2024 年【科學探究競賽-這樣教我就懂】

普高組 成果報告表單

題目名稱：NBA 勝負預測

一、摘要

為了分析 NBA 各場比賽的勝負，在本研究中，我們利用網路爬蟲爬取各種可能影響球隊勝負的特徵，例如：各個球員在 2K 遊戲裡的能力值、主客場、上場時間、球員年薪 … 等特徵。接者我們利用機器學習中的邏輯斯回歸依照特徵重要性，建立一個最佳方程式來預測獲勝機率，獲勝機率高的隊伍便視為勝利方。最後，我們會以歷史資料來驗證此公式的泛化性，然後評估這些數據對勝負的影響大小。

二、探究題目與動機

NBA 的賽事總是讓人熱血沸騰，且 NBA 為全世界水平最高的籃球賽事，吸引世界各地許多球迷的關注。而球隊的勝負更是其中重要的一環。有許多體育迷和賭博愛好者對於預測比賽結果有濃厚的興趣。提供準確的比賽勝負預測可以提高比賽的觀賞價值和參與度。在 NBA 中，球隊經理的職責主要就是透過交易球員、選秀...等方式來打造最佳球員陣容，但要找到合適的人並不是那麼的容易。為了準確分析賽事輸贏，我們計畫蒐集各種球員可能會影響球賽勝負的因素，再以近十年總冠軍賽勝負分析出這些因素對球賽勝負影響的大小，最後整理出一個公式。透過這個公式，球隊經理可以模擬出目前陣容預測的勝率為多少，以作為是否要買斷或選某個球員的依據。

三、探究目的與假設

我們計畫製作出運用 Python 中的 requests、BeautifulSoup 與 Pandas 等套件進行資料蒐集、整理後而產生的 NBA 勝負預測公式。

1. 利用網路爬蟲爬取各種資料
2. 以 NBA 近十年例行賽勝負分析出因素影響力大小
3. 分析出勝算較大之球隊

四、探究方法與驗證步驟

(一) 研究方法

1、網路爬蟲

網路爬蟲，是一種可以自動蒐集網路上資料的技術，許多搜尋引擎都會透過網路爬蟲蒐集各種資訊，進一步分析後成為使用者搜尋的資料，許多公司也會自行開發不同的網路爬蟲程式，進行大數據收集與分析。在網路爬蟲程式中的套件，requests 套件程式是相對流行的，它具備了多種指向目標主機 request 的用法，如下表一示，requests 套件用法

表一：requests 套件用法

GET	向網站請求資源
POST	把資源透過請求傳輸給網站

PUT	把資源存入網站的主機內部
DELETE	把資源從網站主機內部刪除

當網站收到我們的 request，會回傳一個回應(response)，這個回應中有伺服器回傳的資訊，如下表二 response 回傳的資訊。

表二：response 回傳的資訊

url	網站的位址
content	回應資訊的內容，以 byte 形式回傳
text	回應資訊的內容，以 string 形式回傳
status_code ● 200 ● 403 ● 404 ● 500	回應的狀態 ● 正常 ● 沒有權限 ● 找不到網站 ● 伺服器錯誤

2、邏輯斯迴歸

邏輯斯迴歸是用來處理分類問題，希望結果是找到一條最佳的直線方程式將我們的蒐集的資料做分類。其核心概念主要是利用激勵函數中的 sigmoid function 來將輸出轉換成 0~1 的值，這個值代表的是可能為這個類別的機率。簡而言之，Logistic Regression 就是一種利用利用直線方程式及機率轉換的方法，用來預測某一事件發生的機率，特別適用於二元分類的問題，例如贏/輸、及格/不及格等等。

Scikit-learn：為目前最主流的機器學習開源套件，裡面包含了各種常見的機器學習演算法，以及各種分析、訓練模型的好用函式。除此之外，他還提供各式資料庫，供我們驗證自己的機器學習演算法。下表三所示，為本專題所使用到的函式。

