

動靜之間學 AI

李柏翰

臺灣師大附中物理科教師、臺北科大光電工程兼任助理教授

前言

當 ChatGPT 在 2022 年 12 月發表時，頓時吸引大眾的眼球，視覺饗宴的震撼同步出現，原來人工智慧 (artificial intelligence, AI) 和大型語言模型 (large language model, LLM) 不再是科幻片的內容，而是輔助生活的利器。舉凡生活聊天、查找資料 AI 都能對答如流，一時之間 ChatGPT 成為當紅炸子雞。OpenAI 最先進的 AI 模型也能激起千堆雪，引領許多話題，像是具備挑戰數學奧林匹亞銀牌的解題實力、具備撰寫學術論文的能力、甚至是量子電腦的計算，又或者是 Meta 公司推出的 LLaMA 開源大型語言模型 (Code Llama 70B 模型)，其具有 700 億個參數，並允許開源下載，如此大量的記憶內容與模型，驅使更多公司投入 LLM 的研發。2022 年之後，OpenAI 持續擴大資料庫，從早期 15 億個參數的 GPT-2 進化到 GPT-3 的 1,750 億個參數，並持續更新版本；Google 發行的 Gemma 3 270M，備有 2.7 億個參數，和其他模型相比更為親民，能在手機上執行；中國大陸最新的 DeepSeek R1 最高則是來到了 6,710 億個參數，激起一池水春水！自此之後，百家爭鳴的 AI 盛世，無不引起各方面的高度關注。

不得不說，人工智慧不再是科技愛好者的興趣，而且是已經變成當代的顯學，2024 年諾貝爾物理學獎就是頒給了約翰·霍普菲爾德 (John Hopfield) 和傑弗里·辛頓 (Geoffrey Hinton) 在人工神經網路方面與 AI 科研奠基性工作。AI 的應用遍地開花，從小遊戲、辨識物件到公車上提醒駕駛與其他車輛距離的 YOLOv4、預測房價漲跌的模型，都可以看到 AI 演算法的無限身影。不過，雖然其應用已相當廣泛，但是新手在學習上，由於 AI 涉及的內容包羅萬象，從資料預處理、模型訓練、演算法選擇，到程式語言的使用與調整，每一環節都可能讓人感到眼花撩亂，難以上手。因此，本文介紹一些動態和靜態的應用及自主學習 AI 的一些門路和資源，讓有志於研究 AI 朋友們可以較容易上手些。

在動態 AI 的應用中，最精彩吸睛的莫過於遊戲了，其中剪刀石頭布更是家喻戶曉，小朋友都會玩的遊戲。因此，結合機器學習 (machine learning, ML)、數學及 Python 程式語言來製作一個和 AI 玩剪刀石頭布的遊戲，並從中學習 AI 是一個非常有趣的學習過程。利用 AI，透過數學分析玩家過去的出拳模式，並預測下一步出手機率，讓 AI 學習人類的猜拳的規律，並且戰勝使用者。此遊戲發想能幫助初學者理解

Python 程式設計、AI 動態學習和矩陣的數學計算概念。

另一方面，奠定 AI 基礎的根本之法也包含靜態演算法的學習。因此，此點在學校探究與實作課程開發過，著力於機器學習的不同演算法、利用深度學習理論和應用，並討論 AI 和生活結合的任何可能性，實作中學習 AI。一個不動產價值預測模型就是應用機器學習生活化的例子，透過臺北市房價的數據分析理解決策樹（decision tree）的應用，及使用集成學習（ensemble learning）的兩大經典模型：隨機森林（random forest）與 XGBoost 演算法，使用不同模型、參數、訓練資料，比較訓練時間及預測準確度的差異。此應用除了展現機器學習在商業中的基本應用外，也可以幫助新手學員理解 AI 「分包」和機器學習的概念。

另外一個生活化的應用是圖像辨識系統，我們發想了一個人行道地磚破損辨識系統，利用實際採集雙北共 3,131 片地磚作為資料庫，並使用深度學習（deep learning）的卷積神經網路（convolutional neural network, CNN）和 YOLO（you only look once）演算法，來比較不同模型、路段相對應的圖像辨識準確度，辨識破損和正常地磚的分析。此點子是除了展示基本深度學習的應用外，也能讓新手理解神經網路的結構和理論實踐所需的態度。

再者，筆者在教授探究與實作課程時曾提供自建的教學網站 Buzz Note（<https://sites.google.com/view/buzznote/p5/p56>），曾經開發過八週的課程協助學員進行 AI 和跨領域的學習，這些都是開放的網路資源讓學員有更多的學習 AI 的機會。初學者可以使用現有的演算法，搭配自己的創意去嘗試建立新的資料庫，新的執行 CODE，此舉也是一個可從中發現 AI 運行的困難的地方，提早學習與應對的不錯機會。

動態小遊戲

構想之一，因為剪刀石頭布是一個小朋友常玩的遊戲，甚至大人都樂此不疲。而正是這個小遊戲，如果可以使用 AI 演算法呈現，讓大家用不同的方式來玩遊戲，應該是很不錯的發想，其邏輯如下：AI 從玩家的出拳學習，預測下一局玩家最可能出的拳，然後「想出」能戰勝玩家的出拳。在初期，AI 因尚未累積足夠的出拳資料，因此玩家可以輕鬆戰勝它。然而，隨著遊戲的進行，AI 開始記錄玩家的出拳序列資訊，並根據這些資料建立轉移矩陣計算。這個矩陣記錄了每一種手勢之間的轉移機率，也就是在玩家出某個手勢後，下一步最可能出現的手勢比例。舉例來說，如果玩家在出「剪刀」之後常常接著出「布」，那麼 AI 便會在玩家出剪刀後預測下一步是布，並選擇出剪刀來贏得比賽。

