

壹、前言

一、研究動機

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的勝率預測為多少，以作為是否要買斷或選某個球員的依據。

二、研究目的

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。

(一) 利用網路爬蟲爬取各種資料

當我們進行 NBA 近十年例行賽勝負的分析時，首先必須執行一項關鍵任務，就是透過網路爬蟲技術來擷取多種資料。這些資料包括球隊的陣容、球員表現、比賽統計數據，以及其他相關資訊。這些資料的獲取是分析 NBA 總冠軍賽勝負的基礎，並有助於我們確定可能影響比賽結果的因素。

(二) 以 NBA 近十年例行賽勝負分析出因素影響力大小

我們將分析 NBA 例行賽的勝負結果，尤其是在過去十年的比賽中，我們會考慮多個潛在的影響因素，並評估它們的相對重要性。這些因素可能包括球隊的整體實力、主客場優勢、球員表現、教練戰術、比賽日程，甚至是裁判判罰。通過深入的數據分析和統計技術，我們可以確定哪些因素對 NBA 例行賽的勝負具有最大的影響力。

(三) 分析出勝算較大之球隊

透過這個分析，我們將能夠識別出那些在 NBA 總冠軍賽中擁有較高勝算的球隊。這些球隊可能在特定因素上表現出色，並且能夠利用這些優勢來贏得比賽。我們的分析將有助於揭示成功的關鍵，並提供有價值的見解，以便球隊和球迷更好地了解 NBA 總冠軍賽的競爭環境。

貳、文獻探討

一、文獻回顧

表一：文獻回顧概要與本研究分析

作者及年份	文章概要	收穫及對研究幫助
林億蓉 (2020)	<p>1.使用球員個人數據建構迴歸模型用於預測得分，使用迴歸樹與線性迴歸建構迴歸模型，預測兩隊伍的每位上場球員的得分後，加總並比較雙方預測總得分高低，即可得到球隊的預測結果。</p> <p>2.使用 Basketball Reference 查尋各球隊及球員數據。</p>	我們獲得了可以查出各項球員和球隊資料的網站，也了解可以用查出來的各項數據來創一個迴歸模型來預測各隊得分。在此文章中，我們還學習到了建構模型的方法，以及有參考性的參數。
陳政逸 (2013)	本研究對於以往的對戰數據或歷年來球員逐場表現數據，來預估下一場的球隊勝負以及算出整年度球隊的預期勝負，並利用統計軟體創立一套完整的系統來加速推算以及更進一步更精準的預測。	透過這篇文章，我們理解到賽場上會有一些重要特徵能夠影響比賽勝負，例如：主客場、球員 2K 能力值等等。而藉由統計這些特徵，我們可以建立一個完整的系統來對目前球員陣容預期的勝場數進行預測。
錢寧 (2022)	<p>1.在數據的預測方面，我們透過使用時序模型預測賽前球員表現取代傳統的數學運算，以更貼近真實世界的賽事勝負預測。</p> <p>2. 本論文運用到所有上場球員表現，並首度將圖神經網路運用於賽事預測，其準確率達到 76.9%，已超越現有研究方法。</p>	<p>我們獲得更多有關數據統計的方法和概念，對於如何對特徵進行統計有了更多的認識、理解。</p> <p>這篇文章讓我們認識先進的預測、統計方法，在未來如果我們想延續此研究，可以沿用這篇文章的想法。</p>

表一資料來源：研究者自行彙整

二、網路爬蟲

網路爬蟲 (web crawler) 是一種能夠自動收集網路資料的技術。許多搜尋引擎如 Google、Yahoo 等，都透過網路爬蟲收集各種資訊，進一步分析後成為使用者的搜尋結果。許多公司也開發不同的網路爬蟲程式，用於大數據收集與分析。在網路爬蟲程式中，requests 套件是相對流行的工具，它具備了多種指向目標主機的 request 用法。該套件提供了簡潔而強大的 API，讓開發者能夠輕鬆地發送 HTTP 請求、獲取網頁內容並進行處理。這使得使用 requests 套件的程式能夠有效地爬取網路資料，進而進行後續的分析和應用。如下表二所示，requests 套件用法。