表三：使用到的邏輯斯迴歸函式

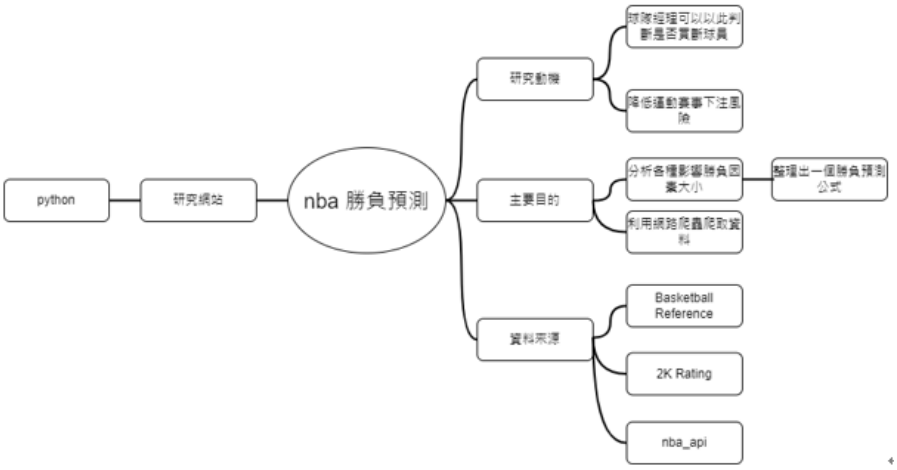
LogisticModel=LogisticRegression()	建立邏輯斯回歸模型
LogisticModel.fit(X, y)	使用輸入 X 及輸出 y 這兩個訓練資料集對邏輯斯回歸模型進行訓練
LogesticModel.score(X, y)	使用輸入 X 及輸出 y 這兩個資料集進行準確率評分

3、HTML (超文本標記語言)

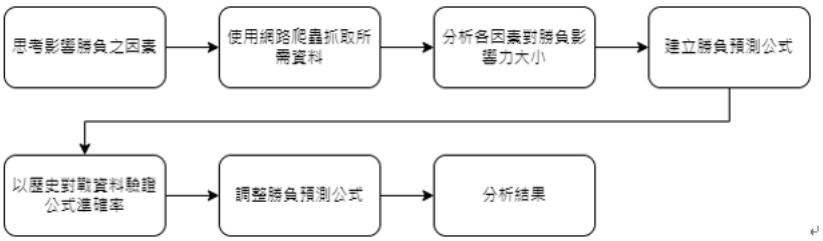
HTML 是打造網頁最基礎的工具，他用來表述及定義網頁的內容，並呈現在網路瀏覽器上。HTML 由一系列的標籤(tag)組成，這些標籤描述了 HTML 各元件的意義和屬性。

在爬蟲結束後，回傳的內容許多都是由 HTML 組成，而我們必須解析 HTML 檔案中的各種標籤，才能獲得我們想要的資訊，以本專題為例，我們會很常需要解析 HTML 檔案，而此檔案中又以 tr 標籤最為常見。tr 標籤定義為水平。

(二) 研究架構



圖一：研究架構圖



圖二：研究流程圖

(三) 研究分析與結果

1、爬取球員各項特徵的歷史資料

(1) 爬取球員 2K 能力值

我們從 hoopshype 網站中找尋 2013 到 2022 賽季中各個球員在 2K 遊戲中的資料，分別有：球員能力值排名、球員姓名、球員能力值。為了獲取此網站的資料，我們使用 Python 的 requests 套件來對此網站進行 GET 的動作，並獲得回應。在確定此回應沒問題後，我們便使用 Python 的另外一個套件 BeautifulSoup 來對此回傳內容 (response.text) 進行解析。

