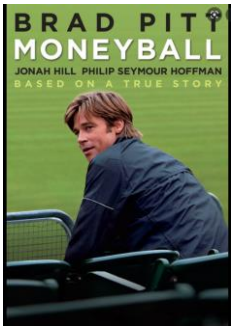
MLB daily game prediction

&

Stat Analysis

一、 前言與問題定義



<魔球>MONEY BALL

魔球是一部真實故事改編電影，故事背景為 2001~2002 年的奧克蘭運動家隊，2001 球季結束因為球團不想繼續花大錢綁住明星球員，所以 2002 年剛開季戰績嚴重受到影響，所以故事主角比力.比恩與他的球探團隊們應用數據分析的方式，了解到想要贏球上壘率佔了很大的關鍵，球隊應用了各項數據最後還在季末打出 20 連勝晉級季後賽

身為一個資深棒球迷加上中階的棒球數據迷，除了應用現有知識觀察數據，還想要試著自己建構一個模型去預測在每天的比賽之前去預測兩隊贏球的機率

二、 資料蒐集

利用爬蟲去找出每天比賽的勝負結果，以及找出球隊的近期數據，主要參考以下兩網站:

1. Fangraph: https://www.fangraphs.com/
2. Baseball reference: https://www.baseball-reference.com/

我 2011~2021 的團隊數據當作 train set(2020 因為疫情，賽季縮減，所以沒蒐集)，我的 target，也就是每天比賽的結果範圍都選擇每年的 5 月開始到 9 月 25 日。以下是我收集的團隊數據(**前 7 天的數據**，target 是 5/9 的話，就蒐集 5/2~5/8 的團隊數據)

以下表示方式 B 為打者數據，P 為投手數據，T 為整隊一起，括號後面為通常數字範圍

TEAM:球隊(T) **HR:**全壘打數(T) **BB%:**保送率(B,0~1)
K%:三振率(B, 0~1) **ISO:**純長打率(B,0~1) **BABIP:**場內打擊率(B,0~1)
AVG:打擊率(B,0~1) **OBP:**上壘率(B,0~1) **SLG:**長打率(B,0~1)
WOBA:進攻加權指數(B,0~1) **WRC+:**加權得分製造指數(B,-0~200)
BSR:跑壘貢獻指數(B,0 為 30 隊平均)
DEF:防守貢獻指數(B,0 為 30 隊平均)
K/9:每九局三振數量(P,0~27) **BB/9:**每九局保送數量(P,0~27)
HR/9:每九局全壘打數量(P,0~27) **FIP:**扣除守備自責分率(P,0~9)
E**RA:**自責分率(P,0~9) **GS:**七天內比賽數(T)
PBABIP:被場內打擊率(P,0~1)
YEAR:年 **MONTH:**月 **DAY:**日
WPER:目前勝率(T,0~1) **WPERN:**七天前勝率(T,0~1) **OPP:**對手球隊
TARGET:球隊勝負(TEAM 的勝負不是 OPP 的勝負，TEAM 贏為 1 輸為 0，沒比賽為-1)

