# Python-棒球勝負預測

以類神經網絡方式建構中華職棒的勝隊預測模式

周緬緬 2023/1/5

# 專案動機

1. 棒球是國球，也是從小便有在瞭解與觀看的運動賽事。
2. 從世界棒球經典賽(2023年3月開打)到有自己支持的中華職棒球隊。
3. 除了想將棒球的熱情透過所學的知識體現外，更享受在喜歡的事物上有探索的機會與過程。
4. 所以想嘗試預測出運動彩券中棒球不讓分的情況下，可以如何預測出來，誰為勝誰為敗?

# 重點介紹-大綱

**方法**

- 研究對象
- 資料來源
- 研究流程

**結論**

- 訓練資料之描述性分析
- 神經網絡預測分析
- 總結

# 方法-研究對象

- **對象：**
  - 統一7-ELEVEn獅隊（UniLions，以下簡稱：統一隊）
  - 中信兄弟隊（Brother，以下簡稱：兄弟隊）
- **源由：**
  1. 兩隊都為四支創始球隊之一。
  2. 在2020年統一隊為冠軍，2021年兄弟隊為冠軍。

# 方法-資料來源【資料處理】

- **資料來源：**改寫API的方式去提取中華職棒大聯盟官方網站
- **資料長度：**
  - 在2015年至2022年上半季共計7.5個例行賽球季，從2015至2020年每年40場，2021年30場，2022年15場，共計285場對戰組合。
  - 採用前一期(上一場)的比賽數據作為本次比賽預測的依據，故需刪除每年的第一場對戰比賽，最終使用的數據，為277筆資料。
- **資料處理：**數據匯入至EXCEL的Power Query中進行

| 年份 | 2015 | 2016 | 2017 | 2018 | 2019 | 2020 | 2021 | | 2022 |
|---|---|---|---|---|---|---|---|---|---|
| 場次 | 39 | 39 | 39 | 39 | 39 | 39 | 26 | 3 | 14 |
| 訓練與測試樣本 | 260 | | | | | | | 17 | |
| 總計 | 277 | | | | | | | | |

# 方法-資料來源【變數設定】

- **自變數：**
  - 投手：防禦面，先發投手表現。
  - 打擊：攻擊面，打擊群表現。
  - 單場：單場比賽之狀態來計算出進階數據。
  - 累計：累計每場直觀數據球員(們)能力。
- **應變數：**
  - 比賽輸贏狀態，輸、贏、平局

