生特徵，使模型不只參考球員的累積成績，也能觀察其效率型表現。打者部分加入 ISO、BB%、K%、HR_per_PA、HR_per_600PA 等特徵；投手部分則加入 SO_per_IP 與 BB_per_IP，分別用來描述投手每局製造三振與控制保送的能力。這些特徵的目的，是降低單純累積數據受到出賽機會影響的問題。例如，兩位打者同樣擊出 30 支全壘打，但若其中一位只用了 450 個打席，另一位用了 650 個打席，兩人的長打效率其實並不相同。加入 HR_per_PA 或 HR_per_600PA 後，模型可以更清楚分辨這種差異，也能避免只用總量數據判斷球員能力。

特徵	計算方式	意義
ISO	SLG - AVG	純長打能力
BB%	BB / PA	保送率
K%	SO / PA	三振率
HR_per_600PA	HR / PA × 600	標準化全壘打能力
SO_per_IP	SO / IP	投手每局三振能力
BB_per_IP	BB / IP	投手每局保送比例

表 4-2：衍生特徵說明

4.4 投手 IP 欄位處理

投手資料中的 IP 需要特別處理，因為棒球的投球局數不是一般小數。例如 123.1 代表 123 又 1/3 局，而不是 123.1 局，如果直接把它當成一般小數，SO_per_IP 與 BB_per_IP 會計算錯誤，因此程式中加入 parse_ip()函式，將.1 轉換為 1/3，將.2 轉換為 2/3。這個細節對投手模型很重要，因為投球局數會影響三振率與保送率等效率特徵。

4.5 資料限制

目前資料處理仍有幾個限制。其中一個是：2020 年是 MLB 縮水賽季，球季場數與一般年份不同，會影響 PA、HR、RBI、IP、SO 等累積型數據。這也是後續模型 2021 年 walk-forward validation 中出現較大波動的重要原因之一。另外，模型目前主要使用歷史成績資料，尚未加入傷病、球場、球隊環境與打線位置等外部資料。因此，資料本身仍不足以完整描述球員未來表現的所有影響因素。

五、特徵工程與模型架構

5.1 三年滑動視窗

完成資料整理後，下一步是把年度球員資料轉換成模型可以學習的訓練樣本。這次專題採用「前三年預測下一年」的方式建立資料，也就是用球員目標年份前 3 年的表現作為輸入特徵，再用目標年份的真實成績作為答案。例如要預測 2024 年，模型輸入會使用球員 2021、2022、2023 年的資料；2024 年真實成績只在最後用來計算誤差，不會進入訓練。相同邏輯也會套用到其他年份，例如：

2016–2018 → 2019

2017–2019 → 2020

2018–2020 → 2021

2019–2021 → 2022

2020–2022 → 2023

表 5-1：三年滑動視窗範例

這種方法稱為三年滑動視窗。它的優點是能夠讓模型從不同年份區間中學習「過去三年表現與下一年結果」的關係，而不是只看單一年資料。雖然每一筆預測輸入只使用前三年資料，但整個模型會從多個年份窗口與多位球員的歷史案例中學習規律。

5.2 特徵設計

為了讓模型同時參考球員長期能力、近期狀態與變化方向，這次專題將輸入特徵分成三類：三年平均、最近一年與趨勢特徵。

第一類是三年平均特徵，例如 HR_3yr、OPS_3yr、PA_3yr、ERA_3yr、WHIP_3yr 等。這類特徵用來表示球員較穩定的長期表現，也能降低單一年份波動造成的影響。

第二類是最近一年特徵，例如 HR_latest、OPS_latest、ERA_latest、IP_latest 等。三年平均雖然穩定，但可能掩蓋最近狀態。例如兩位球員三年平均 HR 都是 30，但一位是逐年上升，另一位是逐年下降，未來預測結果不應完全相同。因此加入最近一年資料，可以讓模型掌握球員較接近目標年份的狀態。

第三類是趨勢特徵，例如 HR_trend、OPS_trend、PA_trend、ERA_trend、WHIP_trend 等。趨勢特徵使用「最近一年數據 − 三年前數據」計算，描述球員表現是上升還是下降。這能讓模型區分長期平均相同但變化方向不同的球員。

特徵類型	範例	目的
三年平均	HR_3yr、OPS_3yr、ERA_3yr	描述長期能力
最近一年	HR_latest、OPS_latest、IP_latest	描述近期狀態
趨勢特徵	HR_trend、OPS_trend、ERA_trend	描述成長或退化方向
效率特徵	HR_per_600PA、SO_per_IP	描述標準化後的能力

表 5-2：特徵設計整理

這樣設計的目的，是避免模型只依賴單純平均。三年平均本身是一個很強的 baseline，但若加入 latest 與 trend，模型就有機會學到單純平均無法捕捉的資訊。

5.3 XGBoost 與排名分數

這次專題使用 XGBoost 作為主要模型。XGBoost 是一種適合表格型資料的機器學習模型，能處理非線性關係。對運動數據來說，球員未來表現通常不是由單一因素決定，而是多個因素共同影響。例如打者下一年的全壘打數，可能同時受到過去 HR、ISO、PA、年齡與近期趨勢影響；投手下一年的 ERA 或 WHIP，也可能同時與過去壓制能力、投球局數與保送率有關。

由於模型需要同時預測多個目標，因此使用 MultiOutputRegressor。打者模型預測 HR、RBI、SB、AVG、OPS；投手模型預測 W、SO、ERA、WHIP、SV。模型訓練完成後，會輸出每位球員在目標年份的預測數值。

