答辩

周緬緬2023/1/5

1）數據來源

- 資料來源:
  - 主要架構參考他人API(https://github.com/yuetsu001/cpbl-crawler.git)
  - 根據網頁原始碼進行改寫
    1) 取得RequestVerificationToken
    2) 取得GameDetailJson/BattingJson/PitchingJson，並將 JSON 的資料型態轉換為 Python 的資料型態
    3) 儲存為CSV檔

view-source:https://cpbl.com.tw/box/index?gameSno=154&year=2022&kindCode=A

```
253            </div>
254        </div>
255
256 <form action="/box/live" id="MainForm" method="post"><input name="__RequestVerificationToken" type="hidden" value="f0fYejCsSshcLm86RZ1l3dAMmCZRwa6rWK22zjSb27Kd0bc6ZetXSAlOmlVqmDzTqtQmB1p8DzAdZ6W1gmTTJQb07LI1" /><inpu
257            <div class="GameSearch FormElmt">
258                <div class="item game_type">
259                    <select v-model="SelectKindCode" id="SelectKindCode" name="SelectKindCode" v-on:change="changeKindCode()">
260                        <option :value="KindCodeOpt.Value" v-for="KindCodeOpt in KindCodeOpts">{{KindCodeOpt.Text}}</option>
261                    </select>
262                </div>
```

view-source:https://cpbl.com.tw/box/index?gameSno=154&year=2022&kindCode=A

```
2770            }
2771
2772            //到後台action(api)撈資料
2773            $.ajax({
2774                url: "/box/getlive",
2775                type: "post",
2776                data: $("form[id='MainForm']").serialize(),
2777                success: function (result) {
2778                    if (result.Success) {
2779
2780                        _this.GameDetails = JSON.parse(result.GameDetailJson)
2781                        _this.Scoreboards = JSON.parse(result.ScoreboardJson)
2782                        _this.LiveLogs = JSON.parse(result.LiveLogJson)
2783                        _this.Battings = JSON.parse(result.BattingJson)
2784                        _this.Pitchings = JSON.parse(result.PitchingJson)
2785                        _this.FirstSnos = JSON.parse(result.FirstSnoJson)
2786                        _this.curtGameDetail = JSON.parse(result.CurtGameDetailJson)
```

BP_pytorchV8.py | CPBL.py | datatype.py

```python
396     def get_keys(self, url, regex):
397         res = self.ses.post(url, verify=False)
398         for l in res.text.split('\n'):
399             if 'RequestVerificationToken' in l:
400                 pattern = re.compile(regex, re.S)
401                 keys = pattern.findall(l)
402                 tuple_keys = (keys[0][0], keys[0][1], keys[0][2])
403                 self.token = tuple_keys[0]
404                 self.year = tuple_keys[2]
405                 self.gamesno = tuple_keys[1]
406
407     def get_data(self):
408         data = {}
409         # 需加入'defendStation'
410         res = self.ses.post(datatype.root + '/box/getlive',
411                             params={'__RequestVerificationToken': self.token,
412                                     'GameSno': self.gamesno,
413                                     'KindCode': 'A',
414                                     'Year': self.year}
415                             )
416
417         data['pitching'] = json.loads(res.json()['PitchingJson'])
418         ...
```

1）數據預處理方式

1. 先合併各年數據（2015-2022，每年方式爬取)
2. 累計部分數據（變數分為單場與累計）與計算單場之進階數據
3. 有些變數屬於類別變數(Dummy Variable)則使用One-hot encoding
4. 數據標準化：使用min-max標準化（歸一化）

BP_pytorchV8.py | CPBL.py | datatype.py

```python
# 轉換原Excel資料
def TransformData(OriginData):
    Union = pd.read_csv(OriginData, delimiter=",")
    Union['IntPitcherHabit'] = Union['PitcherHabit'].astype(int)
    Union['IntBrother.PitcherHabit'] = Union['Brother.PitcherHabit'].astype(int)
    Union['IntVisitingHomeType'] = Union['VisitingHomeType'].astype(int)
    Union['IntWinLoss'] = Union['WinLoss'].astype(int)
    Union = pd.concat([Union,
                       pd.get_dummies(Union['DayNight'], prefix='DayNight'),
                       pd.get_dummies(Union['IntPitcherHabit'], prefix='PitcherHabit'),
                       pd.get_dummies(Union['IntBrother.PitcherHabit'], prefix='Brother.PitcherHabit'),
                       pd.get_dummies(Union['IntVisitingHomeType'], prefix='VisitingHomeType'),
                       pd.get_dummies(Union['IntWinLoss'], prefix='WinLoss')
                       ], axis=1)