這種預測方式的核心是利用馬可夫鏈 (Markov chain) 的數學性質：也就是系統的下一步狀態僅取決於目前的狀態，而與更久遠的歷史無關，當前狀況和歷史是相互獨立的。在此遊戲中，玩家的每一次出拳都被視為一個狀態，AI 根據目前狀態與轉移機率來預測下一步，並做出最有利的選擇。以剪刀石頭布為例，上述狀態即為玩家過去的出拳序列，過渡機率則是在序列之下，預測出下一步各手勢的出現的可能性。AI 透過記錄「玩家歷史手勢」，建立出轉移矩陣，再預測下一步可能的出拳。這就是馬可夫鏈的精神，在此遊戲的原理和邏輯如下：

1. 先定義三個動作集合

$$S = \{\text{石頭 (R)}, \text{布 (P)}, \text{剪刀 (S)}\}$$

2. 使用馬可夫鏈的轉移機率公式

$$P(X_{n+1} = s_j \mid X_n = s_i) = P_{ij} \quad (1)$$

3. 使用轉移矩陣表示之

$$P = \begin{bmatrix} R \rightarrow R & R \rightarrow P & R \rightarrow S \\ P \rightarrow R & P \rightarrow P & P \rightarrow S \\ S \rightarrow R & S \rightarrow P & S \rightarrow S \end{bmatrix} \quad (2)$$

4. 在每回合，AI 根據最近 k 步的歷史狀態 (例如 k=2) 進行計算

$$P(\text{下一步} = S_j \mid S_{n-k+1}, \dots, S_n) = \frac{\text{次數}(S_{n-k+1}, \dots, S_n \rightarrow S_j)}{\sum_j \text{次數}(S_{n-k+1}, \dots, S_n \rightarrow S_j)} \quad (3)$$

5. 預測對手最可能的出拳

假設有一位玩家最近 5 次出拳如下：剪刀、石頭、布、剪刀、石頭，可以使用上述的轉移矩陣表示為：

$$P = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 2 & 0 & 0 \end{bmatrix} \quad (4)$$

又最近兩次出拳序列為：剪刀、石頭，故查看矩陣，石頭轉移成布的機率最大 ($R \rightarrow P=1$ ，其他石頭的轉移機率均為 0)，所以 AI 判定下一次玩家出布，故出剪刀，嘗試戰勝玩家，此法取決於多次對戰經驗之累計。

除此之外，Python 的程式實作也是此活動的重點之一，學生需要將看似直觀的事件用電腦的方式思考並呈現，code 如附錄。例如：和人類出剪刀石頭布時，兩人是在同時心裡博弈與出拳的，那要如何用 Python 呈現？就要透過設計函式去解決「思考」，也就是建立屬於此動作的馬可夫鏈和轉移矩陣；以及「出拳」決策，也就是定義剪刀石頭布的動作是要如何在應用程式顯示和運作。

學員除了可以學習馬可夫鏈的性質、轉移矩陣的數學問題和實作 Python 程式之外，也可實驗一下當作玩家的思考方式，例如：學生嘗試完全隨機出拳或著是認真思

考過程，記錄每一步後推測 AI 的下一步並反制 AI 等不同的玩法，這些策略可以訓練學生的邏輯思維與分析能力，也能鼓勵他們進行創新的思考與策略規劃。

圖 1

活動海報——挑戰 AI：AI 用馬可夫鏈打敗人類的剪刀石頭布

挑戰AI：AI用馬可夫鏈打敗人類的剪刀石頭布

剪刀石頭布

師大附中: 蔡昕翰 / 指導老師: 李柏翰
leepohan@gmail.com

「教育部補助中小學校人工智慧教育計畫」

AI的核心-馬可夫鏈

遊戲規則：石頭贏剪刀、剪刀贏布、布贏石頭。兩方同時出拳，比較勝負。

理論上：隨機選擇三者，勝率應為 50% 平手、25% 贏、25% 輸。

馬可夫鏈是一種隨機模型，下一步狀態僅取決於目前狀態，而與更久遠的歷史無關。

應用在剪刀石頭布：
狀態 = 玩家過去的出拳序列（例如過去 2 步）
過渡機率 = 在這個序列下，下一步各手勢的出現次數比例。
AI 透過記錄「人類歷史手勢」，建立轉移矩陣，預測下一步最可能的出拳。

為什麼人類會輸？

我們有潛在的習慣模式（例如避免連續出相同手勢、或是輸後換手）。

AI 累積次數越多，預測準確率越高。

當回合數足夠，AI 建立的轉移矩陣逐漸穩定，幾乎能精準預測人類下一步。

AI 無情且持續學習，不會情緒化，只依照數據推論最佳手勢。

馬可夫鏈數學原理


定義遊戲三個狀態集合：
 $S = \{\text{石頭}(R), \text{布}(P), \text{剪刀}(S)\}$

馬可夫鏈的轉移機率公式為：
 $P(X_{n+1} = s_j | X_n = s_i) = P_{ij}$


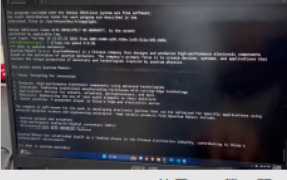
轉移矩陣可表示為：
$$P = \begin{bmatrix} R \rightarrow R & R \rightarrow P & R \rightarrow S \\ P \rightarrow R & P \rightarrow P & P \rightarrow S \\ S \rightarrow R & S \rightarrow P & S \rightarrow S \end{bmatrix}$$