我們又觀察到 tr 欄位內部含有 td 欄位，所以我們遍歷了所有找到的 tr 欄位再解析其 td 欄位，並取得一個串列，為解析後的結果，此串列中的內容：第 0 的位置_球員能力值排名、第 1 的位置_球員姓名、第 2 的位置_球員能力值。之後，我們再創建三個串列，分別把所有的排名、姓名、能力值給儲存起來，並做成一個字典。最後，我們再利用 Python 中的 Pandas 套件來將字典做成 DataFrame 型態，並轉換成 CSV 格式保存，如下圖三所示。

(2) 爬取各場比賽的勝方以及主場球隊

為了比對各種特徵對於賽場的影響，我們必須先蒐集各場比賽的結果 (勝方)，來當作這些資料集的基準真相 (Ground Truth)，之後將我們所列的特徵訓練出迴歸模型後代入，來驗證特徵的影響力，除此之外，我們還能順便蒐集到主客場的資料。

我們發現 Basketball Reference 此網站中有各賽季各場比賽的資訊，於是我們一樣使用 request 套件向此網站進行 GET 的動作，得到回應後確認無誤，再使用 BeautifulSoup 套件來對回應內容進行解析。

解析完 tr 表格中所有的 td 欄位後，我們得到了一個串列，而此串列的內容為：客場球隊、客場球隊分數、主場球隊、主場球隊分數。由於此網站並未直接提供勝方，因此我們必須自己從「客場球隊分數」及「主場球隊分數」來進行判斷。接著我們使用六個串列分別把每一場比賽的客場球隊隊名、客場球隊得分、主場球隊隊名、主場球隊得分、比賽輸贏和主場記下來，並做成一個字典，之後透過 Python 中 Pandas 的套件將其轉換成 DataFrame 的格式，再轉換成 CSV 格式保存，如下圖四所示。

```
19            player_name = tr_column[1].text.strip()
20            value = tr_column[2].text.strip()
21
22            rank_list.append(rank)
23            player_list.append(player_name)
24            value_list.append(value)
25
26    nba_dict = {
27        "rank": rank_list,
28        "player_name": player_list,
29        "value": value_list,
30    }
```

圖三：爬取球員 2K 能力值程式碼

```
1   import requests
2   import json
3   import time
4   import pandas as pd
5
6   from bs4 import BeautifulSoup
7   from team import return_player_list
8
9   response=requests.get("https://www.basketball-reference.
10
11  team1_list=[]
12  score1_list=[]
13  team2_list=[]
14  score2_list=[]
15  winner_list=[]
16  home_list=[]
17  players1_list=[]
18  players2_list=[]
19  print(response.headers)
20
21  with open("team.json", "r") as json_file:
22      name_dict = json.load(json_file)
```

圖四：爬取各場比賽的勝方以及主場球隊程式碼

(3) 爬取各球隊球員名單

我們從網站 https://www.basketball-reference.com/ 尋找各球隊 2014-2023 賽季的球員名單。Basketball Reference 完整提供了我們這些資料，於是我們一樣使用 request 套件向其進行 GET 的動作，得到回應後確認無誤，再使用 BeautifulSoup 套件來對回應內容進行解析。

解析完 tr 表格中所有的 td 欄位後，我們得到了一個串列，而此串列的內容就是球隊裡所有球員的名單。最後我們再用一個串列把這些球員名單記下來。由於後面程式會使用到這些球員名單，因此我們把它做成一個函式，如果之後其他程式要使用，只要直接從此程式檔引入就好了，如下圖五所示。

(4) 利用 API 爬取球員各賽季平均上場時間

我們在網站 https://github.com/swar/nba_api 中找尋到他人已建立好的 NBA 各年數據資料庫，並下載其製作好的 Python 套件 nba_api，透過這個 API，我們可以爬取球員的各年各項數據，例如：球員 ID、球員出場場次、總上場時間，最後以球員總上場時間除以出場場次來獲取球員平均上場時間。

```python
def return_player_list(team):
    response=requests.get(f"https://www.basketball-reference.

    name_list=[]
    if response.ok:
        soup=BeautifulSoup(response.text,"html.parser")
        results=soup.find_all("tr")
        for result in results:
            th_row = result.select("th")
            if th_row[0]["data-stat"] == "number":
                tr_row=result.select("td")
                if len(tr_row)==0:
                    continue
                name = tr_row[0].text.strip()
                name_list.append(name)

        return name_list
```