```

BP_pytorchV8.py | CPBL.py | datatype.py

```python

    def normalized(self, x_data, y_data):
        e = 1e-7  # 防止出现0
        for i in range(x_data.shape[1]):
            max_num = np.max(x_data[:, i])
            min_num = np.min(x_data[:, i])
            x_data[:, i] = (x_data[:, i] - min_num + e) / (max_num - min_num + e)
        y_data = (y_data - np.min(y_data) + e) / (np.max(y_data) - np.min(y_data) + e)
        return x_data, y_data
```

2）特徵（Features）選擇方式

- 普通數據，參閱《建構美國職棒大聯盟的勝隊預測模式：以人工類神經網路方式》之變數作為特徵。
- 進階數據，參閱「野球革命」中對於進階數據的定義與計算。

重點需要（爬蟲後）原始數據中有相關數據！！

3）模型（Model）選擇的依據

- 參閱《利用資料探勘技術建立運動彩券投注之預測模式-以職業棒球聯盟為例》，以BPN(倒傳遞神經網絡)與logit model(羅吉斯迴歸分析)進行預測，為BPN(倒傳遞神經網絡)的預測為最佳。
- 參閱《建構美國職棒大聯盟的勝隊預測模式：以人工類神經網路方式》，同上以BPN(倒傳遞神經網絡)的預測為最佳。
- 參閱《中華職棒球隊勝負預測模型之建立》:以Stepwise(逐步迴歸分析法)、 logit model(羅吉斯迴歸分析)、 BPN(倒傳遞神經網絡)、CART決策樹法、決策樹輔助之BPN模型，5種模型進行預測，最終以決策樹輔助之BPN模型為最佳預測模型。

4）查準率（Precision）有點低

- 在本研究中，主要是考慮輸贏的狀況(以運彩營運模式)。而針對TimeSeriesSplit的切割方式，分為4個Fold，每個驗證之數據集類別數量並沒有存在顯著差異，故選擇用「準確率」進行評估。

		統一隊贏	統一隊輸	平局	總計
Split1		23	28	1	52
Split2	Fold1	25	25	2	52
Split3	Fold2	31	20	1	52
Split4	Fold3	19	32	1	52
Split5	Fold4	22	26	4	52
總計		120	131	9	260

4）查準率（Precision）有點低

- 查準率(Precision) ： 70%
- 召回率(Recall) ： 100%
- 而準確率(Accuracy)： 82.3529%

	真實勝	真實敗
預測勝	7	3
預測敗	0	7

有關於查準率(Precision)有點低的問題，未來可能試圖將準確率改為，F1 –Score，$\boldsymbol{F1 = 2 / ((1/ Precision) + (1/ Recall))}$，來調和查準率與召回率，看是否可以改善這部分的問題。

5）最好有兩個或兩個以上模型的對比

- 考量到自身能力與時間，透過python建立一個模型，是目前能力所及。

- 老師的建議，是後續可以去探索的，我查閱到有一篇2011年論文《資料採礦於職業棒球勝隊預測模式之建構》，是透過各種分析軟體進行鑑別分析(DA)、羅吉斯迴歸(LR)、人工神經網絡(ANNs)、多元適應性雲性迴歸(MARS)、支援向量機(SVM模式)，最終多元適應性雲形迴歸不但分別具有最高的正確判別率且可挑選出重要變數。

6）最終的結論

本次專題，是想透過Python預測出比賽的輸贏兩種狀況(運彩中的不讓分情況)，透過客觀的比賽數據，來推敲出下一場比賽誰勝誰負，以利投注方向。其研究貢獻：

1. 以爬蟲方式爬取中華職棒的數據。
2. 變數加入進階數據。
3. 考慮含有時序的資料。
4. 以Python方式來訓練BPN模型。