Project 3

DNN 明星賽預測

學生：　　徐嘉澤
指導老師：　邱銘傳　老師

5W1H

WHAT

WHEN

WHO

WHERE

WHY

HOW

當NBA賽季開打時，部分專家抑或是我們這些全球各地的觀眾，都會因為自身偏好，期望自己喜歡的球員能被選上明星賽備受重視。接下來將由DNN深度學習的方式藉由官方提供之球員數據進行明星賽預測

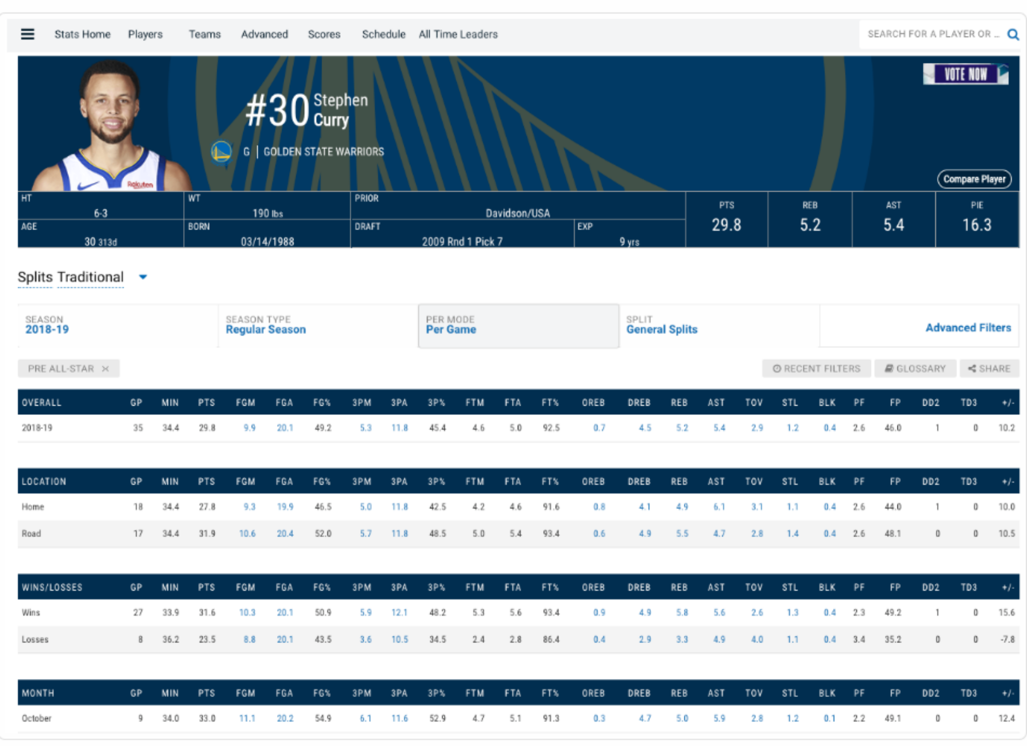
CONTENT

- DATA COLLECTING
- RAW ANALYSIS
- DNN MODEL
- PREDICTION
- CONCLUSION
- Future Work

PART ONE

DATA COLLECTING

DATA COLLECTING



#30 Stephen Curry

G | GOLDEN STATE WARRIORS

HT 6-3
WT 190 lbs
PRIOR Davidson/USA
AGE 30 313d
BORN 03/14/1988
DRAFT 2009 Rnd 1 Pick 7
EXP 9 yrs

PTS	REB	AST	PIE
29.8	5.2	5.4	16.3

Splits Traditional

SEASON 2018-19
SEASON TYPE Regular Season
PER MODE Per Game
SPLIT General Splits
Advanced Filters