圖五：爬取各球隊球員名單程式碼

```python
            tr_column = result.select("td")
            name = tr_column[1].text.strip()
            salary = tr_column[2].text.strip()
            salary = (int(salary[1:].replace(",", "")))
            name_list.append(name)
            salary_list.append(salary)

    nba_dict={
        "name": name_list,
        "salary": salary_list,
    }

    df = pd.DataFrame(nba_dict)
    df.to_csv(f"salary_{year}.csv")
```

圖六：爬取球員年薪程式碼

(5) 爬取各球員年薪

我們從網站 HoopsHype 尋找球員 2014 到 2023 賽季中各球員的年薪。解析完爬蟲爬下來的欄位後，我們可以得到球員名單以及他們各年的年薪，如上圖六所示。

2、資訊整合

為了將收集到的資料進行整合，我們讀取了所有製作好的 CSV 檔，並將年份、球隊、球員、2K 能力值、年薪、平均上場時間進行資料對齊，如下圖七所示，之後，我們分析了每場對局，並依照以下規則萃取資料出來，製作成訓練集：

- 若有球員未出現在 2K 名單中，我們統一設置他的 2K 能力值為 60。
- 若有球員年薪未出現在年薪名單中，我們統一設置他的年薪為 10000 美元。
- 若有球員未出現在上場時間名單中，我們統一設置他的平均上場時間為 10 分鐘。
- 在每隊中，我們取 2K 能力值最高的前 12 人，然後把他們的：2K 能力值、2K 能力值乘以平均上場時間、年薪，分別進行加總，並取平均 (除以 12)。

```
,team1,score1,team2,score2,winner,home,players1,players2
0,Philadelphia 76ers,117,Boston Celtics,126,Boston Celtic
1,Los Angeles Lakers,109,Golden State Warriors,123,Golden
2,Orlando Magic,109,Detroit Pistons,113,Detroit Pistons,De
3,Washington Wizards,114,Indiana Pacers,107,Washington Wi
4,Houston Rockets,107,Atlanta Hawks,117,Atlanta Hawks,Atl
5,New Orleans Pelicans,130,Brooklyn Nets,108,New Orleans
6,New York Knicks,112,Memphis Grizzlies,115,Memphis Grizz
7,Chicago Bulls,116,Miami Heat,108,Chicago Bulls,Miami He
8,Cleveland Cavaliers,105,Toronto Raptors,108,Toronto Rapt
9,Oklahoma City Thunder,108,Minnesota Timberwolves,115,Mi
10,Charlotte Hornets,129,San Antonio Spurs,102,Charlotte
11,Denver Nuggets,102,Utah Jazz,123,Utah Jazz,Utah Jazz,"
12,Dallas Mavericks,105,Phoenix Suns,107,Phoenix Suns,Pho
13,Portland Trail Blazers,115,Sacramento Kings,108,Portla
14,Milwaukee Bucks,90,Philadelphia 76ers,88,Milwaukee Buc
```

圖七：資訊整合的 CSV 檔

```python
X_train, X_test, y_train, y_test = train_test_split(input, out

logisticModel = LogisticRegression(random_state=0)

logisticModel.fit(X_train, y_train)

print(f"模型準確度 : {logisticModel.score(X_test, y_test)}")
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS C:\Users\User\Desktop\project> python predict.py
模型準確度 : 0.6097560975609756
PS C:\Users\User\Desktop\project>
```