在每回合，AI 根據最近 k 步歷史狀態（例如 $k = 2$ ）計算：
$$P(\text{下一步} = s_j | s_{n-k+1}, \dots, s_n) = \frac{\text{次數}(s_{n-k+1}, \dots, s_n \rightarrow s_j)}{\sum_j \text{次數}(s_{n-k+1}, \dots, s_n \rightarrow s_j)}$$

就能預測出對手下一步最有可能出的拳



省電的隨身聊天機器人





^ Raspberry PI 5套件 ^ 使用qwen2.5回答量子力學問題

架設非常容易
只需安裝ollama

開啟模型:
ollama run <模型名>

結束:
/bye



Buzz note: AI 教學 <https://sites.google.com/view/buzznote/p5/p56>

圖 2
剪刀石頭布玩到後面，發現 AI 已經遙遙領先



圖 3
挑戰此遊戲剪刀石頭布的民眾大排長龍

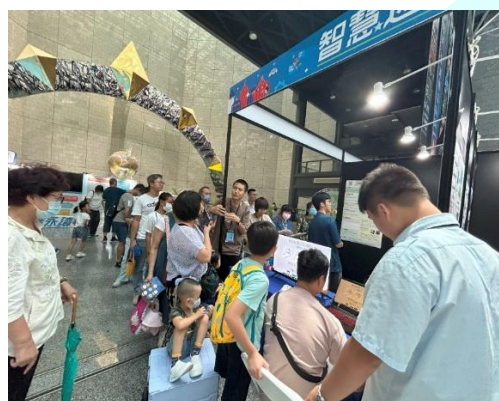
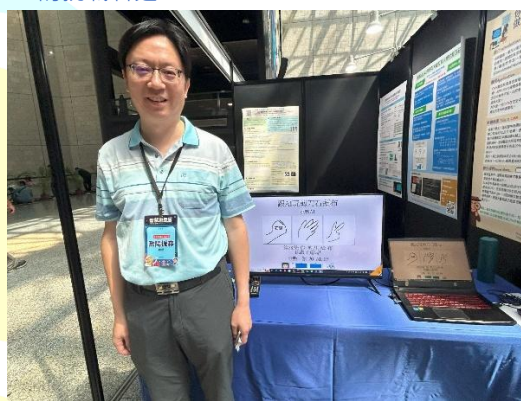


圖 4
歐陽振森教授是本次遊戲規則下，極少數能戰勝 AI 的挑戰者之一



此遊戲第一次在今年（2025）的 8 月 10 日，在高雄國立科學工藝博物館「智慧遊樂園：AI 奇幻之旅」公開試玩後廣受好評，參展的民眾大排長龍來玩此遊戲，甚至有人玩過後被吸引，還邀請親朋好友絞盡腦汁想嘗試征服 AI，其中亦包括智慧遊樂園的工作人員之一：例如來自高雄科大的歐陽振森教授。歐陽教授和其他約五位參與民眾是極少數能夠在先贏 20 場獲勝的要求下戰勝 AI 的人物，歐陽教授每下一步都思考很久，終於克服了馬可夫鏈的數學預測。其餘約百人挑戰 AI 均告失敗，很可能是被 AI 運算出其出拳規律，使其在之後的比賽中落入下風（見圖 1-4）。

從整體來看，AI 版本的剪刀石頭布能讓初學者了解動態的學習方式、AI 背後的數學邏輯和人機互動，是個非常值得學習實作的活動。

探究與實作課程，培養學生的 AI 素養

一、探究與實作課程

本校（臺灣師大附中）探究與實作課程設計之一，是從人工智慧的基礎出發，練習 AI 小專題，課程是從介紹 AI 的學習方式、監督與非監督式學習的種類、演算法和原理到用 Python 程式語言 AI 實作來進行，例如：鳶尾花資料集分類，MNIST 手寫數字辨識等，學習如何進行資料預處理、模型訓練與預測評估。在最後的專題實作中，學生掌握了機器學習與深度學習的基本技巧後，開始思考如何將 AI 應用於日常生活，如發想出：健康與不健康番茄葉的辨識，甚至進一步延伸專題，嘗試 AI 辨識於生活中的所面臨的問題，例如人行道地磚破損辨識、鳥類聲音的辨識，甚至是用相關演算法做房地產房價的推測等等。

再者，此課程除了培養學生理解和應用人工智慧的「硬實力」之外，也注重學生自己發現並正確使用工具的「軟實力」。因此在教授 AI 課程時，除了讓學生熟悉課程本會使用之軟體或網頁，如：Anaconda, Google Colab 等軟體之外，也會提供自學的方法和資料，如李柏翰老師的 Buzz Note AI 教學（見圖 5）。

圖 5

Buzz Note 自學網站提供許多領域的自學資源



註：本站網址 <https://sites.google.com/view/buzznote/p5/p56>

筆者自己建立的 Buzz Note 網站，除了介紹專業的高中物理學知識重點整理，及幫助學生系統性地解題，提升基本觀念與物理各單元完整題庫之外，也提供在眾多領域的研究成果讓學生參考。舉凡科普節目：科教館 GO 好玩的【巴斯無敵】節目紀錄檔，透過寓教於樂的方式，讓小朋友與青少年在輕鬆的氛圍中探索科學的奧妙，激發學習興趣。探究與實作的 AI 學習與程式語言，能夠讓高中生複習或自我進修，甚至對於有心挑戰更高層次的學生，網站亦有微分方程、工程數學等進階內容，讓大學新鮮