PRE ALL-STAR ×

RECENT FILTERS GLOSSARY SHARE

OVERALL	GP	MIN	PTS	FGM	FGA	FG%	3PM	3PA	3P%	FTM	FTA	FT%	OREB	DREB	REB	AST	TOV	STL	BLK	PF	FP	DD2	TD3	+/-
2018-19	35	34.4	29.8	9.9	20.1	49.2	5.3	11.8	45.4	4.6	5.0	92.5	0.7	4.5	5.2	5.4	2.9	1.2	0.4	2.6	46.0	1	0	10.2

LOCATION	GP	MIN	PTS	FGM	FGA	FG%	3PM	3PA	3P%	FTM	FTA	FT%	OREB	DREB	REB	AST	TOV	STL	BLK	PF	FP	DD2	TD3	+/-
Home	18	34.4	27.8	9.3	19.9	46.5	5.0	11.8	42.5	4.2	4.6	91.6	0.8	4.1	4.9	6.1	3.1	1.1	0.4	2.6	44.0	1	0	10.0
Road	17	34.4	31.9	10.6	20.4	52.0	5.7	11.8	48.5	5.0	5.4	93.4	0.6	4.9	5.5	4.7	2.8	1.4	0.4	2.6	48.1	0	0	10.5

WINS/LOSSES	GP	MIN	PTS	FGM	FGA	FG%	3PM	3PA	3P%	FTM	FTA	FT%	OREB	DREB	REB	AST	TOV	STL	BLK	PF	FP	DD2	TD3	+/-
Wins	27	33.9	31.6	10.3	20.1	50.9	5.9	12.1	48.2	5.3	5.6	93.4	0.9	4.9	5.8	5.6	2.6	1.3	0.4	2.3	49.2	1	0	15.6
Losses	8	36.2	23.5	8.8	20.1	43.5	3.6	10.5	34.5	2.4	2.8	86.4	0.4	2.9	3.3	4.9	4.0	1.1	0.4	3.4	35.2	0	0	-7.8

MONTH	GP	MIN	PTS	FGM	FGA	FG%	3PM	3PA	3P%	FTM	FTA	FT%	OREB	DREB	REB	AST	TOV	STL	BLK	PF	FP	DD2	TD3	+/-
October	9	34.0	33.0	11.1	20.2	54.9	6.1	11.6	52.9	4.7	5.1	91.3	0.3	4.7	5.0	5.9	2.8	1.2	0.1	2.2	49.1	0	0	12.4