TEAM	HR	BB%	K%	ISO	BABIP	AVG	OBP	SLG	WOBA	WRC+	BSR	DEF	K/9	BB/9	HR/9	FIP
CLE	14	9.20%	20.00%	0.268	0.329	0.301	0.378	0.569	0.403	159	0.6	-4.2	7.29	2.79	0.93	3
TBR	7	9.50%	17.30%	0.19	0.331	0.292	0.357	0.481	0.364	133	1.1	0.1	6.38	3.11	0.49	3
KCR	7	8.70%	13.00%	0.224	0.31	0.293	0.357	0.517	0.375	134	0.4	-0.8	5.47	4.59	1.94	6
STL	5	9.10%	15.70%	0.133	0.338	0.294	0.369	0.427	0.347	121	-0.8	0.8	5.65	3.25	0.99	4
NYY	9	13.20%	20.50%	0.181	0.287	0.25	0.353	0.431	0.345	114	-0.6	2.4	7	2.71	1	3
LAA	3	7.70%	15.90%	0.133	0.358	0.31	0.359	0.443	0.348	122	-0.8	1.2	5.93	2.8	0.82	3
ARI	7	12.20%	16.70%	0.188	0.259	0.236	0.336	0.424	0.334	106	-0.1	4.9	7.33	2.83	1.5	4
TOR	8	11.10%	17.20%	0.16	0.309	0.278	0.361	0.439	0.351	119	0.8	-2	6.71	3.88	0.88	4
CIN	7	9.90%	17.20%	0.17	0.274	0.25	0.335	0.42	0.335	110	1.4	2.4	6.85	3.33	1.76	5
ATL	7	9.60%	16.50%	0.176	0.286	0.259	0.329	0.434	0.331	109	-1	1.9	7.07	2.41	0	2

蒐集完結果如上圖，一共 44100 筆資料

三、 前處理

STEP1:將對手的資訊(查同一天的['OPP'])，傳給 TEAM，也就是將兩隊對戰組合的資訊存到同一個 row，TARGET 紀錄的是原本 TEAM 有沒有贏，贏的話為 1，OPP 贏的話為 0

STEP2:因為棒球比賽不是每天都有比賽，會有沒比賽的休兵日，所以將 TARGET 為-1 的 row 刪掉(這裡順便提一下，假設有一日雙重賽的話我，假設 A 隊 2 連勝 B 隊，則 A 的 TARGET 記 1，2 連敗的話記 0，1 勝 1 敗的話記-1)

```
Length of original dataset: 44100
After Step 2: 38306
```

STEP3:7 天連續期間假設比賽小於 4 場(GS < 4)，可能球隊因為季中的明星賽休息或是下雨太多影響手感，比賽斷斷續續，我覺得會影響 TARGET 判

斷，故刪除

```
After Step 2: 38306
After Step 3: 37233
```

STEP4:在 STEP1 的時候我不會將兩個對戰組合結果都記錄起來(因為只是相反)，只留其中一個(我後來遇到一個問題就是這樣 model predict 出來的結果放在 test set 上面的平均數會等於 0.524 左右，因為我再做 STEP4 的時候是從 7 天內團隊數據較佳的球隊的 row 保留，所以導致 18688 筆資料有 9800 筆左右是 TARGET=1 的，TARGET=0 的反而只有 8800 多筆，所以後來我 STEP4 就拿掉了)

```
After Step 3: 37233
After Step 4: 18688
```

四、 增加新特徵以及挑選特徵

增加新特徵

STEP1 先看我要取那些特徵，再把需要的對手的特徵加到同一個 row 裡(像是假設我要看 HR BB% ERA，我就會將'OPP'的 HR BB% ERA 加進去)

```
df["OPP_HR"] = np.nan
df["OPP_ERA"] = np.nan
df["OPP_BB%"] = np.nan
```

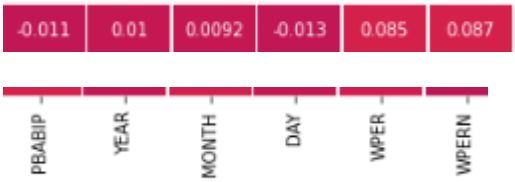
```
df.loc[i*30+k,"OPP_BB%"] = df.loc[i*30+j]['BB%']
df.loc[i*30+k,"OPP_ERA"] = df.loc[i*30+j]['ERA']
df.loc[i*30+k,"OPP_HR"] = df.loc[i*30+j]['HR']
```

STEP2 拿 TEAM 的數據減掉 OPP 的數據(假設 TEAM 的 ERA3.89，OPP 的 4.12，直接這樣看還好，如果拿 TEAM 的減掉 OPP 的就可以先做比較看正負來知道哪個比較大)

```
df['NEW_FIP'] = df['FIP'] - df['OPP_FIP']
df['NEW_WRC+'] = df['WRC+'] - df['OPP_WRC+']
df['NEW_OBP'] = df['OBP'] - df['OPP_OBP']
```