人或資優生能搶先建立數學基礎。最後，筆者也將自己在量子力學、材料與計算物理研究領域的論文公開，供研究生或有興趣者自行取經。Buzz Note 是理工科學生的小幫手，讓不同學習階段的學生在其中找到適合自己的學習資源，進行自主學習。

透過課程安排，學生能夠循序漸進的學習 AI，了解其發展歷史、基本應用 Python 和一些軟體，如 Matplotlib、Pandas 等來撰寫 AI 程式。此課程不僅培養了跨領域的思考能力，也激發了學員創造力與探索生活的熱情，體會 AI 不只是熱門話題，更是在生活中實踐的強大工具。接下來介紹兩位學生根據探究與實作課程發想的小專題，以「以機器學習建構不動產價值預測模型」和「你是我的眼：CNN 如何做人工智慧的『眼睛』」為題去分別實作機器學習和深度學習的演算法。

二、以機器學習建構不動產價值預測模型

房價議題是當前臺灣社會中民眾高度關注的焦點，尤其對一些年輕人而言，房價的波動不僅影響購屋意願，更會牽動生活規劃與發展。因此，如何有效預測房價走勢，成為值得研究的方向。此專題即以臺北市地區為研究範疇，運用機器學習中的集成學習方法，嘗試建立一套能夠準確預測房價變化的模型，並進一步探討不同模型與參數設定對預測結果的影響（周楷蓉、李柏翰，2025）（見圖 6）。

集成學習（ensemble method）是一種元學習演算法（meta learning algorithms），結合數種機器學習計算，建立一個單一的預測模型，其主要原則是將幾個較弱的模型（weaker models）組合在一起，形成一個能夠做出更準確預測的單一模型（strong model），其中此專題採用隨機森林（random forest）和 XGBoost，主旨在討論使用不同模型、參數、訓練資料，比較訓練時間長短和預測精度的差異，觀察模型變化對資料敏感度的差異，流程見圖 7。

圖 6
活動海報——以機器學習建構不動產價值預測模型

以機器學習建構不動產價值預測模型

師大附中：周楷蓉 / 指導老師：李柏翰 (leepohan@gmail.com)

「教育部補助中小學校人工智慧教育計畫」A 類計畫

AI 特色課程介紹

- 課程主題：用 AI 預測台北市房價——探索隨機森林與 XGBoost 的奧秘
- 透過真實數據分析（台北市房價），理解決策樹的應用，以及使用「集成學習」的兩大經典模型：隨機森林 (Random Forest) 與 XGBoost。
- 嘗試使用不同模型、參數、訓練資料，比較訓練時長以及預測準確度。



圖 1：研究方法

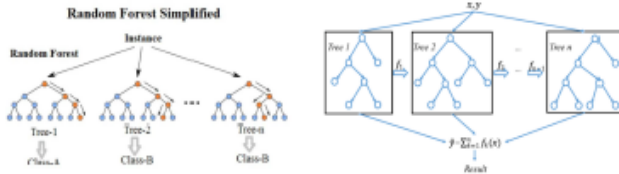


圖 2、3：隨機森林與 XGBoost 的模型架構

參數	範圍	最佳參數
'n_estimators'	[100, 200, 300, 400, 500]	400
'max_features'	[0.2, 0.4, 0.6, 0.8]	0.6
'max_samples'	[0.2, 0.4, 0.6, 0.8]	0.6
'criteria'	['mse', 'mae']	'mae'

圖 4：使用 GridSearchCV 找模型的最佳參數組合範圍的結果。

安排活動 / 遊戲說明

- 到台北市地政雲下載 2021 年 10 月至 2023 年 10 月台北市地區不動產交易公開資料。
- 對資料進行整理與視覺化處理。
- 使用不同模型對資料集進行訓練。
- 分析隨機森林與 XGBoost 對台北市地區未來不動產價格預估的有效性。
- 探討不同變因對不動產價格預估的影響。



台北市地政雲連結 (左)
程式碼導讀資料夾 (右)