障礙：

無現成數據

方法：

網路爬蟲（selenium）

DATA COLLECTING

Selenium 是為瀏覽器自動化（Browser Automation）需求所設計的一套工具集合，讓程式可以直接驅動瀏覽器進行各種網站操作。

前置作業

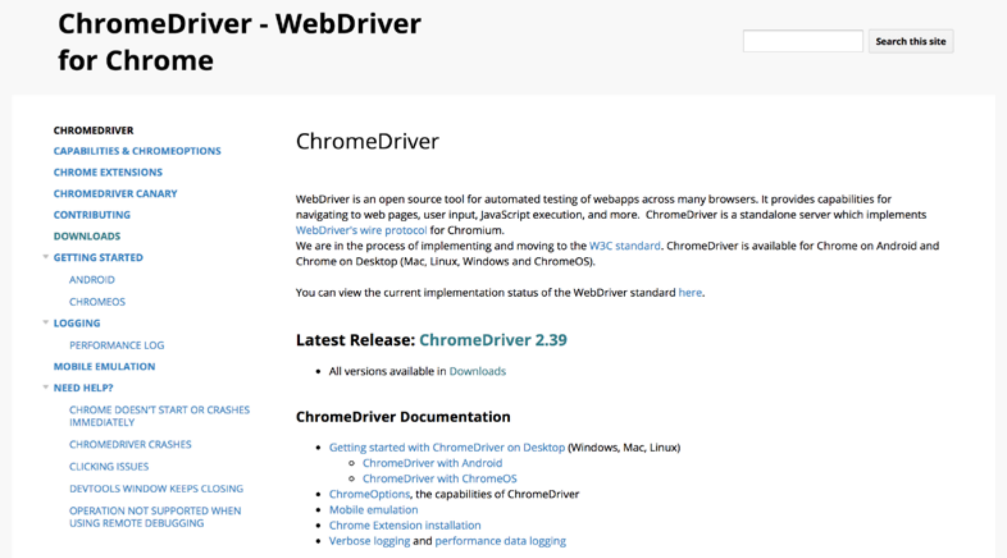
XPATH
定位

盤點動作

資料擷取

前置作業

下載瀏覽器驅動



ChromeDriver

WebDriver is an open source tool for automated testing of webapps across many browsers. It provides capabilities for navigating to web pages, user input, JavaScript execution, and more. ChromeDriver is a standalone server which implements WebDriver's wire protocol for Chromium.
We are in the process of implementing and moving to the W3C standard. ChromeDriver is available for Chrome on Android and Chrome on Desktop (Mac, Linux, Windows and ChromeOS).

You can view the current implementation status of the WebDriver standard here.