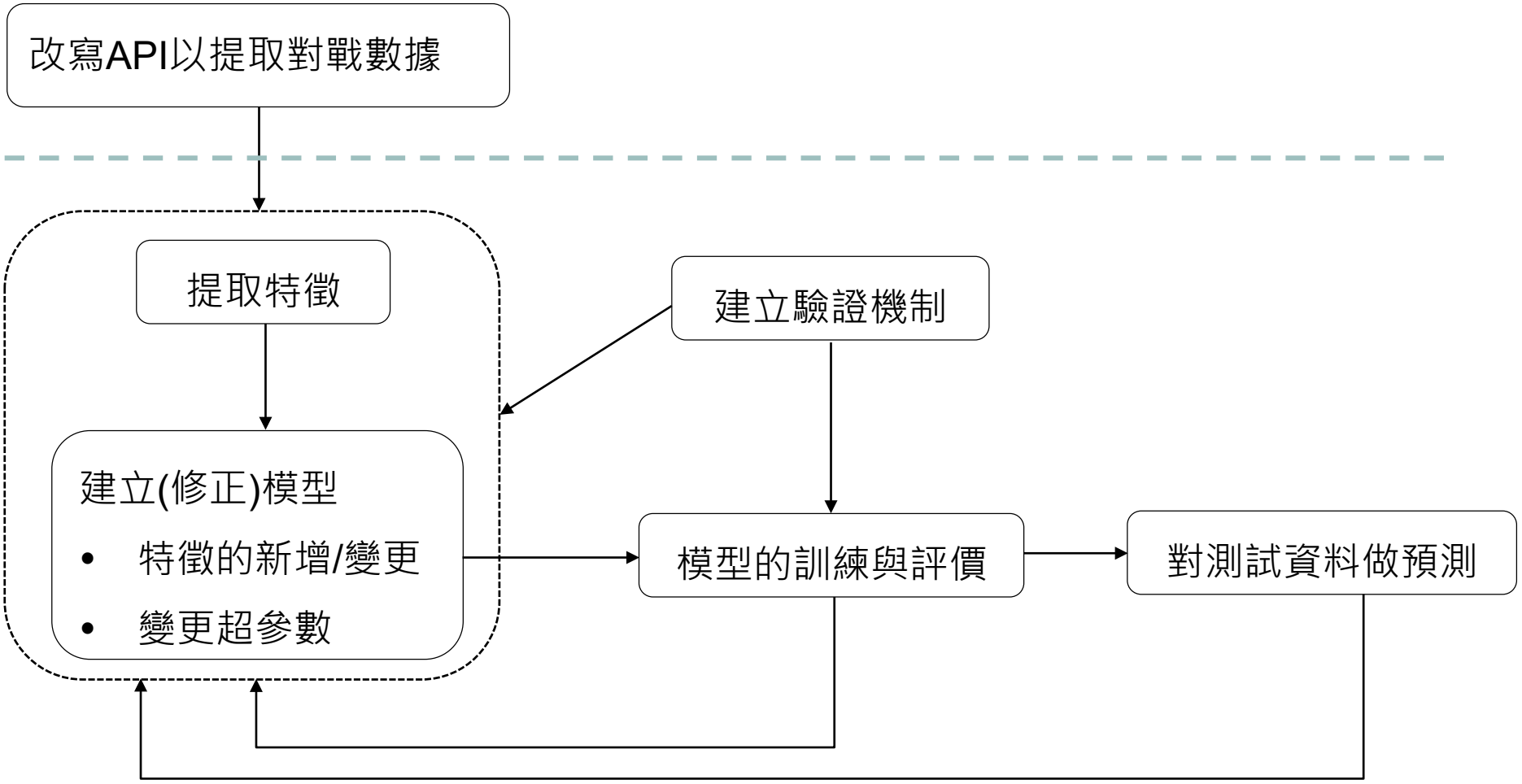
# 方法-資料來源【變數群組表】

| 項目 | | | 變數內容 | | 小計 | 總計 |
|---|---|---|---|---|---|---|
| 自變數 | 防禦(投手)方面 | 單場 | 投手獨立防禦率(FIP) | 予四壞球率(BB%) | 13 | 33 |
| | | | 被全壘打率(HR%) | 奪三振率(SO%) | | |
| | | | 場內安打率(BABIP) | 自責分率(ERA) | | |
| | | | 每局被上壘率(WHIP) | 投手投球習慣_5 | | |
| | | | 投手國籍(本國籍/他國籍) | | | |
| | | 累計 | 個人投球局數(IP) | 面對打席(BF) | 9 | |
| | | | 投球數(NP) | 安打(H) | | |
| | | | 全壘打(HR) | 予四壞(BB) | | |
| | | | 奪三振(SO) | 失分(RA) | | |
| | | | 自責分(ER) | | | |
| | 攻擊(打擊)方面 | 單場 | 場內安打率(BABIP) | 擊球入場率(BIP%) | 5 | |
| | | | 加權攻擊指數(wOBA) | 三振率(SO%) | | |
| | | | 被保送率(BB%) | | | |
| | | 累計 | 打數(AB) | 打點(RBI) | 6 | |
| | | | 得分(R) | 安打(H) | | |
| | | | 予四壞(BB) | 奪三振(SO) | | |
| | 當天比賽資訊 | | 比賽時段(下午/晚上) | 主客場(主場、客場) | 2 | 2 |
| 應變數 | | | 比賽輸贏狀態_3 | | 3 | |

分為:右投右打、右投左右開弓、右投左打、左投右打、左投左打

分為:輸、贏、平局

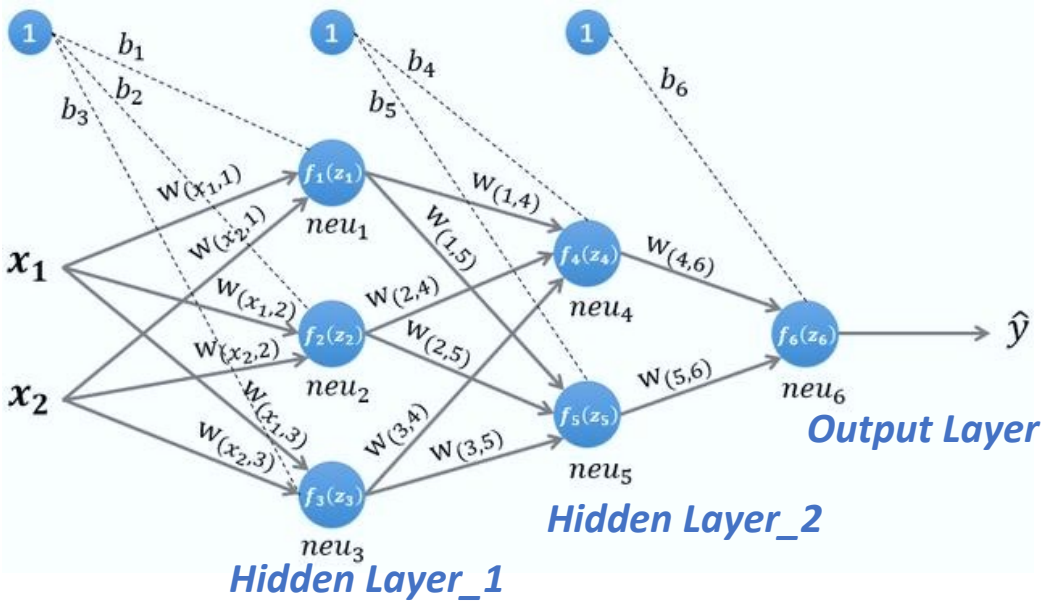
# 方法-研究流程圖




主要參考平松雄司、門脇大輔、阪田隆司、保坂桂佑, 2021(kaggle競賽攻頂秘笈)

# 方法-研究流程【(建立)訓練模型】

- **使用**：「倒傳遞類神經模型」(Back-Propagation Network, BPN)。
- **原理**：將順向傳遞(Forward Propagation)訓練後模型所產生的誤差作為刺激訊號，沿著神經元處理訊號的方向，逐層反向傳遞(Backward Propagation)修正隱藏層中節點的權重(Weight)、偏權值(Bias)，讓誤差減少。