圖八：準確率測試執行結果

3、比賽勝負預測

為了要預測比賽輸贏，我們使用到先前利用網路爬蟲所爬取的各項資料，再利用 Python 中的套件 scikit-learn，將資料上傳進 Logistic Regression 模型裡面做訓練，求出最適合資料的直線方程式。接著我們做一個 dictionary 來儲存各球員所對應到的 2k 能力值，並抓取先前爬出來的對戰資料、主客場、比賽輸贏資料。如果是主場或獲勝的球隊分別給他一個為 1 的數值，是客場或落敗的球隊給他 0。再來我們從先前爬取的各球隊球員名單，計算各隊球員的平均 2k 能力值。最後將球員的各項資料及球隊輸贏各自存在列表內，並以 8：2 的比例劃分為訓練集

和測試集，之後利用先前提到的模型，使用訓練集中的資料進行訓練。訓練完成後，我們再利用測試集對已訓練好的模型進行準確率測試，而目前的測試準確率達 71%，如下圖八所示。

(四) 針對弱點進行 SWOT 分析

表四：系統之 SWOT 分析

優勢(Strenth)	劣勢(Weakness)
1.資料多樣性：透過網路爬蟲技術，可以獲取多種資料，有助於分析 NBA 例行賽的勝負。 2.深入分析能力：透過數據分析，能評估多個潛在影響因素，進而確定影響勝負的主要因素。 3.預測準確性：透過深入的數據分析，有助於識別出在 NBA 總冠軍賽中勝算較大的球隊，為球隊和球迷提供寶貴的資訊。	1.資料品質限制：爬取的資料可能存在品質問題，包括缺失值、錯誤值等，這可能影響分析結果的準確性。 2.潛在偏差：在分析過程中可能存在潛在的偏差，例如對某些因素的過度重視或忽略，這可能導致分析結果的偏差。
機會(Opportunities)	威脅(Threats)
1.應用擴展：這個系統的應用範圍可以擴展到其他領域，如其他體育賽事或金融市場等，從而擴大其應用價值和商業前景。 2.技術改進：可以不斷改進爬蟲技術和分析模型，提高資料的準確性和分析的精度，從而提升系統的性能。	1.法律法規限制：爬蟲活動可能受到法律法規的限制，如版權法或隱私權法，這可能限制系統的發展和應用。 2.競爭威脅：可能有其他競爭對手開發類似的系統，競爭壓力可能影響系統的市場地位和商業前景。

五、結論與生活應用

(一) 結論

1、 利用程式進行比賽勝負預測可以消除情緒上的干擾，但無法對球員的即時身體狀態來做出相對彈性的預測。

2、 隊伍狀態是一個動態的因素，包括球員受傷、球員表現波動等。這些狀態的變化可能難以預測，並對模型產生不確定性。

3、 未來研究可考慮引入更多因素以提高預測能力，同時持續更新擴充數據集是改進模型的重要一環。

(二) 生活應用

1、 賭博：NBA 勝負預測可以作為賭博和下注的參考。透過分析球隊的表現、球員的狀態、比賽場地等因素，可以制定出預測比賽結果的策略，從而在合法的賭博平台上進行下注。

2、 Fantasy Basketball：在比賽中，參與者可以根據球員的表現獲得得分。通過預測 NBA 比賽的結果，可以幫助玩家選擇哪些球員放在他們的陣容中，以獲得最佳表現。

3、 投資和股票市場：一些投資者可能會利用 NBA 比賽的預測結果來做出相關公司股票的投資決策。例如，如果一支球隊在比賽中表現出色，可能會對該球隊所屬的公司的股票價格

產生影響。

4、 觀看比賽：對於喜歡觀看 NBA 比賽的人來說，對比賽結果的預測可以增加他們的參與感和興趣。通過預測結果，觀眾可以更深入地理解球隊之間的比賽動態，並且在比賽期間享受更多的期待和刺激。

參考資料

1. 錢寧 (2022 年 7 月 18 日)。基於時序模型和圖神經網路之 NBA 季後賽勝負預測。國立台北科技大學資訊工程研究所：碩士論文。
2. Code Gym（無日期）。用 Python 輕鬆取得 NBA 所有數據。https://reurl.cc/dLZ642