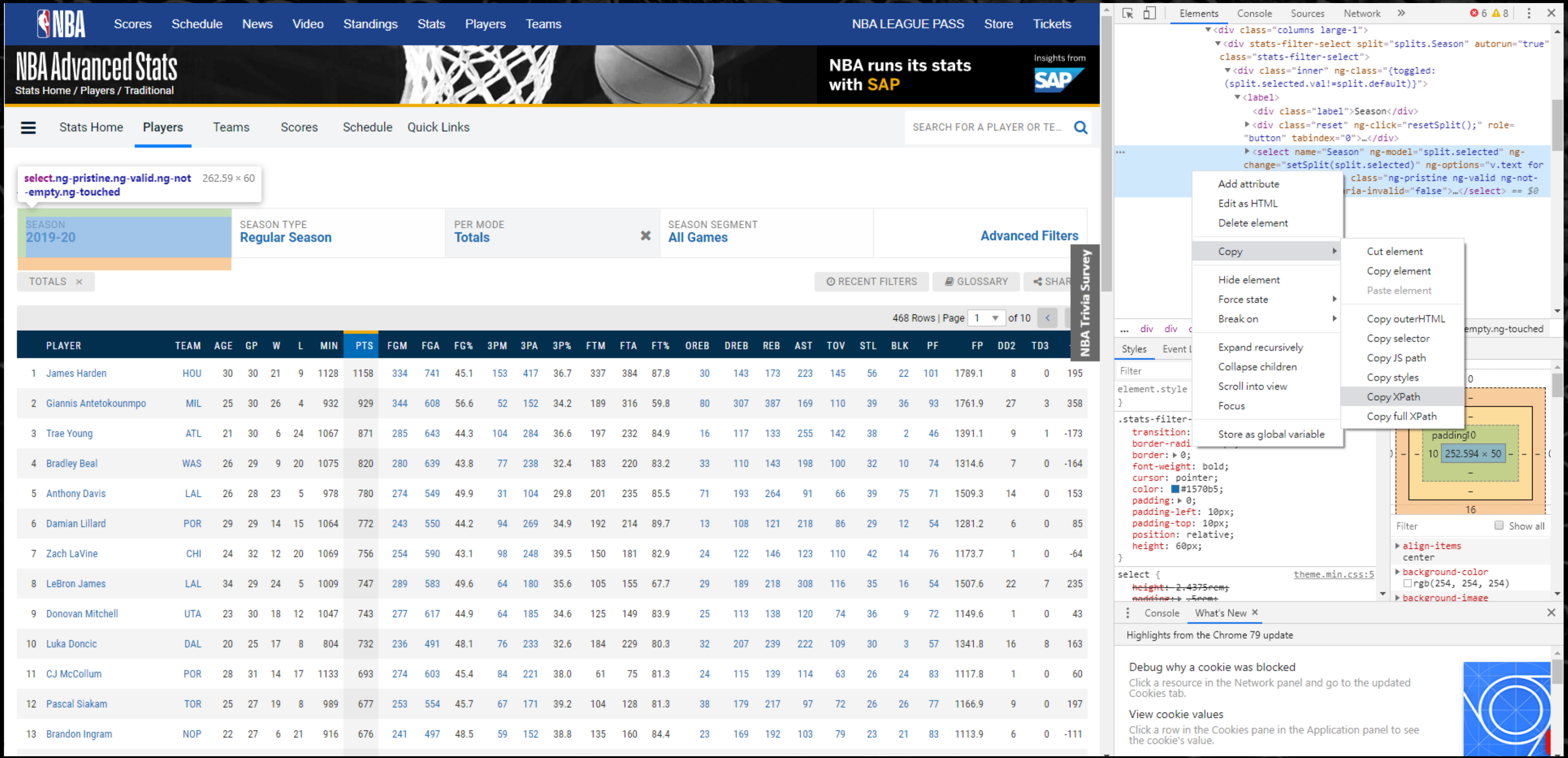
Latest Release: ChromeDriver 2.39

- All versions available in Downloads

ChromeDriver Documentation

- Getting started with ChromeDriver on Desktop (Windows, Mac, Linux)
  - ChromeDriver with Android
  - ChromeDriver with ChromeOS
- ChromeOptions, the capabilities of ChromeDriver
- Mobile emulation
- Chrome Extension installation
- Verbose logging and performance data logging

XPATH定位



SEASON 2019-20 | SEASON TYPE Regular Season | PER MODE Totals ✖ | SEASON SEGMENT All Games | Advanced Filters

468 Rows | Page 1 of 10

	PLAYER	TEAM	AGE	GP	W	L	MIN	PTS	FGM	FGA	FG%	3PM	3PA	3P%	FTM	FTA	FT%	OREB	DREB	REB	AST	TOV	STL	BLK	PF	FP	DD2	TD3	
1	James Harden	HOU	30	30	21	9	1128	1158	334	741	45.1	153	417	36.7	337	384	87.8	30	143	173	223	145	56	22	101	1789.1	8	0	195
2	Giannis Antetokounmpo	MIL	25	30	26	4	932	929	344	608	56.6	52	152	34.2	189	316	59.8	80	307	387	169	110	39	36	93	1761.9	27	3	358
3	Trae Young	ATL	21	30	6	24	1067	871	285	643	44.3	104	284	36.6	197	232	84.9	16	117	133	255	142	38	2	46	1391.1	9	1	-173
4	Bradley Beal	WAS	26	29	9	20	1075	820	280	639	43.8	77	238	32.4	183	220	83.2	33	110	143	198	100	32	10	74	1314.6	7	0	-164
5	Anthony Davis	LAL	26	28	23	5	978	780	274	549	49.9	31	104	29.8	201	235	85.5	71	193	264	91	66	39	75	71	1509.3	14	0	153
6	Damian Lillard	POR	29	29	14	15	1064	772	243	550	44.2	94	269	34.9	192	214	89.7	13	108	121	218	86	29	12	54	1281.2	6	0	85
7	Zach LaVine	CHI	24	32	12	20	1069	756	254	590	43.1	98	248	39.5	150	181	82.9	24	122	146	123	110	42	14	76	1173.7	1	0	-64
8	LeBron James	LAL	34	29	24	5	1009	747	289	583	49.6	64	180	35.6	105	155	67.7	29	189	218	308	116	35	16	54	1507.6	22	7	235