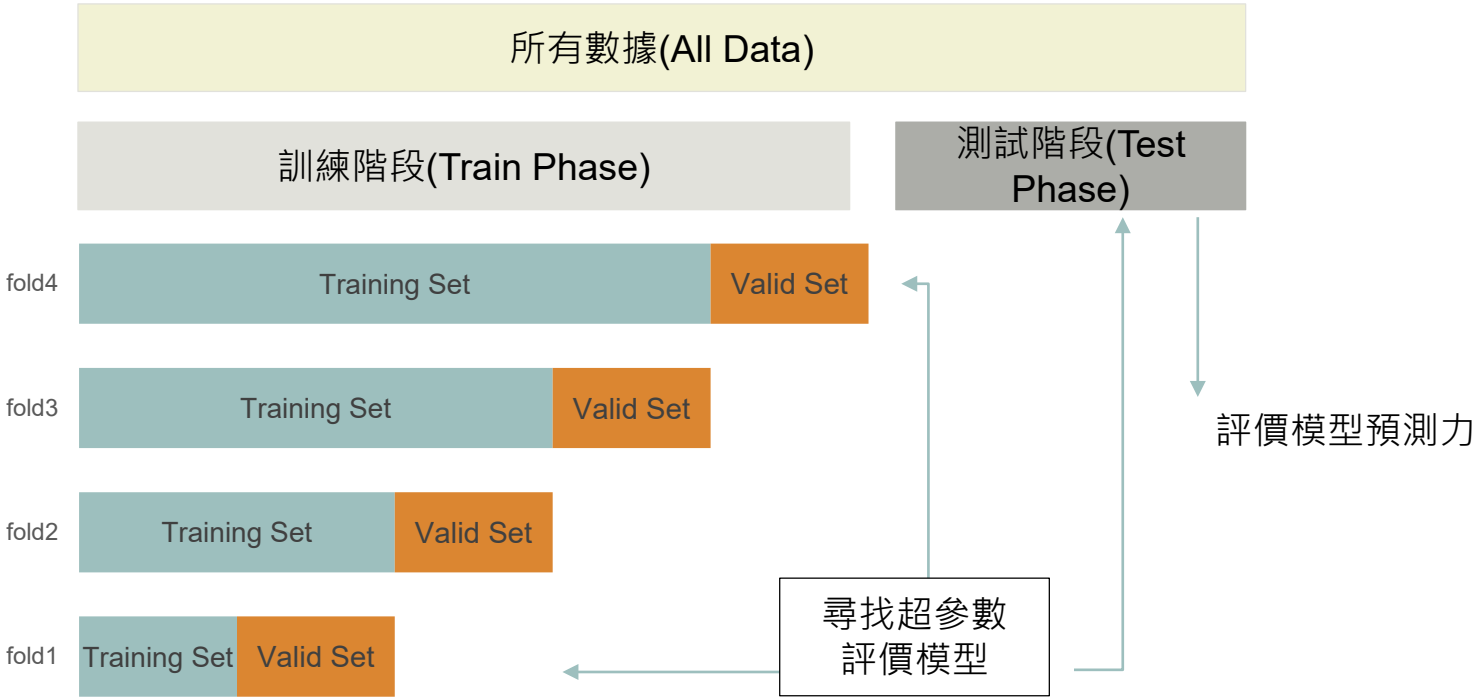

**使用類神經網絡傳遞訓練方式 (人工智能遇见磐创, 2018)**

# 方法-研究流程【(建立)訓練模型】

- **設定：**
  1. 使用PyTorch框架創建一個帶有3層全連接網絡模型。
  2. 啟動函數(activation function )：非線性Mish函數以及Softmax演算法。
  3. 損失函數:交叉熵(CrossEntropyLoss)。
  4. 最佳化器(Optimizer)：Adam()。
  5. 學習率衰減：ReduceLROnPlateau。

# 方法-研究流程【驗證模型】

- **使用**：scikit-learn中提供的TimeSeriesSplit方法進行時序的劃分。
- **原理**：此交叉驗證是K-Fold的特殊情況，在第k個劃分中，會產生出：前k組資料作為訓練集，第$(k+1)$組資料作為驗證集。