表二：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

表二資料來源：研究者自行彙整。參考網站：Code Gym（無日期）。用 Python 輕鬆取得 NBA 所有數據。<https://www.codegym.tech/blog/python-nba-api>

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，如下表三所示，response 回傳的資訊。

表三：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	回應的狀態
● 200	● 正常
● 403	● 沒有權限
● 404	● 找不到網站
● 500	● 伺服器錯誤

表三資料來源：研究者自行彙整。參考網站：Code Gym（無日期）。用 Python 輕鬆取得 NBA 所有數據。<https://www.codegym.tech/blog/python-nba-api>

NBA 歷史悠久，站在賽場上的球員也非常多。如果想找出這些球員的各種數據，一個一個手動慢慢找一定會非常慢。而網路爬蟲技術可以幫助我們更快速的找出這些資料。因此，我們需要學習如何利用 Python 撰寫出網路爬蟲的程式。亦或者，我們可以透過人家已建立好的 API 來抓取相關資料。例如我們在本研究中使用到 nba_api 此 API 來抓取球員數據。如下表四所示，為 nba_api 所提供的函式。

表四：nba_api 提供的函式

get_active_players()	抓取現役球員名單
PlayerCareerStats(player_id)	透過 player id 抓取球員賽場上生涯數據。

表四資料來源：研究者自行彙整。參考網站：Code Gym（無日期）。用 Python 輕鬆取得 NBA 所有數據。<https://www.codegym.tech/blog/python-nba-api>

二、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。

三、邏輯斯迴歸 (Logistic Regression)

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。下表五所示，為本專題所使用到的函式。

表五：本專題所使用到的邏輯斯迴歸函式

<code>LogisticModel=LogisticRegression()</code>	建立邏輯斯回歸模型
<code>LogisticModel.fit(X, y)</code>	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
<code>LogisticModel.score(X, y)</code>	使用輸入 X 及輸出 y 這兩個資料集進行準確率評分

表五資料來源：研究者自行彙整。參考網站：**Code Gym**（無日期）。用 **Python** 輕鬆取得 NBA 所有數據。<https://www.codegym.tech/blog/python-nba-api>