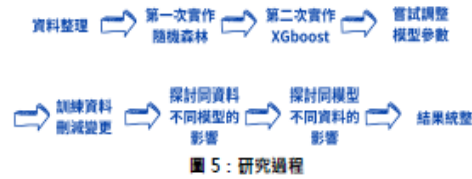


圖 5：研究過程



圖 6：訓練資料視覺化

AI 與數位音樂的交匯時刻 (2025)
主講：林育誼 (音樂總監)
日期：2025 年 4 月 11 日
觀看連結：YouTube



```

1 [10] True  0.346204813104  0.0011  0.250000
2  = 0.0011  True
3  = 0.0011  True
4  = 0.0011  True
5  = 0.0011  True
6  = 0.0011  True
7  = 0.0011  True
8  = 0.0011  True
9  = 0.0011  True
10 = 0.0011  True
11 = 0.0011  True
12 = 0.0011  True
13 = 0.0011  True
14 = 0.0011  True
15 = 0.0011  True
16 = 0.0011  True
17 = 0.0011  True
18 = 0.0011  True
19 = 0.0011  True
20 = 0.0011  True
21 = 0.0011  True
22 = 0.0011  True
23 = 0.0011  True
24 = 0.0011  True
25 = 0.0011  True
26 = 0.0011  True
27 = 0.0011  True
28 = 0.0011  True
29 = 0.0011  True
30 = 0.0011  True
31 = 0.0011  True
32 = 0.0011  True
33 = 0.0011  True
34 = 0.0011  True
35 = 0.0011  True
36 = 0.0011  True
37 = 0.0011  True
38 = 0.0011  True
39 = 0.0011  True
40 = 0.0011  True
41 = 0.0011  True
42 = 0.0011  True
43 = 0.0011  True
44 = 0.0011  True
45 = 0.0011  True
46 = 0.0011  True
47 = 0.0011  True
48 = 0.0011  True
49 = 0.0011  True
50 = 0.0011  True
51 = 0.0011  True
52 = 0.0011  True
53 = 0.0011  True
54 = 0.0011  True
55 = 0.0011  True
56 = 0.0011  True
57 = 0.0011  True
58 = 0.0011  True
59 = 0.0011  True
60 = 0.0011  True
61 = 0.0011  True
62 = 0.0011  True
63 = 0.0011  True
64 = 0.0011  True
65 = 0.0011  True
66 = 0.0011  True
67 = 0.0011  True
68 = 0.0011  True
69 = 0.0011  True
70 = 0.0011  True
71 = 0.0011  True
72 = 0.0011  True
73 = 0.0011  True
74 = 0.0011  True
75 = 0.0011  True
76 = 0.0011  True
77 = 0.0011  True
78 = 0.0011  True
79 = 0.0011  True
80 = 0.0011  True
81 = 0.0011  True
82 = 0.0011  True
83 = 0.0011  True
84 = 0.0011  True
85 = 0.0011  True
86 = 0.0011  True
87 = 0.0011  True
88 = 0.0011  True
89 = 0.0011  True
90 = 0.0011  True
91 = 0.0011  True
92 = 0.0011  True
93 = 0.0011  True
94 = 0.0011  True
95 = 0.0011  True
96 = 0.0011  True
97 = 0.0011  True
98 = 0.0011  True
99 = 0.0011  True
100 = 0.0011  True
    
```

圖 7：模型訓練結果與評估

學校辦理過的 AI 推廣活動

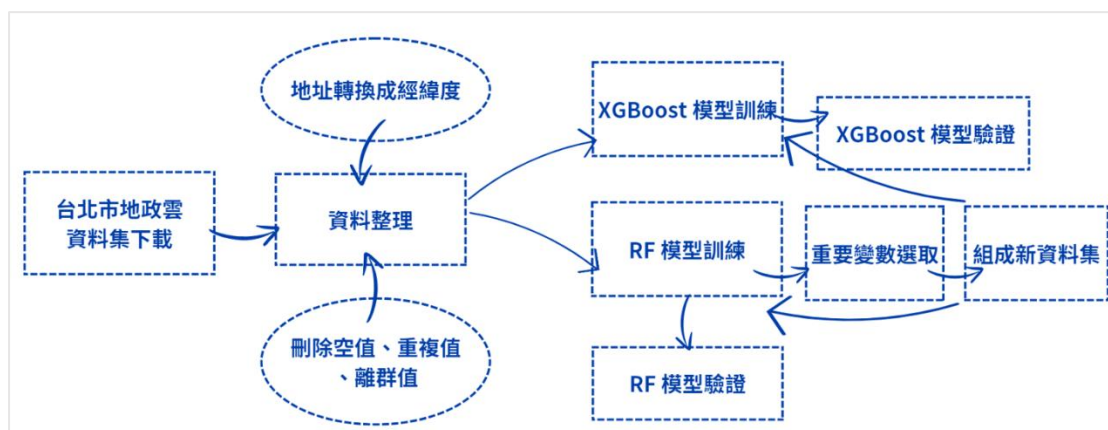
- 在多元選修課堂上邀請 AI 領域專家到校演講，分享專業領域與研究成果。
- 校內專題課成果發表，邀請學弟妹與家長到場聆聽。
- 在探究與實作課程中加入基礎機器學習模型理論，鼓勵學生進行相關領域專題探究。



圖 8：校內專題研究成果發表

圖 7

以機器學習建構不動產價值預測模型的研究流程



註：圖中 RF 指隨機森林 (random forest, RF) 。

此機器學習採用臺北市地政雲下載之 2021 年 10 月至 2023 年 10 月臺北市地區不動產交易公開資料進行訓練，再調整模型的部分參數、變更訓練資料，進而分析隨機森林與 XGBoost 對臺北市地區未來不動產價格預估的有效性，同時探討不同變因對不動產價格預估的影響。訓練完成後再將成果視覺化，其成果詳如圖 8-10，這些成果反映出同學已經能掌握 AI 學習的關鍵步驟。