# 方法-研究流程【驗證模型】

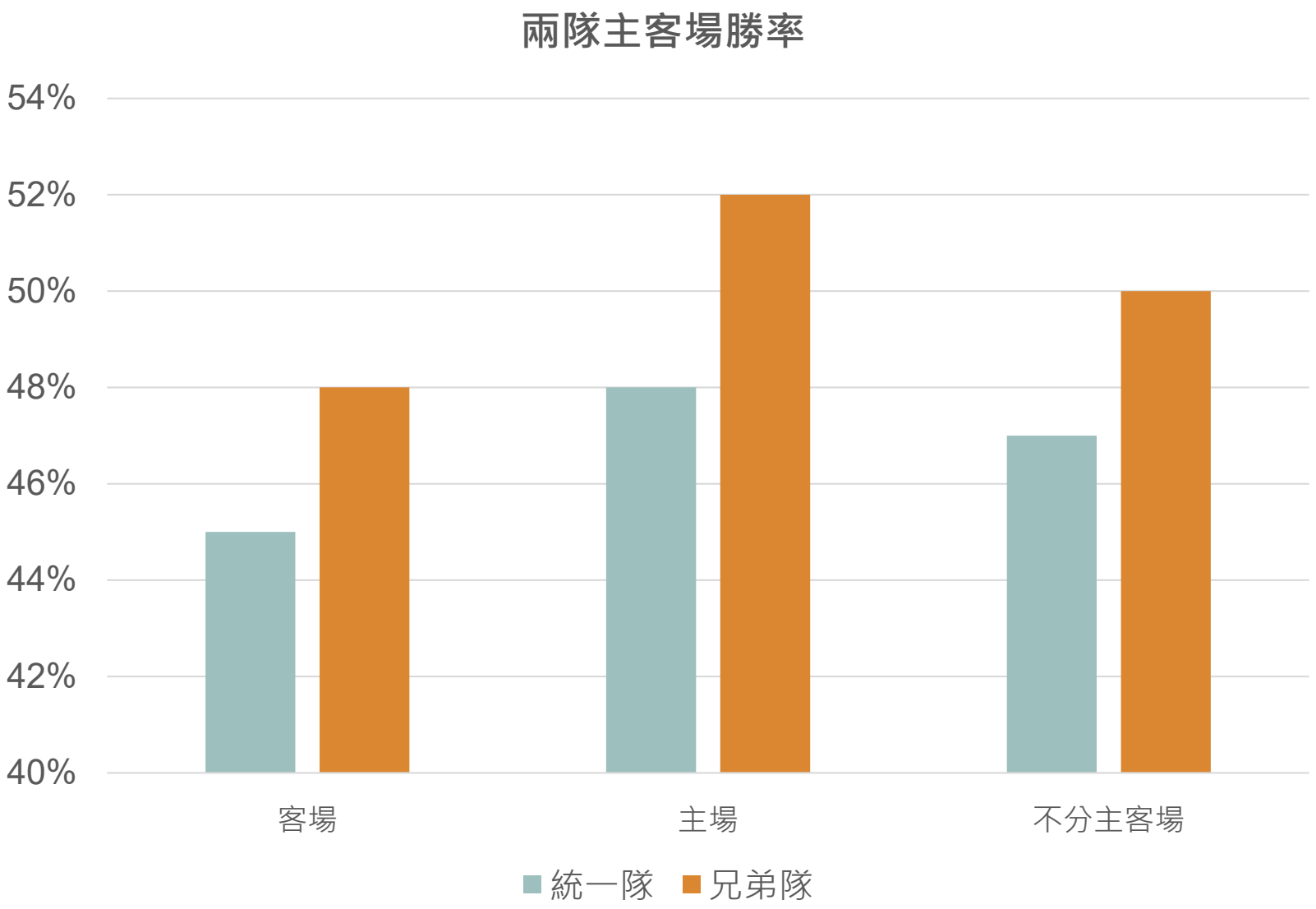
- **設定：**

1. 資料分為「訓練階段(Train Phase)」以及「測試階段(Test Phase)」，分別為260筆與17筆。
2. 透過TimeSeriesSplit將訓練階段的數據，分為訓練集(Training Set)與驗證集(Valid Set)。
3. 採取時序性的方式將訓練資料劃分成4個fold後，一邊維持訓練和驗證資料的時序關係性；一邊反覆進行驗證資料誤差值的計算。

# 方法-研究流程【數據預測】

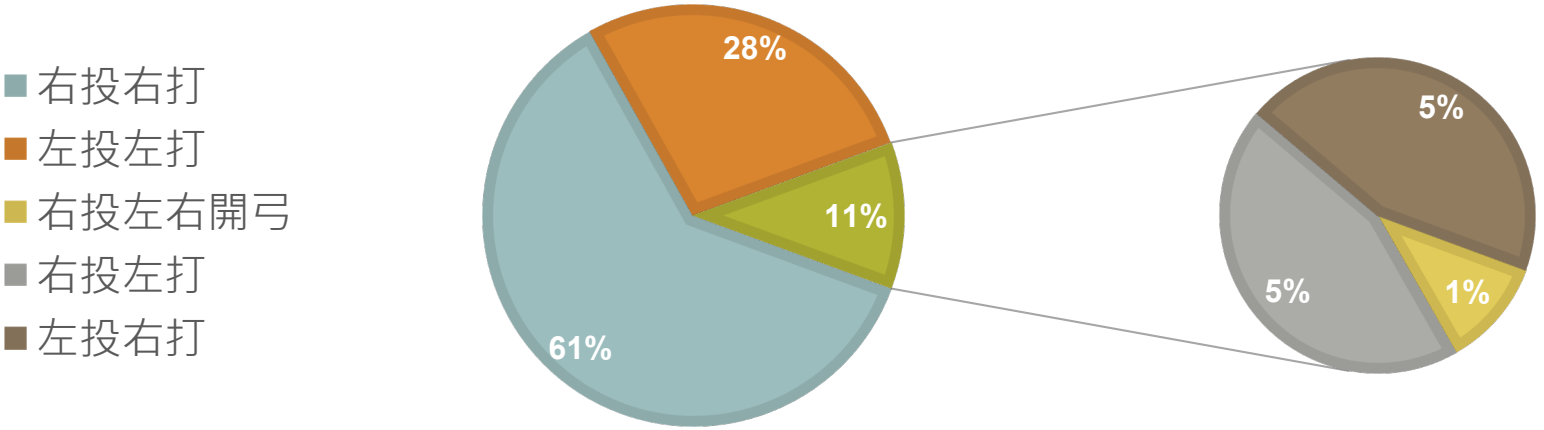
- **使用**：Optuna 函數庫(自動探詢函數庫)
- **原理**：進行貝氏最佳化(Bayesian Optimization)，考慮過去探索紀錄，有效率地找到需要的超參數。
- **設定**：
  1. 針對Optuna，除了設定每個超參數搜尋範圍，並將驗證後分數(準確率及損失值)進行探索，找尋最適超參數解。
  2. n_trials為[50,1500]，以每50為一區間，來控制將執行多少個參數空間，每個區間都會找出一個最適解，共計30個。
  3. 最後，以Fold4之驗證集(52筆資料)與劃分於訓練集中的2021年數據(26筆)之準確率，進行最終最適超參數解的尋找。