參、研究方法

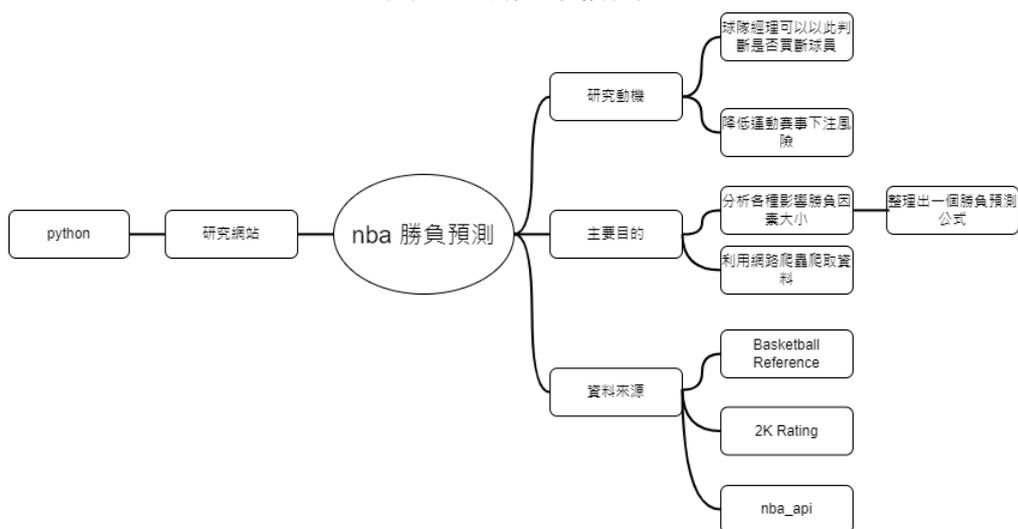
一、研究方法

本研究將以下列三種研究方法進行研究：

- (一) 文獻分析法：這個階段涉及蒐集來自相關網站的大量資料。研究者將透過爬蟲程式蒐集資料，這些資料將用於建立研究的背景和基礎，並提供有關主題的全面洞察。
- (二) 分析比較法：這個方法將在文獻探討之後進行。在這一階段，研究者將對多一點的資料進行深入分析和比較。這可能包括統計分析、機器學習等技術。目標是判別資料中的模式和關聯性。
- (三) 實作分析法：我們將根據分析結果開發實際應用的工具或模型，例如：**NBA** 賽場勝負預測公式。這個階段可能涉及複雜的資料前處理、模型訓練和測試。最終目標是將資料轉化為實際應用，以提供實用價值。

二、研究架構

圖一：研究架構圖



圖一資料來源：研究者使用 Draw.io 自行繪製

三、研究流程

圖二：研究流程圖



圖二資料來源：研究者使用 Draw.io 自行繪製

肆、研究分析與結果

一、爬取球員各項特徵的歷史資料

(一) 爬取球員 2K 能力值

我們從網站 [hoopshype\(https://hoopshype.com/nba2k/\)](https://hoopshype.com/nba2k/) 中找尋 2K14 到 2K23，也就是 2013 到 2022 賽季中各個球員在 2K 遊戲中資料，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，並獲得一個回應。在確定此回應沒問題後，也就是並未回傳錯誤碼，我們便使用 Python 的另外一個套件 BeautifulSoup 來對此回傳內(response.text)進行解析，因

為我們在回傳內容(HTML file)中發現，我們所需的資訊都是在<tr></tr>欄位之間，於是我們解析回傳內容中的所有 tr 欄位。

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，為爬取球員 2K 能力值程式碼。

圖三：爬取球員 2K 能力值程式碼

```

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     }
31
32     df = pd.DataFrame(nba_dict)
33
34     df.to_csv("NBA_2K13.csv")

```

圖三資料來源：研究者由 Visual Studio Code 實作截圖

(二) 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，為爬取各場比賽的勝方以及主場球隊程式碼。

圖四：爬取各場比賽的勝方以及主場球隊程式碼

```

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.com/leagues/NBA_2023_games.html")
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)

```

圖四資料來源：研究者由 Visual Studio Code 實作截圖

(三) 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，為爬取各球隊球員名單程式碼。

圖五：爬取各球隊球員名單程式碼

```

1 import requests
2 import pandas as pd
3
4 from bs4 import BeautifulSoup
5
6
7 def return_player_list(team):
8     response=requests.get(f"https://www.basketball-reference.com/teams/{team}/2018.html")
9
10    name_list=[]
11    if response.ok:
12        soup=BeautifulSoup(response.text,"html.parser")
13        results=soup.find_all("tr")
14        for result in results:
15            th_row = result.select("th")
16            if th_row[0]["data-stat"] == "number":
17                tr_row=result.select("td")
18                if len(tr_row)==0:
19                    continue
20                name = tr_row[0].text.strip()
21                name_list.append(name)
22
23    return name_list

```

圖五資料來源：研究者由 Visual Studio Code 實作截圖

(四) 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。

(五) 爬取各球員年薪

我們從網站 HoopsHype (<https://hoopshype.com/salaries/>)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。如下圖六所示，為爬取球員年薪程式碼。

圖六：爬取球員年薪程式碼

```

22
23         tr_column = result.select("td")
24         name = tr_column[1].text.strip()
25         salary = tr_column[2].text.strip()
26         salary = (int(salary[1:].replace(",","")))
27         name_list.append(name)
28         salary_list.append(salary)
29
30     nba_dict={
31         "name": name_list,
32         "salary": salary_list,
33     }
34
35     df = pd.DataFrame(nba_dict)
36     df.to_csv(f"salary_{year}.csv")
    
```