圖 8

行政區和房價的圖表

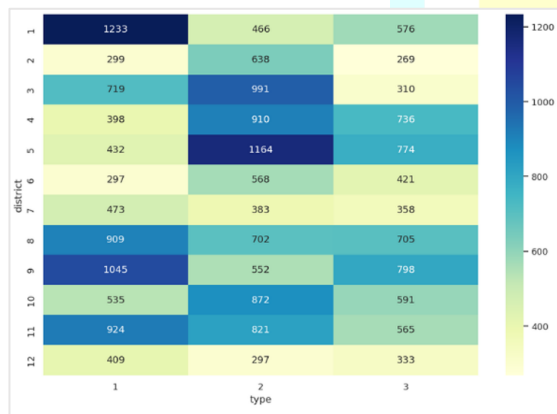


圖 9

房價直方圖

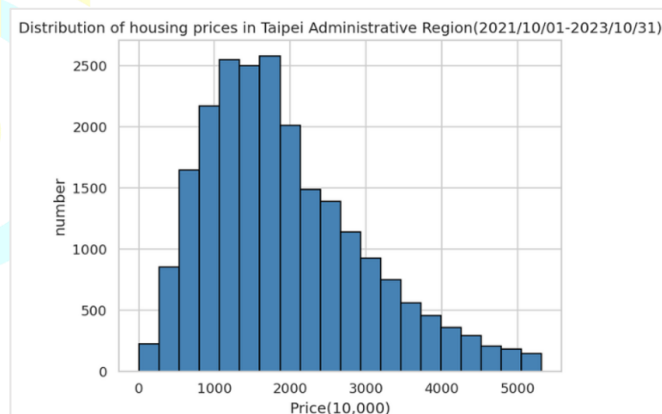
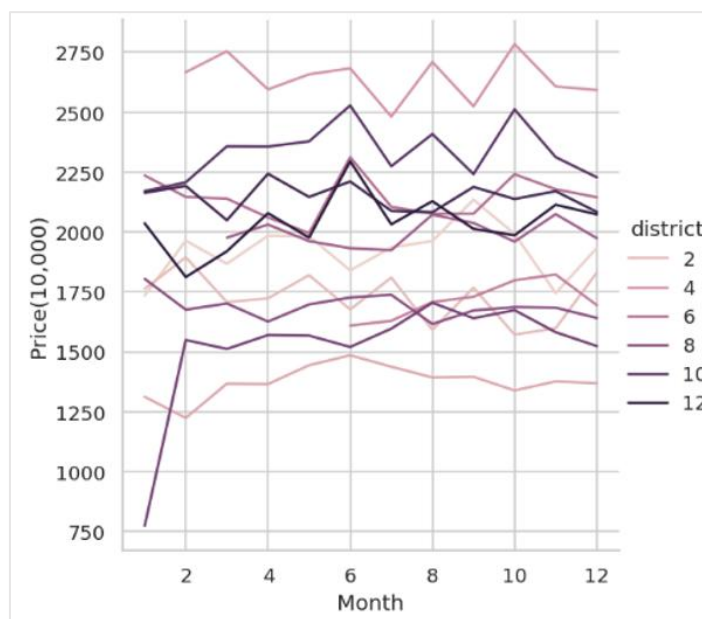


圖 10
房價和月份的折線圖



三、你是我的眼：CNN 如何做人工智慧的「眼睛」

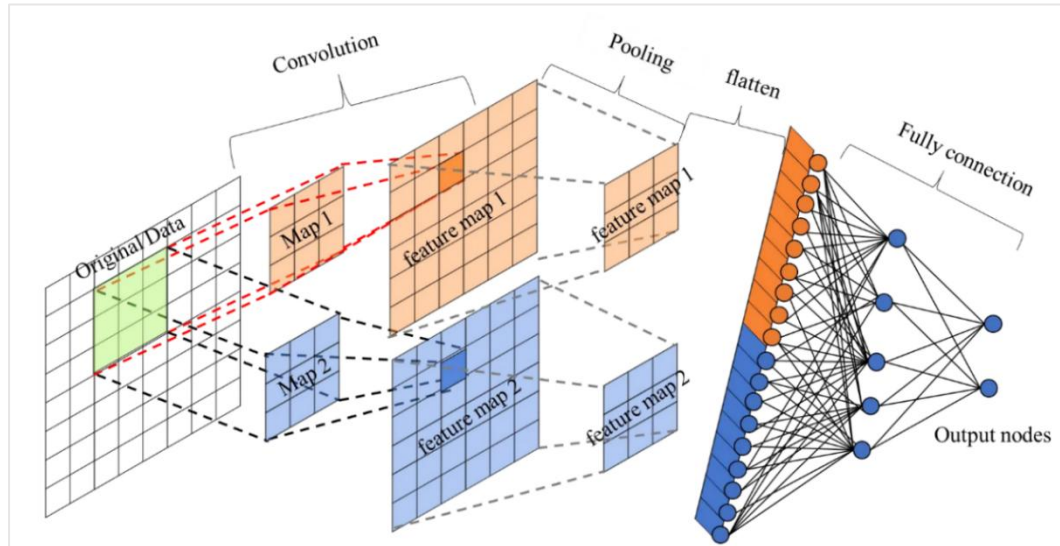
再看看生活中的例子。由於人行道破損的地磚，因常出現高低差及破裂情況會影響行人安全，造成民眾受傷的案例也屢見不鮮，新聞媒體的相關報導也不少，所以此專題想利用自行收集雙北地區的地磚圖片（共計 3,131 片），來自己建立資料庫，並使用深度學習的卷積神經網路（CNN）和 YOLO 來進行圖像辨識。

卷積神經網路主要應用於圖片辨識，相當精準，目前已運用於各大停車場的車牌辨識，非常實用。卷積神經網路有三層，卷積層（convolutional layer）篩選特徵資訊，池化層（pooling layer）抓取重要資訊的層級，最後有攤平（flatten）與全連接層（FC layer）相輔相成，攤平後的資料可以確保每個資料有相對應的神經元處理，展現 CNN 的基本架構。YOLO 是基於 CNN 所發展的即時辨識模型，能夠辨識會行動的物體，或是在動態的平台上辨識物件，像是手機、平板等。其 CNN 的架構詳見圖 11。

深度學習和機器學習相似，亦需要標記和建立模型，並使用驗證集來驗證模型精確度，YOLO 的訓練流程圖見圖 12。圖 13 實驗結果表明，雖然地磚的髒污程度、陰影、水漬會導致誤判情形發生，但是在調校模型之後，其準確率均達到 88% 以上。執行專題學生亦發現訓練次數並不是越多愈好。在實驗 YOLOv8 的過程中，比較 150 次訓練和 200 次訓練，產生的 mAP50 均為 94.2%，比訓練 125 次 mAP50 96.6% 還低，說明了過多的訓練甚至會造成減少一些正確率的結果（mAP：目標檢測演算法性能評估參數），這些學習均為同學研究後的心得，所以說，學 AI 還是得動手做才有感覺，