# 結論-訓練資料之描述性分析

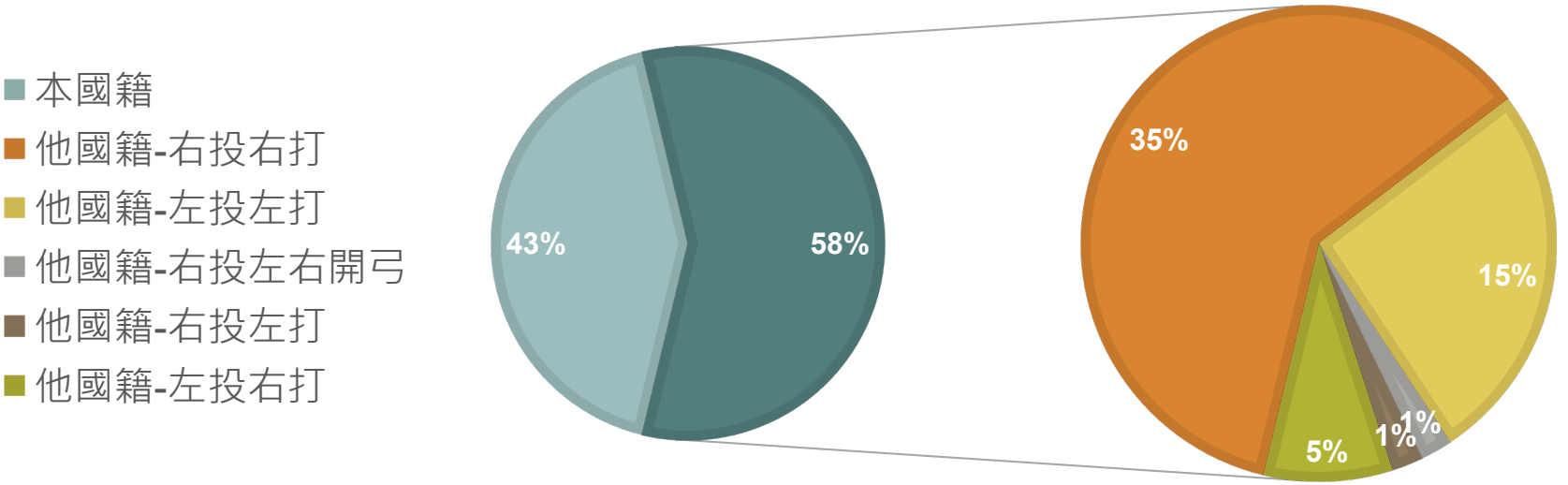


# 結論-訓練資料之描述性分析

【整體】先發投手投球習慣佔比

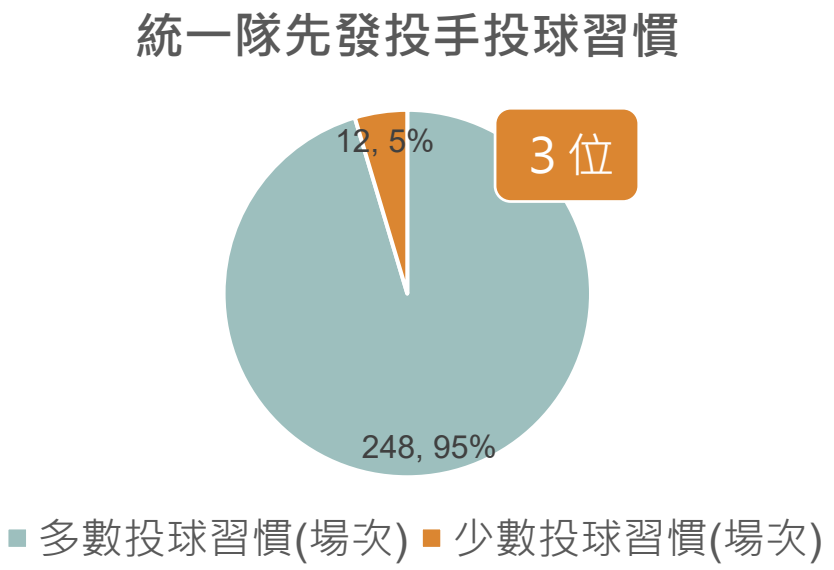


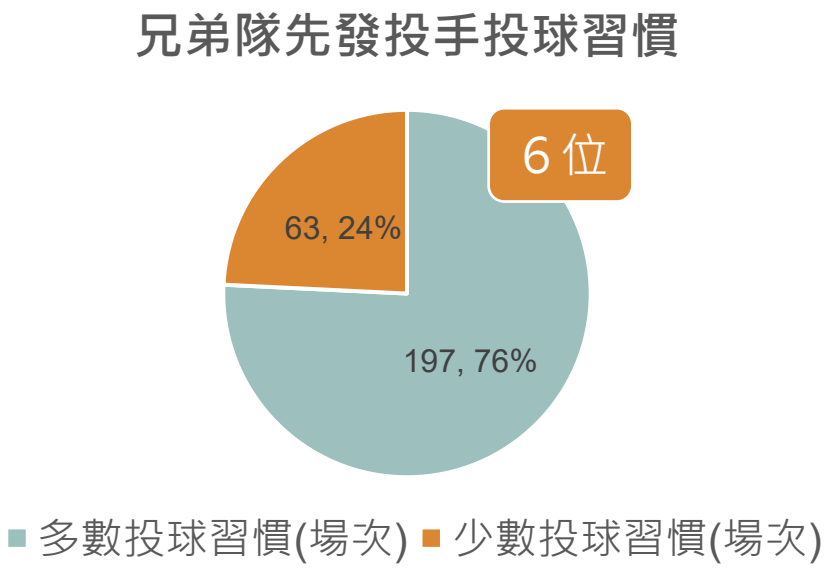