圖六資料來源：研究者由 Visual Studio Code 實作截圖

二、資訊整合

為了將收集到的資料進行整合，我們讀取所有製作好的 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，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。如下圖七所示，為資訊整合的 CSV 檔。

- 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。
- 若有球員平均上場時間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（一）2K 能力值、（二）2K 能力值乘以平均上場時間、（三）年薪分別進行加總，並取平均(除以 12)。

圖七：資訊整合的 CSV 檔

```

1     ,team1,score1,team2,score2,winner,home,players1,players2
2     0,Philadelphia 76ers,117,Boston Celtics,126,Boston Celtics,Boston Celtics,['Julian Champagnie', 'Dewayne
3     1,Los Angeles Lakers,109,Golden State Warriors,123,Golden State Warriors,Golden State Warriors,['Mo Bamba
4     2,Orlando Magic,109,Detroit Pistons,113,Detroit Pistons,Detroit Pistons,['Cole Anthony', 'Mo Bamba', 'Pac
5     3,Washington Wizards,114,Indiana Pacers,107,Washington Wizards,Indiana Pacers,['Deni Avdija', 'Will Barto
6     4,Houston Rockets,107,Atlanta Hawks,117,Atlanta Hawks,Atlanta Hawks,['Josh Christopher', 'Darius Days', '
7     5,New Orleans Pelicans,130,Brooklyn Nets,108,New Orleans Pelicans,Brooklyn Nets,['Jose Alvarado', 'Dyson
8     6,New York Knicks,112,Memphis Grizzlies,115,Memphis Grizzlies,Memphis Grizzlies,['Ryan Arcidiacono', 'RJ
9     7,Chicago Bulls,116,Miami Heat,108,Chicago Bulls,Miami Heat,['Patrick Beverley', 'Tony Bradley', 'Alex Ca
10    8,Cleveland Cavaliers,105,Toronto Raptors,108,Toronto Raptors,Toronto Raptors,['Jannet Allen', 'Mamadi D
11    9,Oklahoma City Thunder,108,Minnesota Timberwolves,115,Minnesota Timberwolves,Minnesota Timberwolves,['Da
12    10,Charlotte Hornets,129,San Antonio Spurs,102,Charlotte Hornets,San Antonio Spurs,['Lamelo Ball', 'James
13    11,Denver Nuggets,102,Utah Jazz,123,Utah Jazz,Utah Jazz,['Christian Braun', 'Bruce Brown', 'Thomas Bryant
14    12,Dallas Mavericks,105,Phoenix Suns,107,Phoenix Suns,Phoenix Suns,['Davis Bertans', 'Reggie Bullock', 'F
15    13,Portland Trail Blazers,115,Sacramento Kings,108,Portland Trail Blazers,Sacramento Kings,['Ryan Arcidia
16    14,Milwaukee Bucks,90,Philadelphia 76ers,88,Milwaukee Bucks,Philadelphia 76ers,['Grayson Allen', 'Giannis
17    15,Los Angeles Clippers,103,Los Angeles Lakers,97,Los Angeles Lakers,Los Angeles Lakers,['Nicolas Batum',
18    16,New Orleans Pelicans,124,Charlotte Hornets,112,New Orleans Pelicans,Charlotte Hornets,['Jose Alvarado'
19    17,San Antonio Spurs,137,Indiana Pacers,134,San Antonio Spurs,Indiana Pacers,['Dominick Barlow', 'Charles
20    18,Chicago Bulls,100,Washington Wizards,102,Washington Wizards,Washington Wizards,['Patrick Beverley', 'T
21    19,Orlando Magic,98,Atlanta Hawks,108,Orlando Magic,Atlanta Hawks,['Cole Anthony', 'Mo Bamba', 'Paolo Ban
22    20,Toronto Raptors,105,Brooklyn Nets,109,Brooklyn Nets,Brooklyn Nets,['Precious Achiuwa', 'OG Anunoby', '
23    21,Boston Celtics,111,Miami Heat,104,Boston Celtics,Miami Heat,['Malcolm Brogdon', 'Jaylen Brown', 'Justi
24    22,Detroit Pistons,106,New York Knicks,130,New York Knicks,New York Knicks,['Marvin Bagley III', 'Saddiq
25    23,Memphis Grizzlies,129,Houston Rockets,122,Memphis Grizzlies,Houston Rockets,['Steven Adams', 'Santi Al
26    24,Utah Jazz,132,Minnesota Timberwolves,126,Utah Jazz,Minnesota Timberwolves,['Ochai Agbaji', 'Nickail Al
27    25,Denver Nuggets,128,Golden State Warriors,123,Denver Nuggets,Golden State Warriors,['Christian Braun', '
28    26,Phoenix Suns,111,Portland Trail Blazers,113,Portland Trail Blazers,Portland Trail Blazers,['Deandre Ay
29    27,San Antonio Spurs,114,Philadelphia 76ers,105,San Antonio Spurs,Philadelphia 76ers,['Dominick Barlow',
    
```