為了將預測數值轉換成 Fantasy Baseball 排名，這次專題另外設計 ranking score。打者部分使用 Power Score：

$$Power\ Score\ =\ z(HR)\ +\ z(RBI)\ +\ z(OPS)$$

HR、RBI 與 OPS 先經過 z-score 標準化後再相加，避免不同數值尺度造成影響。投手部分使用 Pitcher Score：

$$Pitcher\ Score\ =\ z(SO)\ +\ z(W)\ +\ z(SV)\ -\ z(ERA)\ -\ z(WHIP)$$

其中 SO、W、SV 越高越好，因此為正向加總；ERA 與 WHIP 越低越好，因此以負號處理。這兩個分數都不是 MLB 官方排名，而是依照 Fantasy Baseball 需求設計的自定義排名方式。流程如下：

$$歷史資料\ \rightarrow\ 三年特徵\ \rightarrow XGBoost \rightarrow\ 預測數值\ \rightarrow Ranking\ Score$$
$$\rightarrow 球員排名$$

5.4 模型設定與樣本篩選

為了使模型訓練流程更清楚，本專題將打者與投手分開建立模型。打者模型以 HR、RBI、SB、AVG、OPS 作為預測目標；投手模型則以 W、SO、ERA、WHIP、

SV 作為預測目標。由於每一類球員都需要同時預測多個數值，因此本專題使用 MultiOutputRegressor 搭配 XGBRegressor，讓模型能針對多個目標分別進行預測。

在樣本建立上，本專題採用「前三年預測下一年」的資料結構。若球員在前三年資料中缺少必要欄位，或無法形成完整三年滑動視窗，則不納入該年度訓練樣本。這樣的處理方式可以避免模型使用不完整資料進行訓練，降低因缺失值造成的預測誤差。此外，極少出賽的球員可能會造成數據波動。例如打者若打席數過少，HR_per_600PA、AVG、OPS 等效率型指標可能因樣本太小而失真；投手若投球局數過少，ERA、WHIP、SO_per_IP 等指標也容易出現極端值。因此，本專題在模型建立時會優先保留具有足夠出賽紀錄的球員，使模型更能學習穩定的球員表現模式。

本專題的模型仍定位為 supervised machine learning prototype，因此主要目標不是追求最佳化參數，而是建立一個邏輯正確、能避免資料洩漏，並可與 baseline 比較的預測流程。XGBoost 的主要可調參數包含 n_estimators、max_depth、learning_rate 等；目前尚未進行完整 hyperparameter tuning，未來若要提升模型表現，可進一步透過交叉驗證或 walk-forward validation 調整模型參數。完成模型設定與樣本篩選後，下一章將說明本專題如何使用 time-based split、blind test、walk-forward validation 與 baseline comparison 檢驗模型結果。

六、模型驗證方法

6.1 資料洩漏與時間序驗證

機器學習模型能輸出結果，不代表模型一定有效。尤其是運動預測這類時間序問題，如果模型在訓練時已經看過未來年份的資料，就會產生資料洩漏。資料洩漏會讓模型表面上看起來很準，但實際上並不具備真正預測能力。一開始在開發模型時，我也遇到這個問題。若使用一般機器學習常見的 random split，資料會被隨機分成訓練集與測試集。這種方法在一般資料分析中很常見，但不適合用在「用過去預測未來」的任務中。因為隨機切分可能會讓 2024 年的資料進入訓練集，接著又拿 2024 年當測試資料，這樣就會變成模型已經看過答案。因此與 AI 溝通後，後來模型改成使用 time-based split，也就是依照年份順序切分資料。

以 2024 年 blind test 為例，模型只使用 2024 年以前的資料訓練，2024 年真實成績完全不進入訓練，只在模型預測完成後用來計算誤差。

例如預測 2024 年時，模型訓練資料包含：

2015–2017 → 2018

2016–2018 → 2019

2017–2019 → 2020

2018–2020 → 2021

2019–2021 → 2022

2020–2022 → 2023

最後再使用：

輸入 2021–2023 → 預測 2024

這樣的設計可以確保模型是在真正模擬「用過去資料預測未來」的情境，而不是看過答案後再進行預測。除了 final blind test，這次專題也使用 Walk-forward Validation。Walk-forward Validation 的概念，是逐年模擬預測過程。例如測試 2020 年時，只使用 2020 年以前可形成的資料訓練；測試 2021 年時，只使用 2021 年以前的資料訓練。這樣可以觀察模型是否只是在某一年表現良好，還是在不同年份中都有一定穩定性。

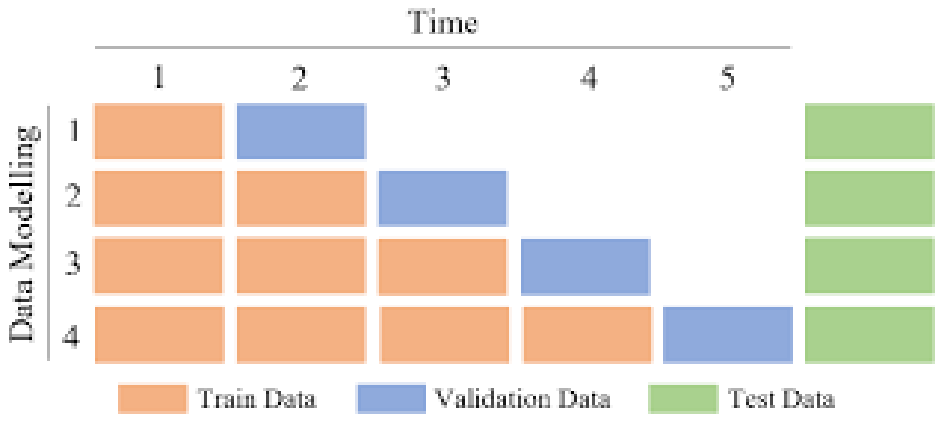


圖 6-1：Walk-forward Validation 示意圖[2]