【整體】本國籍與他國籍佔比



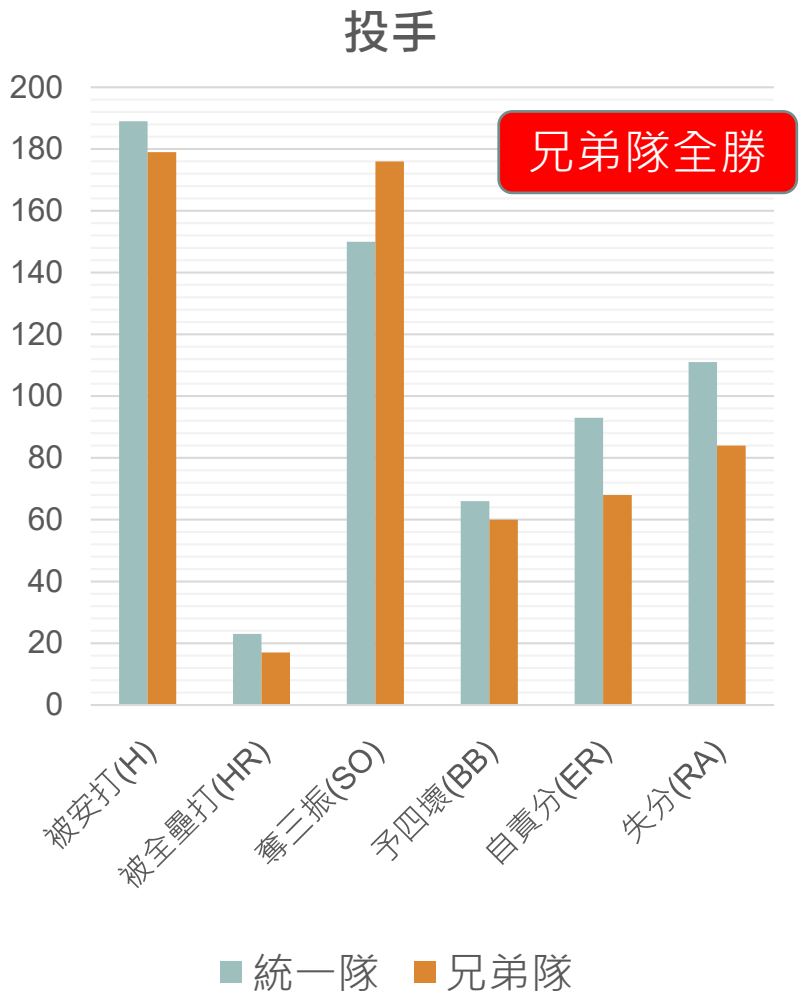
# 結論-訓練資料之描述性分析

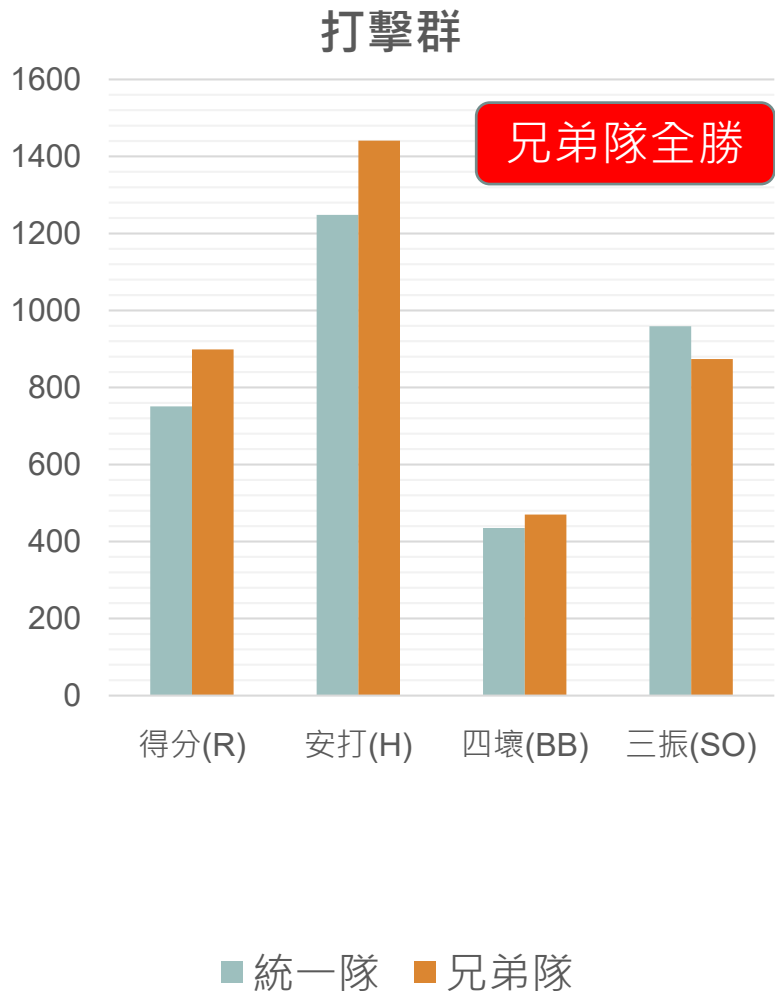
- 平均每位先發球員負擔場次：
  - 統一隊: 7.22場/每位先發投手
  - 兄弟隊: 5.91場/每位先發投手
- 兩隊先發投手投球習慣佔比：