上圖為[新特徵]=[TEAM 特徵]-[OPP 特徵]

挑選特徵:



觀察:由於棒球真的是太難預測了，像是 ATL 在 14 連勝時竟然會連續輸兩場原本已經 10 連敗的 CHC，所以這些 HEATMAP 的相關係數都非常的小，不過至少像是 BB/9 這樣的特徵的確是跟 TARGET 是負的相關，所有特徵的正負相關都還在邏輯之內。然後這裡面最大的竟然就是 WPER(球隊勝率)，也就是說，假設有一個勝率沒很高的隊伍在一段時間打出高潮，團隊數據都表現得不錯，但下場剛好上勝率高的隊伍，這樣可能就會馬上被打回原形了

根據 HEAT MAP 以及我對棒球的認知，我總共挑了兩個類型的特徵

1. 只挑傳統數據的特徵(像是 AVG HR ERA)

```
0.5520945220193341
```
選取 OBP AVG ERA WPER

2. 進階數據的特徵(進階數據像是 FIP 是已經有學者將 BB K HR BABIP 等因素套用不同的權重產生出的一個評斷投手控場能力的數據)

```
0.560687432867884
```
選取 DEF WRC+ FIP WPER

這兩種作比較的話進階數據準確度好一點

我評斷準確的方式是先將 Y_pred(通常 output 在 0.4~0.6 間)，先將大於 0.5 的變 1，小於的變 0

再拿 Y_pred 跟 Y_true 計算 tp tn fp fn 等數值

```
tn, fp, fn, tp = confusion_matrix(Y_test, y_pred).ravel()
print(tn, fp, fn, tp)
print((tp+tn)/len(Y_test))
```

五、 模型訓練與結果討論

模型訓練:

我使用像是 HW3 所使用到的模型，得到 linear regression 欸有最好的效果，所以最後再 report 裡面顯示的各個數據都是使用 linear regression 的

結果

結果討論:

因為進階數據都是現代人拿一些傳統數據做加權得到的，所以其實理論上進階數據的結果會比較好沒錯，然後 WPER 也就是原本球隊的勝率是最重要的，因為棒球真的變化太多，所以很多數據都只能追求長時間下來的結果，如果一天一天來預測真的變化會太大，所以 56%我就覺得不錯了，還有一些部份留在下面的特別討論再討論

六、 特別討論

(一)不同數據近年變化

2015 16 球季以後 statcast 系統開始發展成熟，每個球隊也跟開始依賴科技進行訓練

進攻方面:各球隊開始追求把球往天空打產生更多全壘打，去追求得分效益更高的長打，打擊率(AVG)因此跟年分成負相關，HR 跟 ISO 跟長打相關的都成正相關(Year 都是 2011~2021)

Corr(YEAR,HR) = 0.85

Corr(YEAR,AVG) = -0.76

Corr(YEAR,ISO) = 0.81

守備方面:守備布陣使得 BABIP 有些微往下(打進場內的求更容易被布陣吃掉)

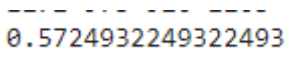
Corr(YEAR,BABIP) = -0.39

投手方面: 科技訓練使得投手會加強球的轉速轉軸等因素，三振率每年持續增加，在 YEAR 以及 K/9 這兩個相關係數高達 0.39，是所有數據正相關最明顯的。BB/9 則是上面提到的長打，追求長打中，如果投手遇到很強的打者，則可能害怕被一棒全壘打造成重傷害，所以投的比較閃躲，所以近年來 BB/9 也有上升