比較實在。圖 14-15 是和活動相關照片。

圖 11
卷積神經網路 (CNN)



註：圖像取自 Huang, 2018。

圖 12
訓練 YOLOv8 的流程圖



圖 13

驗證模型準確性——模型得到未知的圖片後，會將認為符合破損地磚的部分框住

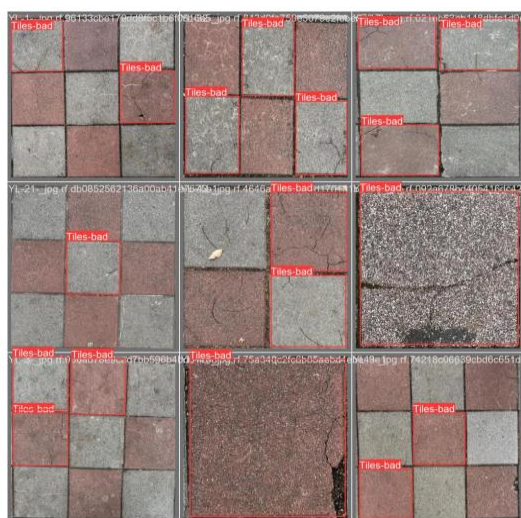
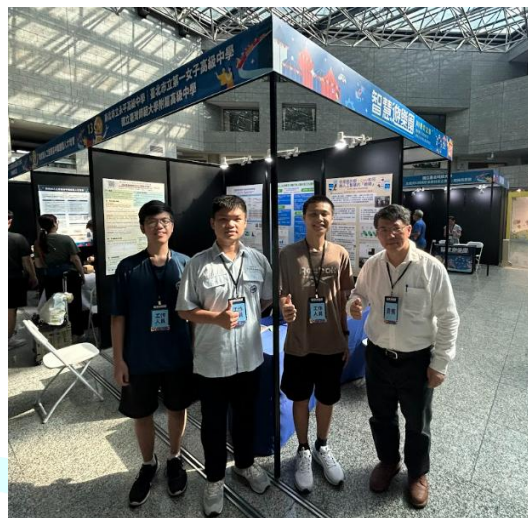


圖 14

2025 年 114 年 8 月 10 日，在高雄國立科學工藝博物館「智慧遊樂園：AI 奇幻之旅」師大附中 AI 專題公開展示



註：參展同學左起蔡昕翰、黃偉晟、曾毅展、右一為李柏翰老師。

圖 15

師大附中計畫主持人李柏翰老師（右），執行「教育部 113 年補助中小學校人工智慧教育計畫」榮獲 A 類計畫績優學校之佳績



註：師大附中廖純英校長（中）代表接受教育部資科司長官頒獎，廖校長感謝本校計畫主持人李柏翰老師，以及陳和寬、林宏宗、吳孟修、梁雅雯老師協同合作，也感謝師大附中資訊室李啟龍主任與阮麗如同仁協助幫忙計畫執行，提升本校 AI 能力，共創佳績。

結語

人工智慧是當代教育的核心議題，AI 從課堂上的實作接軌生活中的應用，可以逐步改變我們對「AI 學習」的想像與學習門檻。本文所介紹的動態與靜態應用，正是一種實踐 AI 教育的具體成果展現。透過動態學習的剪刀石頭布遊戲，學生不僅能理解馬可夫鏈、轉移矩陣的數學邏輯和程式設計，更能理解 AI 如何「發現」人類行為模式；而房價預測與地磚辨識等靜態 ML 專題，則讓學生在資料預處理、模型選擇與調整程式的過程中，逐步掌握基礎機器學習與深度學習的參數空間，並從中發現新概念，如：訓練次數有其優化的可能；同樣架構的模型只要調整一些資料可能產生不同的結果。

學生們的專題展現思考歷程與邏輯，證明了探究與實作課程啟發學生對生活的創新思考，並鼓勵他們在實作的困難與 Bug 中學習。從完全隨機出拳，被 AI 壓制到認真思考 AI 的想法並戰勝之，都能看到學生的成長，此外，自主學習的能力亦是 AI 教育不可或缺的一環。透過 Buzz Note 等開放資源，學生得以依據自身興趣與能力進行延伸學習，從科普影片到高中物理，甚至深入工程數學、量子力學與論文的參考。自學所提供的跨領域、跨階段的網路學習，不僅打破了教學的框架，也讓學生跨域思考並整合，無形中提供更多 AI 應用的可能性。

AI 的興盛結合自主學習的理念，除了是新技術導入校園，也是教育角色的轉變。教師不同於過往做為知識的單向傳遞者，反而像是多方面學習的領航員，以另外一種方法作育英才。同時，學生將不再是被動的單向接受內容，而是主動的探索與創新的整合，一種嶄新的概念，這些專題研究，學生寫下自己的故事。

參考文獻

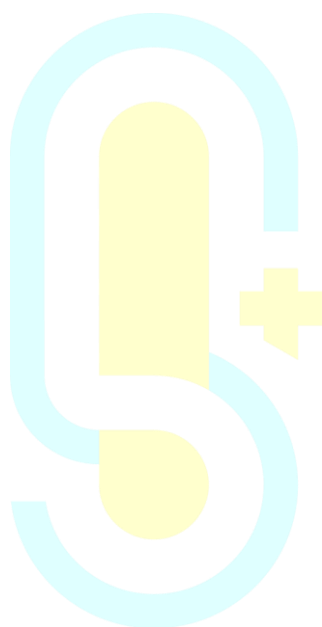
Meta 釋出擁有 700 億參數的 Code Llama。 <https://www.ithome.com.tw/news/161085>