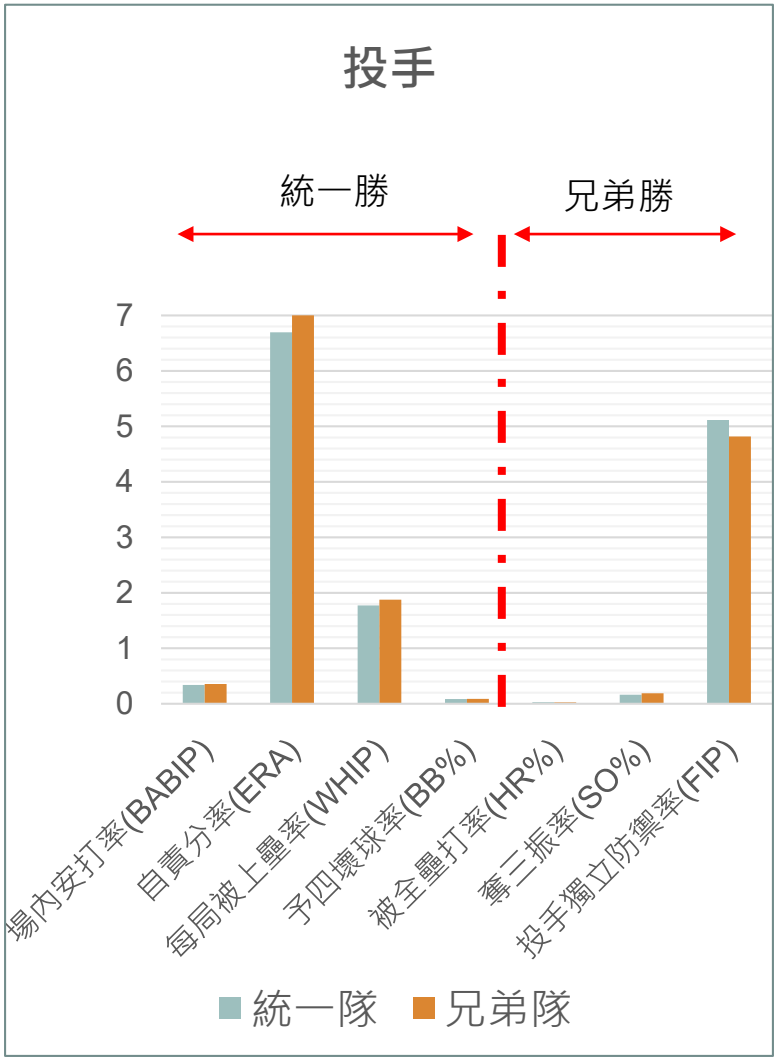
# 結論-訓練資料之描述性分析

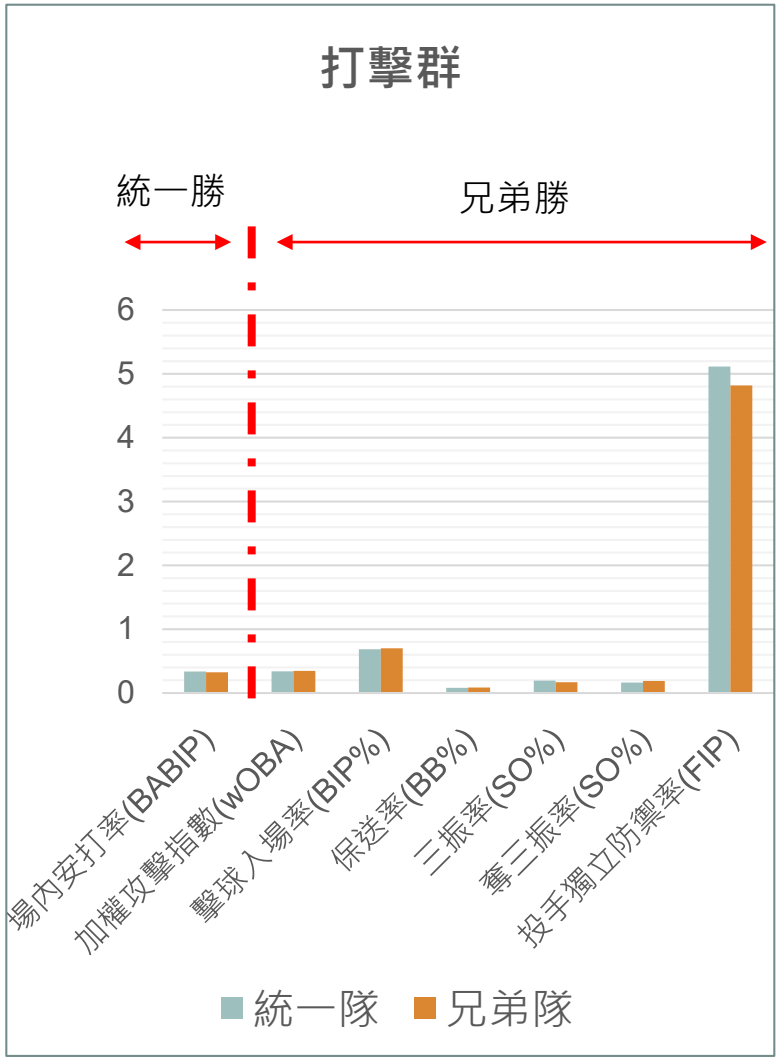



前一場比賽之數據分析(累計-直觀數據)-最大值

# 結論-訓練資料之描述性分析



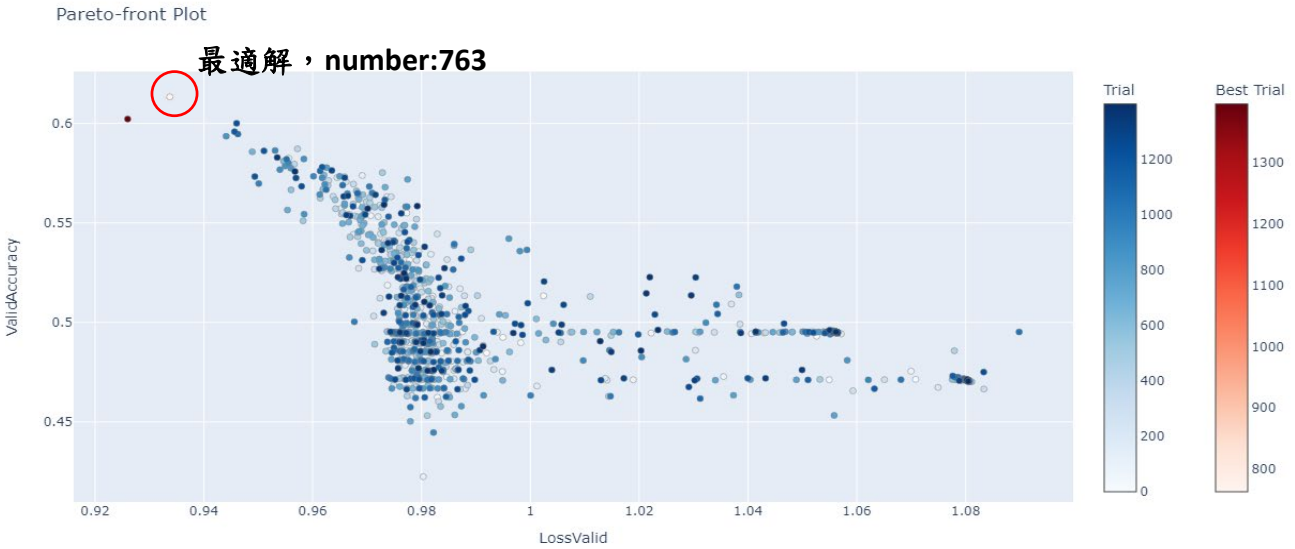




前一場比賽之數據分析(單場-進階數據)-平均值

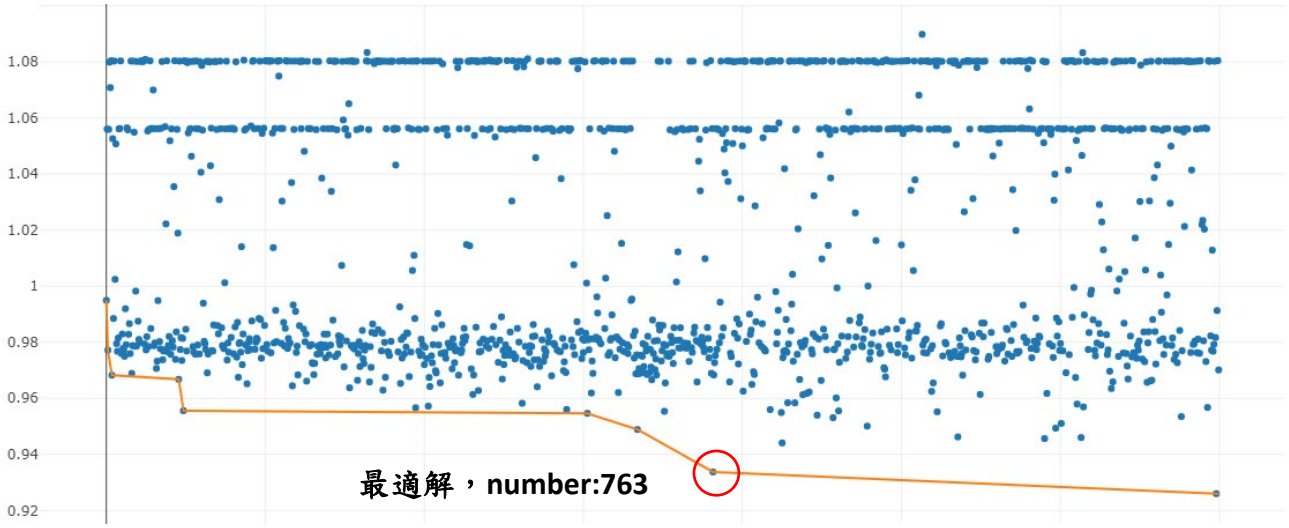
# 結論-神經網絡預測分析

- 透過Optuna與驗證準確率找到最適解在n_trails為1400集合中:

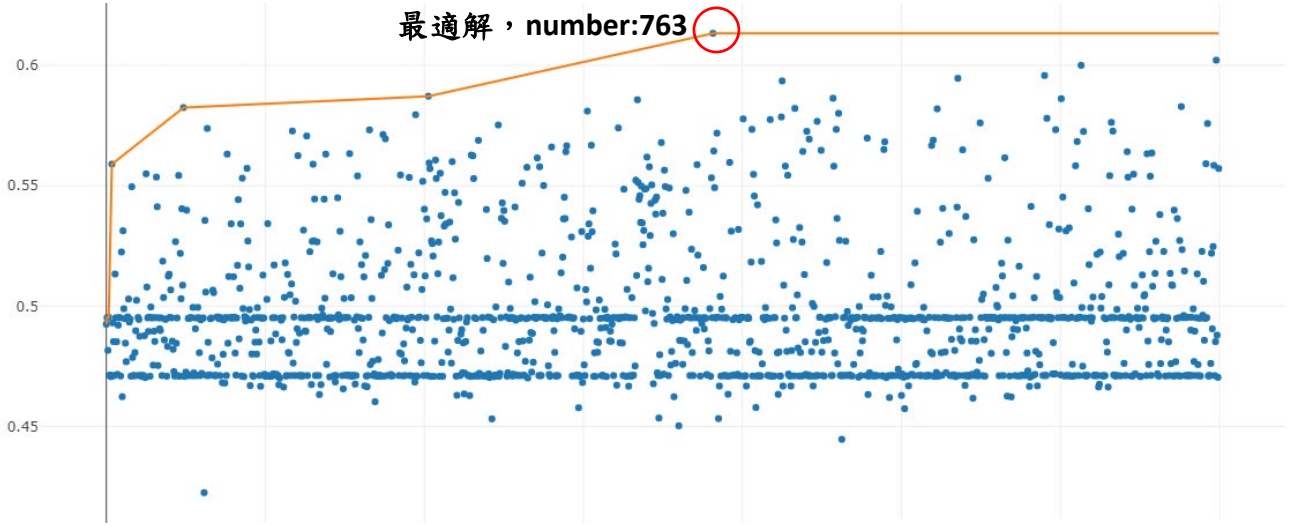
1. 隱藏層中神經元數目：第一層為146個神經元、第二層為28個神經元。
2. 丟棄率：第一層丟棄率0.07704681063017517、第二層丟棄率0.01101962939725109 6。
3. 循環次數：289
4. 訓練的學習率：0.025517738842518 93。



# 結論-神經網絡預測分析




平均驗證損失值最小化




平均驗證準確率最大化

# 結論-神經網絡預測分析

- **最後預測結果：**
  - 召回率：100%，精確率為70%。
  - 代表統一獅贏球時，都可以被鎖定，其預測有誤的部分，都在於預測統一隊勝，但實際上統一隊是敗的情況。
  - 預測準確率為82.3529%。

1為統一隊勝；0為統一隊敗(兄弟隊勝)；2為平局之虛擬變數。

| | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **實際勝負** | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 1 | 0 |
| **預測勝負** | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 1 | 0 |

| | 真實勝 | 真實敗 |
|---|---|---|
| 預測勝 | 7 | 3 |
| 預測敗 | 0 | 7 |

# 總結(貢獻)

- 將模型的預測，運用在運動彩券中，推敲出下一場比賽的輸贏狀態。
  1. 爬取中華職棒網頁中的數據資料。
  2. 變數中加入進階數據，而非僅使用直觀數據。
  3. 考量數據含有時間序列。
  4. 透過Python中Pytorch框架建立模型(非使用統計軟體)。