9	Donovan Mitchell	UTA	23	30	18	12	1047	743	277	617	44.9	64	185	34.6	125	149	83.9	25	113	138	120	74	36	9	72	1149.6	1	0	43
10	Luka Doncic	DAL	20	25	17	8	804	732	236	491	48.1	76	233	32.6	184	229	80.3	32	207	239	222	109	30	3	57	1341.8	16	8	163
11	CJ McCollum	POR	28	31	14	17	1133	693	274	603	45.4	84	221	38.0	61	75	81.3	24	115	139	114	63	26	24	83	1117.8	1	0	60
12	Pascal Siakam	TOR	25	27	19	8	989	677	253	554	45.7	67	171	39.2	104	128	81.3	38	179	217	97	72	26	26	77	1166.9	9	0	197
13	Brandon Ingram	NOP	22	27	6	21	916	676	241	497	48.5	59	152	38.8	135	160	84.4	23	169	192	103	79	23	21	83	1113.9	6	0	-111

```
▼<div class="columns large-1">
  ▼<div stats-filter-select split="splits.Season" autorun="true"
   class="stats-filter-select">
    ▼<div class="inner" ng-class="{toggled:
     (split.selected.val!=split.default)}">
      ▼<label>
         <div class="label">Season</div>
        ▶<div class="reset" ng-click="resetSplit();" role=
         "button" tabindex="0">…</div>
...     ▶<select name="Season" ng-model="split.selected" ng-
         change="setSplit(split.selected)" ng-options="v.text for
                               class="ng-pristine ng-valid ng-not-
                              ria-invalid="false">…</select> == $0
```

```
element.style {
}
.stats-filter-
  transition:
  border-radi
  border: ▶ 0;
  font-weight: bold;
  cursor: pointer;
  color: #1570b5;
  padding: ▶ 0;
  padding-left: 10px;
  padding-top: 10px;
  position: relative;
  height: 60px;
}
select {                          theme.min.css:5
  ~~height: 2.4375rem;~~
  ~~padding: ▶ .5rem;~~
```

```
▶ align-items
  center
▶ background-color
  □rgb(254, 254, 254)
▶ background-image
```