[2] 圖片取自：https://audhiaprilliant.medium.com/walk-forward-optimization-cross-validation-technique-for-time-series-data-61739f58f2c0

6.2 Baseline、MAE 與 Top-K Accuracy

模型評估不能只看 XGBoost 的預測誤差，還需要有比較基準。因此這次專題使用「三年平均」作為 baseline。也就是說，除了讓 XGBoost 預測下一年成績，也同時計算「如果直接用過去三年平均當作預測值」會有多大的誤差。這個 baseline 雖然簡單但其實不弱。因為球員表現通常具有一定延續性，過去三年平均本身就能提供基本預測能力。如果 XGBoost 能比三年平均更準，才比較能說明模型確實學到了額外資訊，例如近期狀態、趨勢變化、出賽機會與效率特徵。數值預測的主要評估指標是 MAE（Mean Absolute Error，平均絕對誤差）。MAE 代表模型預測值與真實值之間的平均差距。例如 HR 的 MAE 為 5，代表模型平均約差 5 支全壘打。這次專題同時比較 XGBoost MAE 與 Baseline MAE，並計算 Improvement %：

$$Improvement\ \% = \frac{(Baseline\ MAE - XGBoost\ MAE)}{Baseline\ MAE} \times 100\%$$

除了 MAE，本專題也使用 Top-K Accuracy 評估排名效果。因為 Fantasy Baseball 不只在意每個數值預測差多少，也在意模型能不能找出前段球員。例如 Top 10 Accuracy 是指模型預測前 10 名球員中，有幾位也出現在真實前 10 名。



圖 6-2：模型評估流程圖

透過 time-based split、walk-forward validation、baseline comparison、MAE 與 Top-K Accuracy，模型結果不只是能輸出排行榜，而是能被檢查是否真的具有預測能力。下一章將根據這些評估方式，分析打者模型與投手模型的實際結果。

七、模型結果與分析

7.1 打者模型結果

打者模型在 2024 年 blind test 中，五項預測目標皆優於三年平均 baseline。因為 2024 年資料沒有進入模型訓練，所以這個結果可以視為模型在未知年份上的預測表現。

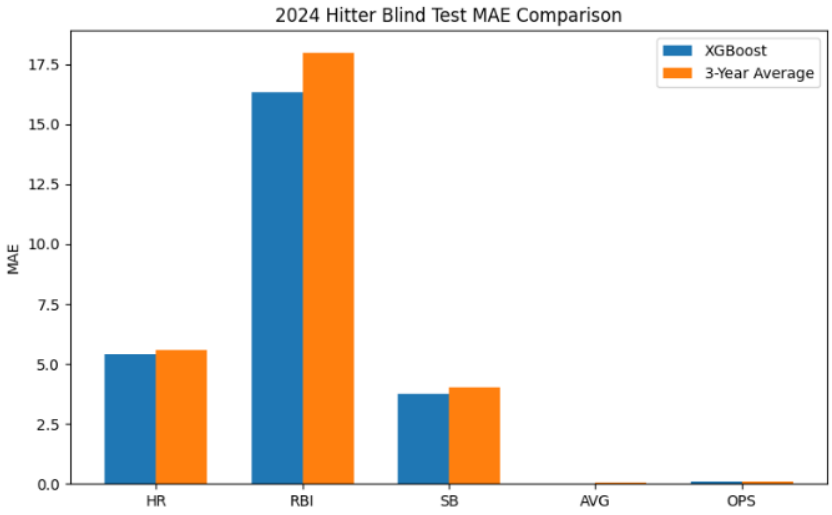


圖 7-1：2024 打者模型 MAE 比較[3]

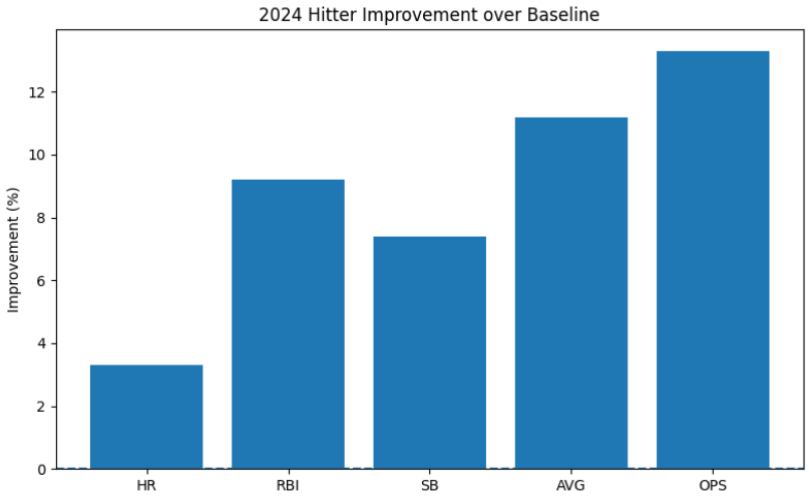


圖 7-2：2024 打者模型 Improvement %[4]

[3] 資料來源：作者依據 Baseball-Reference 歷史資料與本專題模型輸出自行繪製。
[4] 資料來源：作者依據 Baseball-Reference 歷史資料與本專題模型輸出自行繪製。