周楷蓉、李柏翰（2025）。以機器建構不動產價值預測模型：隨機森林預測房價。

<https://youtu.be/cm-UqEfkoRg>

黃偉晟、李柏翰（2025）。〈應用深度學習判讀人行道地磚破損之研究〉，發表於 TANet 2025 臺灣網際網路研討會，已獲接受。

Huang, H. (2018, March 27). 卷積神經網路 (Convolutional neural network, CNN) — CNN 運算流程。 <https://chih-sheng-huang821.medium.com/卷積神經網路-convolutional-neural-network-cnn-cnn運算流程-ecaec240a631>



附錄

挑戰 AI：AI 用馬可夫鏈打敗人類的剪刀石頭布原始碼

```
import tkinter as tk
import random
from collections import defaultdict, Counter

from matplotlib.figure import Figure
from matplotlib.backends.backend_tkagg import FigureCanvasTkAgg
from PIL import Image, ImageTk

# Mapping between names and numbers
MOVE_NAMES = {1: "石頭", 2: "布", 3: "剪刀"}
NAME_TO_MOVE = {v.lower(): k for k, v in MOVE_NAMES.items()}

def winning_move(move):
    """Given a human move (1/2/3), return the move that beats it."""
    return (move % 3) + 1

class RSPredictor:
    def __init__(self, order=2):
        self.order = order
        self.history = []
        self.transitions = defaultdict(Counter)

    def predict(self):
        if len(self.history) < self.order:
            return random.randint(1, 3)
        key = tuple(self.history[-self.order:])
        next_counts = self.transitions.get(key)
        if not next_counts:
            return random.randint(1, 3)
        return next_counts.most_common(1)[0][0]

    def update(self, human_move):
        if len(self.history) >= self.order:
            key = tuple(self.history[-self.order:])
            self.transitions[key][human_move] += 1
        self.history.append(human_move)

class RPSApp:
    def __init__(self, root):
        self.root = root

        self.root.title("剪刀石頭布 AI")

        self.predictor = RSPredictor(order=3)
```



相關檔案下載

```

self.score_human = 0
self.score_ai = 0
self.round_num = 0

# Load move images
self.rock_img = tk.PhotoImage(file="rock.png")
self.paper_img = tk.PhotoImage(file="paper.png")
self.scissors_img = tk.PhotoImage(file="scissors.png")

# Load bottom pictures (resize as needed)
self.human_pic = ImageTk.PhotoImage(
    Image.open("human.png").resize((80, 80)))
self.robot_pic = ImageTk.PhotoImage(
    Image.open("robot.png").resize((80, 80)))

# Title labels
tk.Label(root, text="跟 AI 玩剪刀石頭布 ", font=(None,
60)).pack(pady=10)
tk.Label(root, text="挑戰 AI", font=(None, 40)).pack(pady=10)

# Move buttons
btn_frame = tk.Frame(root)
btn_frame.pack()
for move in ["石頭", "布", "剪刀"]:
    img = {"石頭": self.rock_img,
          "布": self.paper_img,
          "剪刀": self.scissors_img}[move]
    tk.Button(btn_frame, image=img, width=320,
              command=lambda m=move: self.play_round(m)).pack(
                side=tk.LEFT, padx=5)

# Result and score labels
self.result_label = tk.Label(root, text="", font=(None, 40))
self.result_label.pack(pady=10)
self.score_label = tk.Label(
    root, text="分數 - 你: 0 AI: 0", font=(None, 40))
self.score_label.pack(pady=5)

# Bottom frame: human image, chart, robot image
bottom = tk.Frame(root)
bottom.pack(pady=10)

tk.Label(bottom, image=self.human_pic).pack(side=tk.LEFT, padx=10)

# Matplotlib figure for bar chart

```

```

self.fig = Figure(figsize=(4, 2))
self.ax = self.fig.add_subplot(111)
self.canvas = FigureCanvasTkAgg(self.fig, master=bottom)
self.canvas.get_tk_widget().pack(side=tk.LEFT)

tk.Label(bottom, image=self.robot_pic).pack(side=tk.LEFT, padx=10)

# Initial draw
self.update_chart()

def play_round(self, human_input):
    human_move = NAME_TO_MOVE[human_input]
    # AI move
    predicted = self.predictor.predict()
    ai_move = winning_move(predicted)

    # Determine result
    if human_move == ai_move:
        outcome = "平手!"
    elif winning_move(human_move) == ai_move:
        self.score_ai += 1
        outcome = "AI 贏了這局!"
    else:
        self.score_human += 1
        outcome = "你贏了這局!"

    self.round_num += 1
    self.result_label.config(
        text=f"    第    {self.round_num}    局    :    你    :
{MOVE_NAMES[human_move]}, "
        f"AI: {MOVE_NAMES[ai_move]}\n{outcome}")
    self.score_label.config(
        text=f"分數 - 你: {self.score_human} AI: {self.score_ai}")

    self.predictor.update(human_move)
    print("AI 猜你接下來會出:", MOVE_NAMES[self.predictor.predict()])
    self.update_chart()

def update_chart(self):
    self.ax.clear()
    scores = [self.score_human, self.score_ai]
    bars = self.ax.bar(['you', 'AI'], scores)
    for bar in bars:
        h = bar.get_height()
        self.ax.text(bar.get_x() + bar.get_width()/2, h,
            f'{int(h)}', ha='center', va='bottom')

```

```
self.ax.set_ylim(0, max(5, max(scores)+1))
self.canvas.draw()

if __name__ == "__main__":
    root = tk.Tk()
    RPSApp(root)
    root.mainloop()
```