圖七資料來源：研究者由 Visual Studio Code 實作截圖

三、比賽勝負預測

為了要預測比賽輸贏，我們會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`，將資料進去 `Logistic Regression` 的模型裡面做訓練，求出最適合資料的直線方程式。首先我們做一個 `dictionary` 來儲存各球員所對應到的 2K 能力值。接下來抓取先前爬出來的對戰資料、主客場、比賽輸贏。是主場跟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。再來我們從先前已經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約 61%。如下圖八所示，為準確率測試執行結果。

圖八：準確率測試執行結果

```

168 X_train, X_test, y_train, y_test = train_test_split(input, output, test_size=0.2, random_state=42)
169
170 logisticModel = LogisticRegression(random_state=0)
171
172 logisticModel.fit(X_train, y_train)
173
174 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>
    
```

圖八資料來源：研究者由 Visual Studio Code 實作截圖

四、針對遊戲弱點進行 SWOT 分析並提出未來可能發展方向

(一) 針對遊戲弱點進行 SWOT 分析

表六：系統之 SWOT 分析

優勢(Strength)	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：透過統計技術和數據分析，能夠評估多個潛在影響因素的相對重要性，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)

<p>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。</p> <p>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。</p>	<p>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。</p> <p>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。</p>
--	---

表六資料來源：研究者自行彙整

(二) 未來可能發展方向

為提高程式準確率，可著重於擴展資料收集範圍，確保資料多樣性和完整性，以使模型更全面理解問題。深入資料分析可發現潛在模式和關聯，指導特徵工程優化。與領域專家合作挖掘有意義特徵，並考慮使用複雜模型結構如深度學習。調整超參數和使用高級優化算法可提升模型性能。定期更新和優化模型以應對環境變化和新資訊。

伍、研究結論與建議

一、結論

- (一) 使用模型來預測比賽結果需要更多年份的資料，依我們現在使用爬蟲爬取了十年的資料且準確率只有 61% 以得知，訓練的資料多寡是與模型預測的準確率有正相關。
- (二) 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。

二、建議

- (一) 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) 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。我們可以把這些複雜的因素考慮進去，增加模型準確率

陸、參考文獻

林億蓉(2020年6月30日)。應用機器學習於 NBA 球隊勝負預測-以金州勇士隊為例。國立勤益科技大學工業工程與管理研究所：碩士論文。

陳政逸(2013年6月21日)。利用球員能力指標預測 NBA 球賽戰績。逢甲大學統計學系統計與精算碩士班：碩士論文。

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。

Code Gym (無日期)。用 Python 輕鬆取得 NBA 所有數據。
<https://www.codegym.tech/blog/python-nba-api>