從 2024 blind test 來看，打者模型在 HR、RBI、SB、AVG、OPS 五項指標上都比三年平均更好。其中 OPS 改善約 13.3%，AVG 改善約 11.2%，RBI 改善約 9.2%。這代表模型不是單純重複三年平均，而是有利用 latest、trend、PA、ISO、BB%、K% 等特徵學到額外資訊。不過，若觀察 walk-forward validation，可以發現打者模型並不是每個指標都穩定。AVG 與 OPS 在多數年份中穩定優於 baseline，但 HR、RBI、SB 則較不穩定。

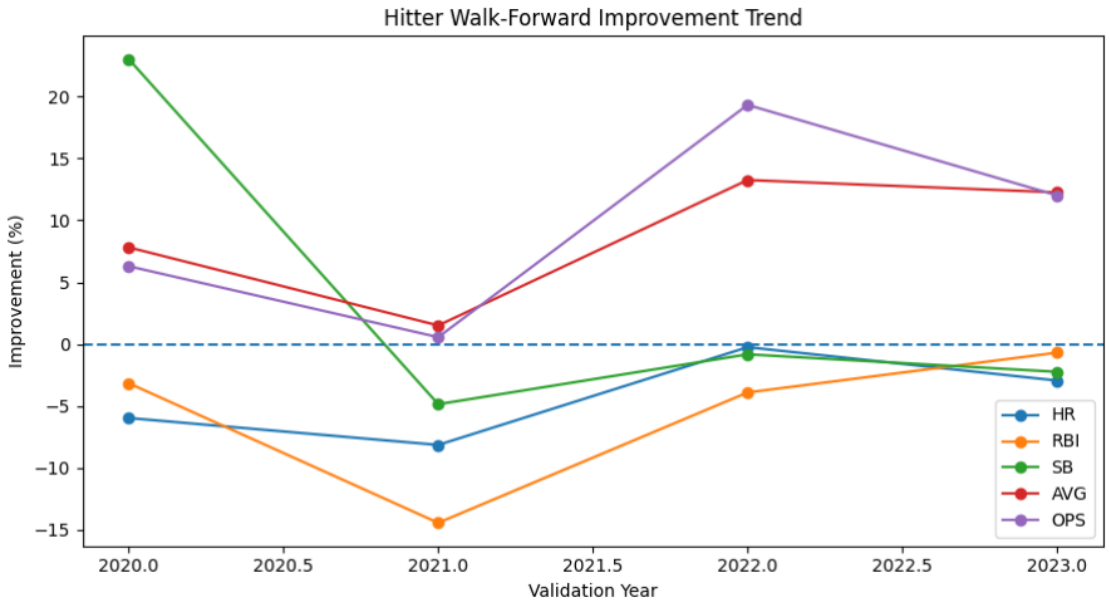


圖 7-3：打者模型 Walk-forward Improvement Trend[5]

這個結果其實符合棒球邏輯。AVG 與 OPS 較接近打者本身的打擊品質，因此模型比較容易透過歷史數據學習。相反地，HR、RBI、SB 雖然也是打者數據，但更容易受到出賽數、傷病、球場環境、球隊打線與戰術安排影響。特別是 RBI，很依賴隊友是否上壘與球員在打線中的位置，因此只靠球員自身歷史成績較難穩定預測。另外，圖中 2021 年附近出現明顯下滑，可能與 2020 年縮水賽季[6]有關。預測 2021 年時，模型輸入包含 2018、2019、2020 年資料，而 2020 年因球季場數縮短，PA、HR、RBI 等累積型數據偏低。當這些資料被當成輸入特徵時，容易讓模型低估球員在 2021 年恢復完整球季後的表現。在排名結果方面，打者模型的 Top 10 命中人數為 3/10，Top 30 命中人數為 19/30。這代表模型對最頂尖的 Top 10 排名仍不夠精準，但對前段球員群有一定辨識能力。

[5] 資料來源：作者依據 Baseball-Reference 歷史資料與本專題模型輸出自行繪製。

[6] 2020 MLB 縮水賽季是指受 COVID-19 疫情影響，原本每隊 162 場例行賽被大幅縮短為 60 場的特殊賽季，該季例行賽延後到 2020 年 7 月 23 日開打。

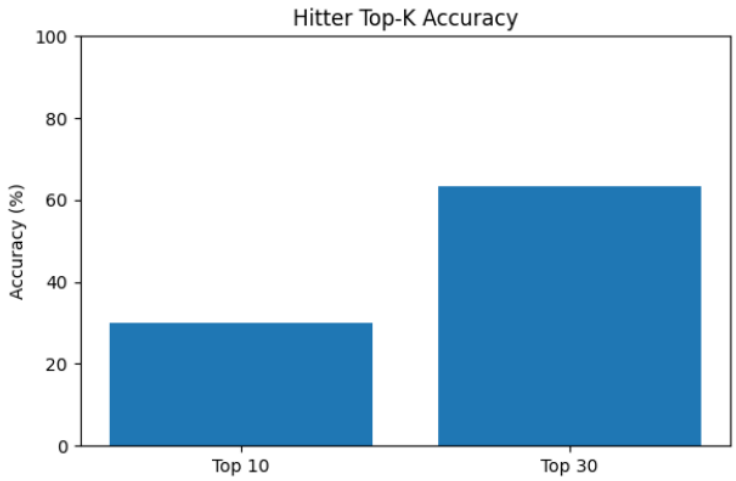


圖 7-4：打者模型 Top-K Accuracy[7]