Corr(YEAR,BB/9) = 0.77

Corr(YEAR,K/9) = 0.98

如果只拿 2015 statcast 系統被大量引進後的數據，就算資料量變少，但是使用進階數據預測的結果從 56%上升到了 57%



(二)什麼數據跟勝率有最大的關聯

我拿勝率[WPER]這個特徵進去時就發現這是原本與 TARGET 最有關係

的一個特徵，也就是說球隊當時的勝率還是最大的因素，所以我就想要進一步的去了解什麼數據跟年度勝率最有關係(取 2011~2021)

取傳統數據來看，打者方面跟魔球裡面一樣是 OBP 正相關最大，不過近幾年棒球更強調進階數去像是 WOBA 和 WRC+，這兩者真的都比 OBP 再好一點。

投手方面因為傳統可能只會看一個投手的表現，而且像是先發投手會看吃的局數，跟球隊失分沒直接的關聯，只要能幫忙多投點局數並且 ERA 低一點就是好投手，看相關係數的圖 ERA 負最多，但 ERA 比較是結果論，失分少理論上勝率當然就高，總不可能說勝率高的隊伍是得分高失分少這樣子，所以除了 ERA 負相關最多的就是 FIP，當一隊投手 FIP 越低勝率就越高，而 FIP 其實就是 BB K HR 這三項加權起來，所以看的出來 FIP 為什麼近幾年會被發明出來且非常有用

(三)選擇年份日期問題

為什麼選擇 2011~2021，我比較開始看 MLB 數據大概從 2010 年代，加上想說剛好就蒐集個十年左右的數據，加上 2020 年沒有蒐集所以剛好十年。日期為什麼選擇 5 月開始到 9 月 25，大聯盟賽季大約都從 4 月初開始到 10/1 前後結束，選 5 月第一個是方便不用查每年什麼時候開季，最重要的因素是因為 4 月才剛開季，各個球隊狀況可能比較不穩定一點，所以就從 5 月開始預測，9 月 25 第一點是每年最後各個球隊如果打不進季後賽就可能開練兵，甚至是擺爛換取隔年好的選秀順位，所以撇除掉球季最後幾場的因素

(四)結果如何再進步

1. 其實我的 project 少了一個很重要的因素，就是該場的先發投手，先發投手再棒球比賽的勝負裡面占了極大的因素，強的投手再爛的球隊勝率也可能到 6 7 成，但是由於資料不好取得所以我就沒蒐集了，想說試試看撇除先發投手，以團隊平均數據去預測，我覺得如果可以加上這個因素大概至少可以再多個 5%以上的準確度
2. 每一場先發打者也不一樣，同一隊裡有可能有主力打者前一天受傷沒辦法打，這樣拿前幾天的數據做參考價值就沒那麼大了，但是要追溯這麼

久以前的紀錄根本不太可能，或是說要花非常多的時間，所以這部份就期待之後有可以去嘗試

3. 2015 年來正式引進 STATCAST 系統，這個系統可以更加的去預測每個打者的打擊率，長打率，預期全壘打數，擊球初速、仰角、揮空率等，也可以撇除球場的因素，像是 COL 的主場因為海拔高空氣稀薄，所以擊出去的求受空氣阻力小，會飛的比較遠，而 STATCAST 可以幫忙撇除掉這些球場因素。如果有這些更完善的數據一定會讓準確度更加提升

七、 感想

準確度的部分，礙於爬蟲蒐集資料不是那麼容易的蒐集，所以像是先發投手，傷兵情況，更多的 statcast 數據很難蒐集到，所以準確度能到 57%我覺得還算可以接受，我也試著拿到 2022 五月到現在(2022/06/15)的比賽當作 testing data，結果是令人滿意的 0.616

```
0.6167400881057269
```

不過說真的這裡只有幾百筆測試資料，希望等我未來 model 修好一點後可以有更好的準確度(我覺得 test data 夠多的話上限大概 67%左右)

做完這次作業讓我對分析棒球數據更有感覺了，以前看都只能用感覺，說出來的結果或許是正確的，但是都沒憑沒據，做完這次 project 之後也可以更有系統性地去分析