因此，打者模型較適合用來判斷球員大致強弱與前段球員群，而不是直接精準決定 Top 10 排名。

7.2 投手模型結果

投手模型在 2024 年 blind test 中，五項指標中有四項優於 baseline，分別是 W、SO、ERA、WHIP；只有 SV 低於三年平均 baseline。

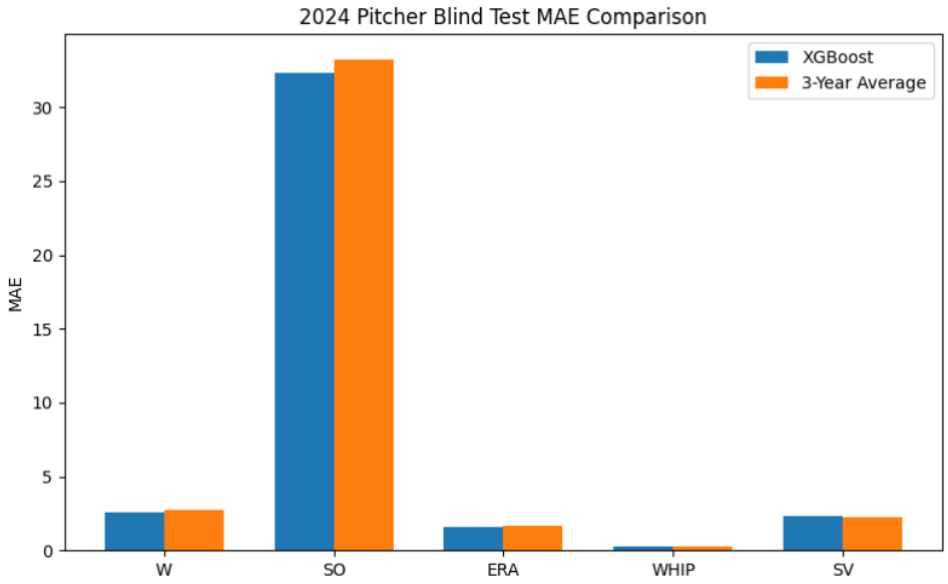


圖 7-5：2024 投手模型 MAE 比較[8]

[7] 資料來源：作者依據 Baseball-Reference 歷史資料與本專題模型輸出自行繪製。
[8] 資料來源：作者依據 Baseball-Reference 歷史資料與本專題模型輸出自行繪製。

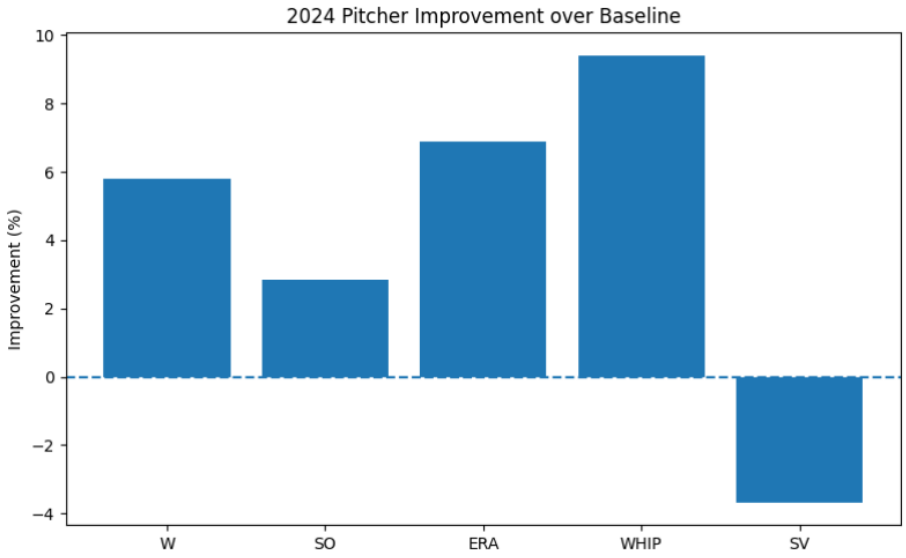


圖 7-6：2024 投手模型 Improvement %[9]

從結果來看，WHIP 改善約 9.4%，ERA 改善約 6.9%，W 改善約 5.8%，SO 改善約 2.9%。這代表模型對投手壓制能力相關指標有一定效果，尤其是 ERA 與 WHIP。投手模型的 walk-forward validation 也顯示類似趨勢。ERA 與 WHIP 在後期年份改善較明顯，W 有時有效，SO 接近 baseline，而 SV 最不穩定。

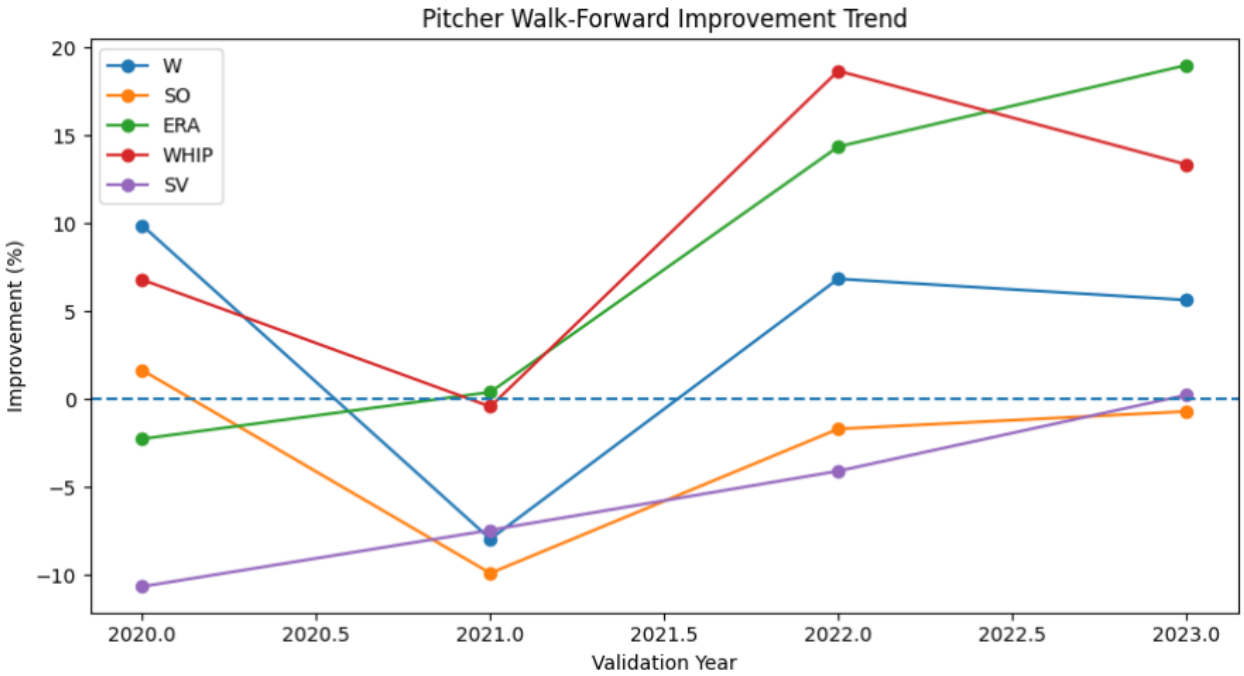


圖 7-7：投手模型 Walk-forward Improvement Trend[10]

[9] 資料來源：作者依據 Baseball-Reference 歷史資料與本專題模型輸出自行繪製。
[10] 圖片來源：作者依據 Baseball-Reference 歷史資料與本專題模型輸出自行繪製。

ERA 與 WHIP 較穩定，可能是因為這兩項指標較接近投手自身能力。ERA 反映失分控制能力，WHIP 反映每局讓打者上壘的情況。雖然它們仍會受到防守、球場與運氣影響，但相較於 W 與 SV，仍更接近投手本身表現。SO 的表現接近 baseline，主要是因為三振數同時受到三振能力與投球局數影響。如果投手受傷、先發轉後援，或球隊限制投球局數，即使三振能力不差，總三振數也可能大幅下降。SV 則更難預測，因為救援成功數高度依賴球隊牛棚安排與終結者角色，而這些資訊目前沒有放進模型。投手排名結果比打者更弱。Top 10 命中人數為 2/10，Top 30 命中人數為 11/30。

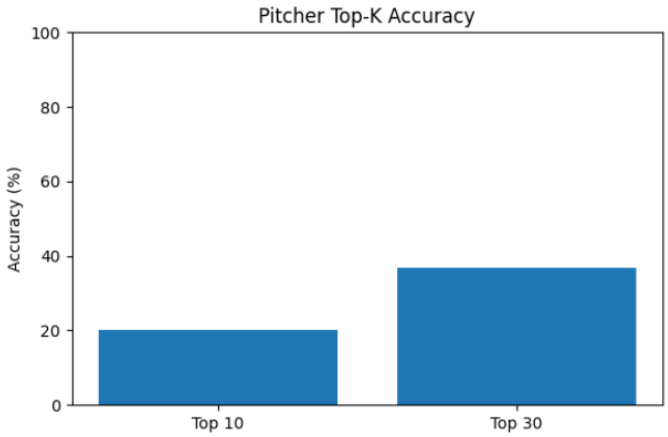


圖 7-8：投手模型 Top-K Accuracy[11]

這代表投手模型在 Fantasy ranking 上仍有較大限制。投手表現受傷病、角色與球隊環境影響很大，因此即使模型能改善 ERA、WHIP 等數值預測，也不一定能準確排出整體 fantasy 排名。

7.3 綜合討論

綜合打者與投手模型可以發現，本專題的模型較擅長預測接近球員自身能力的指標，例如打者的 AVG、OPS，以及投手的 ERA、WHIP。相對而言，HR、RBI、W、SV 等指標則較不穩定，顯示這些數據除了球員能力外，也受到出賽機會、球隊環境、打線位置、投手角色與傷病等因素影響。這個結果與前文對棒球統計指標的分類相互呼應：效率型或能力型指標較容易從歷史資料中學習，累積型或角色型指標則需要更多外部資訊輔助判斷。Feature importance 圖也可作為

[11] 圖片來源：作者依據 Baseball-Reference 歷史資料與本專題模型輸出自行繪製。

輔助解釋。若模型重視 OPS、ISO、PA、ERA、WHIP、SO_per_IP 等特徵，代表模型確實參考了具有棒球意義的變數，而不是隨機產生結果。目前這個模型可以被視為一個有效的機器學習 prototype，已具備正確的資料切分、合理的 baseline 比較以及可解釋的結果分析。

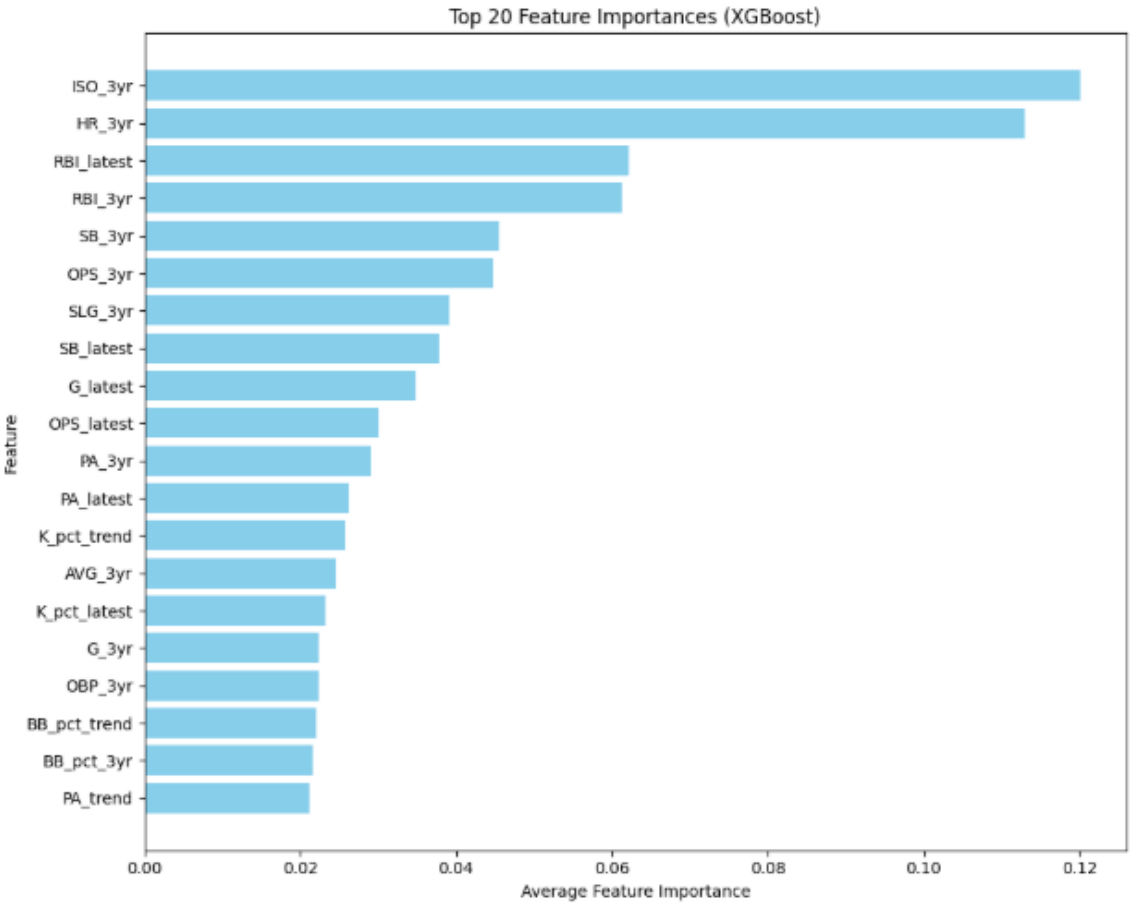


圖 7-9：Feature Importance[12]

八、模型限制與未來發展

8.1 目前限制

雖然這次專題已經完成打者與投手模型，並且使用時間序驗證、blind test、baseline comparison 與 Top-K Accuracy 進行評估，但目前模型仍然有幾個限制。第一，模型主要使用球員歷史成績，尚未加入外部資料。例如傷病狀況、球場因

[12] 圖片來源：作者依據 Baseball-Reference 歷史資料與本專題模型輸出自行繪製。

素、球隊強度、打線位置、投手角色與教練調度都沒有納入模型。這也是 HR、RBI、W、SV 等指標較不穩定的重要原因。這些數據不只由球員能力決定，也很依賴出賽機會與球隊環境。第二，2020 年是縮水賽季，會影響 PA、HR、RBI、IP、SO 等累積型數據。因為模型使用前三年資料預測下一年，所以當 2020 年資料進入輸入特徵時，可能會影響 2021 年的預測結果。這也解釋了 walk-forward trend 中 2021 年附近明顯下滑的現象。第三，Power Score 與 Pitcher Score 是依照 Fantasy Baseball 需求自行設計的排名分數，並不是 MLB 官方排名。因此，模型排名應該被理解為本專題定義下的 fantasy ranking，而不是官方球員實力排名。

8.2 未來發展方向

未來若要讓模型更完整，可以從資料、模型與應用三個方向改進：在資料方面，可以加入更多外部資訊。打者模型可以加入球場因素、打線位置、球隊得分能力與 Statcast 進階數據，例如 Barrel%、HardHit%、Exit Velocity。投手模型可以加入 K%、BB%、球速、球種、CSW% 等資料。這些資料會比傳統成績更直接反映球員能力；在模型方面，可以針對不同目標建立獨立模型。例如 HR、OPS、ERA、SV 的預測難度不同，若全部使用同一組模型架構，可能無法針對個別指標最佳化。未來也可以加入 hyperparameter tuning，嘗試尋找更適合的 XGBoost 參數；在應用方面，可以進一步建立 2026 Fantasy Baseball Projection。也就是使用 2023–2025 年資料預測 2026 年球員表現，並輸出未來排行榜。如果再加上自動更新資料與簡單網頁介面，就能發展成更完整的 Fantasy Baseball 分析工具。

九、AI 協作反思

這次專題最大的收穫，不只是完成 Fantasy Baseball 預測模型，而是更具體地理解生成式 AI 在學習與創作中的角色。過去若要從零開始完成一個資料科學專案，使用者往往需要先熟悉程式語言、資料處理、模型訓練與結果視覺化等技術細節；而在這次實作中，ChatGPT Plus 與 Gemini Pro 則扮演了輔助開發與學習引導的角色，協助我將原本較抽象的專題想法，逐步轉化為可以執行、可以驗證，也可以分析結果的模型流程。在實作過程中，AI 最明顯的幫助，是降低了進入專案的門檻。當我提出資料整理、特徵工程或模型訓練的需求時，AI 可以快速提供程式

架構與修改方向，也能協助解釋錯誤訊息，使我更容易理解每個步驟的目的。這讓我感受到，生成式 AI 不只是用來產生文字，也可以成為學習複雜技術時的輔助工具。尤其是在面對不熟悉的程式流程時，AI 可以幫助使用者先建立整體架構，再透過逐步修正理解細節。然而，這次經驗也讓我意識到，AI 並不能取代使用者的判斷。AI 產生的程式碼或模型流程看似完整，但仍可能存在邏輯問題。例如在早期版本中，資料切分方式曾經不夠嚴謹，使模型有可能在訓練時接觸到測試年份資料。若只看模型輸出的結果，可能會誤以為模型表現良好；但從預測邏輯來看，這種結果並不可靠。因此，使用 AI 完成專題時，使用者仍然需要理解問題本身，並檢查資料來源、模型假設與驗證方法是否合理。

因此，我認為 Vibe Coding 的價值並不在於讓人完全不用學習程式，而是在於讓使用者能更快進入實作，並在實作過程中反過來理解程式、資料與模型。AI 可以提供方向、工具與初步解法，但真正決定專題品質的，仍然是使用者是否能提出清楚問題、辨認錯誤、修正流程，並誠實面對模型的限制。從通識教育的角度來看，這次專題也讓我重新思考 AI 與學習之間的關係。AI 的出現並不代表學習變得不重要，而是使「判斷能力」變得更加重要。當工具能快速產生內容時，使用者更需要知道如何檢查、如何提問，以及如何判斷結果是否可信。也因此，本專題並不將 AI 視為單純的代寫工具，而是將其視為一種協作式學習工具。透過這次實作，我更能理解生成式 AI 在高等教育中的可能性：它可以幫助學生把想法轉化為作品，但同時也要求學生具備更高的驗證能力與反思能力。

十、結論

本專題透過 Vibe Coding 的方式，完成了一個 MLB Fantasy Baseball 機器學習預測模型。整個過程包含資料整理、棒球統計指標說明、特徵工程、XGBoost 模型訓練、walk-forward validation、blind test、baseline comparison、Top-K Accuracy 與結果分析。這個專題的重點不只是產生球員預測排名，而是嘗試建立一個邏輯完整、能被驗證，也能被解釋的資料科學實作流程。從模型結果來看，打者模型在 2024 年 blind test 中五項指標皆優於三年平均 baseline，其中 AVG 與 OPS 表現較穩定；投手模型則在 W、SO、ERA、WHIP 四項指標優於 baseline，但 SV 表現較弱。整體而言，模型較擅長預測接近球員自身能力的效率型指標，例如 AVG、OPS、ERA、WHIP；相對地，對於受球隊環境、角色安排與出賽機會影響較大的

數據，例如 RBI、W、SV，模型表現則較不穩定。這樣的結果也呼應了本報告前文對棒球數據的分類：並非所有棒球數據都具有相同的可預測性。部分指標較能反映球員自身能力，因此較容易從歷史資料中學習；但部分指標則高度依賴外部環境，因此僅靠球員歷史成績仍不足以做出穩定預測。也因此，本專題所建立的模型應被視為一個 supervised machine learning prototype，而不是完整的商業級預測系統。本專題成功將生成式 AI 協作、棒球數據分析與機器學習模型結合，展示了 Vibe Coding 在資料科學實作中的應用可能。雖然模型仍有資料限制與預測誤差，但它已經呈現出清楚的研究流程、合理的驗證方法與可解釋的分析結果。未來若能加入傷病資訊、球場因素、打線位置、投手角色與 Statcast 進階數據，模型將有機會進一步提升預測能力，並發展成更完整的 Fantasy Baseball 分析工具。

九、參考資料

[1] Baseball-Reference.com. (n.d.). Baseball-Reference.com: MLB stats, scores, history, & records. Sports Reference. Retrieved June 3, 2026, from https://www.baseball-reference.com/

[2] Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 785–794. https://arxiv.org/abs/1603.02754

[3] MLB.com. (n.d.). Statcast glossary: Barrel. Major League Baseball. Retrieved June 3, 2026, from https://www.mlb.com/glossary/statcast/barrel

[4] scikit-learn developers. (n.d.). MultiOutputRegressor. scikit-learn. Retrieved June 3, 2026,from https://scikit-learn.org/stable/modules/generated/sklearn.multioutput.MultiOutputRegressor.html

[5] XGBoost developers. (n.d.). XGBoost documentation. XGBoost. Retrieved June 3, 2026, from https://xgboost.readthedocs.io/

[6] Aprilliant, A. (2021, September 15). Walk-forward optimization — Cross-validation technique for time-series data. Medium. https://audhiaprilliant.medium.com/walk-forward-optimization-cross-validation-technique-for-time-series-data-61739f58f2c0

[7] AI 使用聲明：本報告於程式設計、除錯、資料分析流程整理與文字檢查過程中，曾使用 ChatGPT 與 Gemini 作為輔助工具。所有研究主題、資料判讀、模型結果分析與報告內容均由作者理解後整理完成，並已由作者自行檢查其合理性與正確性。