Highlights from the Chrome 79 update

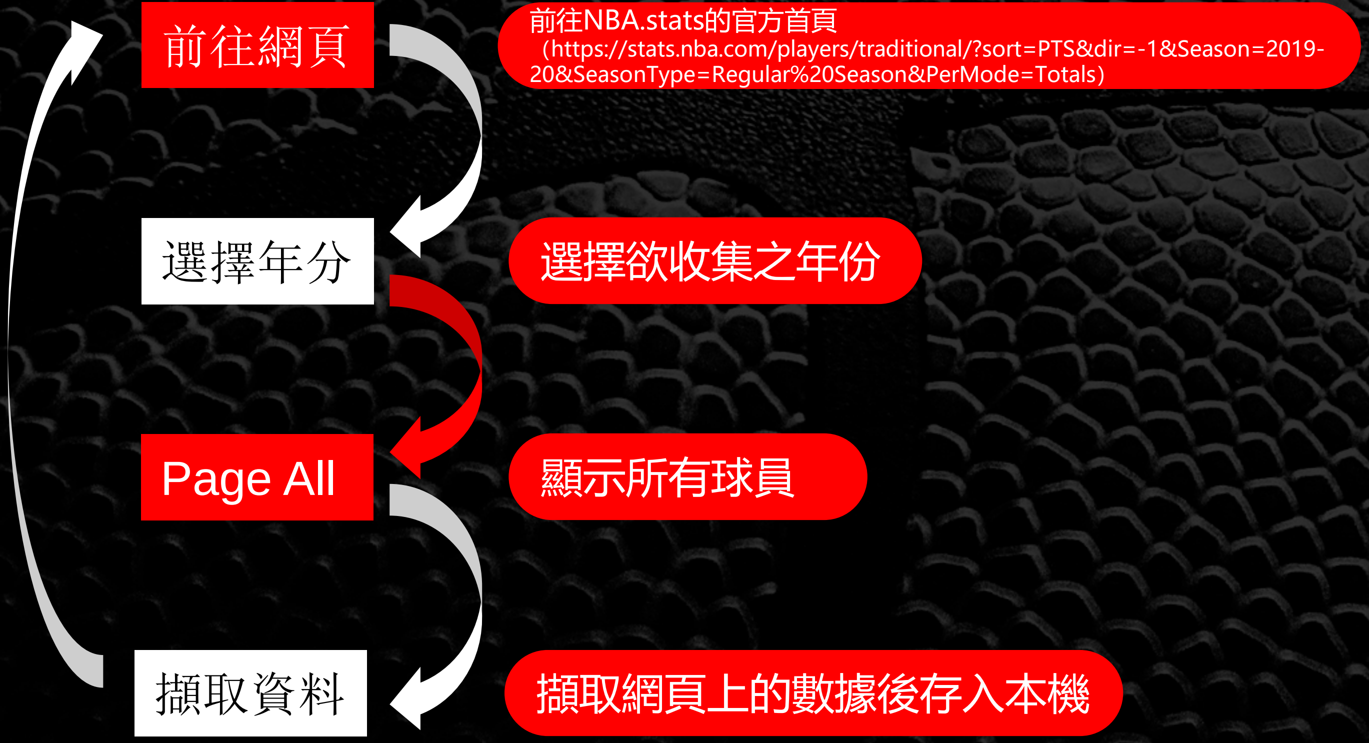
Debug why a cookie was blocked

Click a resource in the Network panel and go to the updated Cookies tab.

View cookie values

Click a row in the Cookies pane in the Application panel to see the cookie's value.

盤點動作



前往網頁

前往NBA.stats的官方首頁（https://stats.nba.com/players/traditional/?sort=PTS&dir=-1&Season=2019-20&SeasonType=Regular%20Season&PerMode=Totals）

選擇年分

選擇欲收集之年份

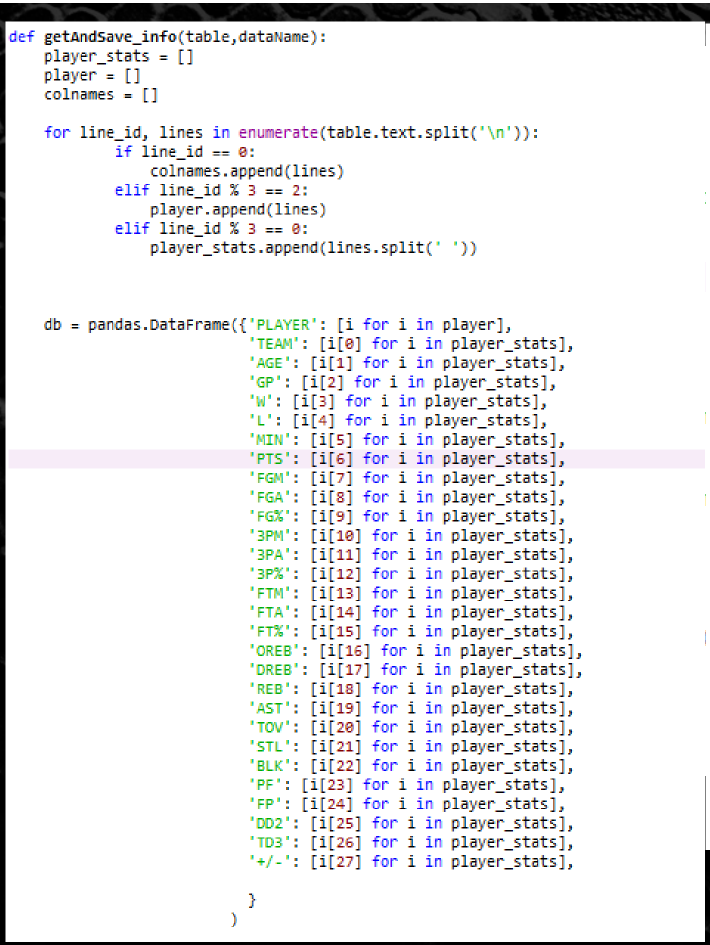
Page All

顯示所有球員

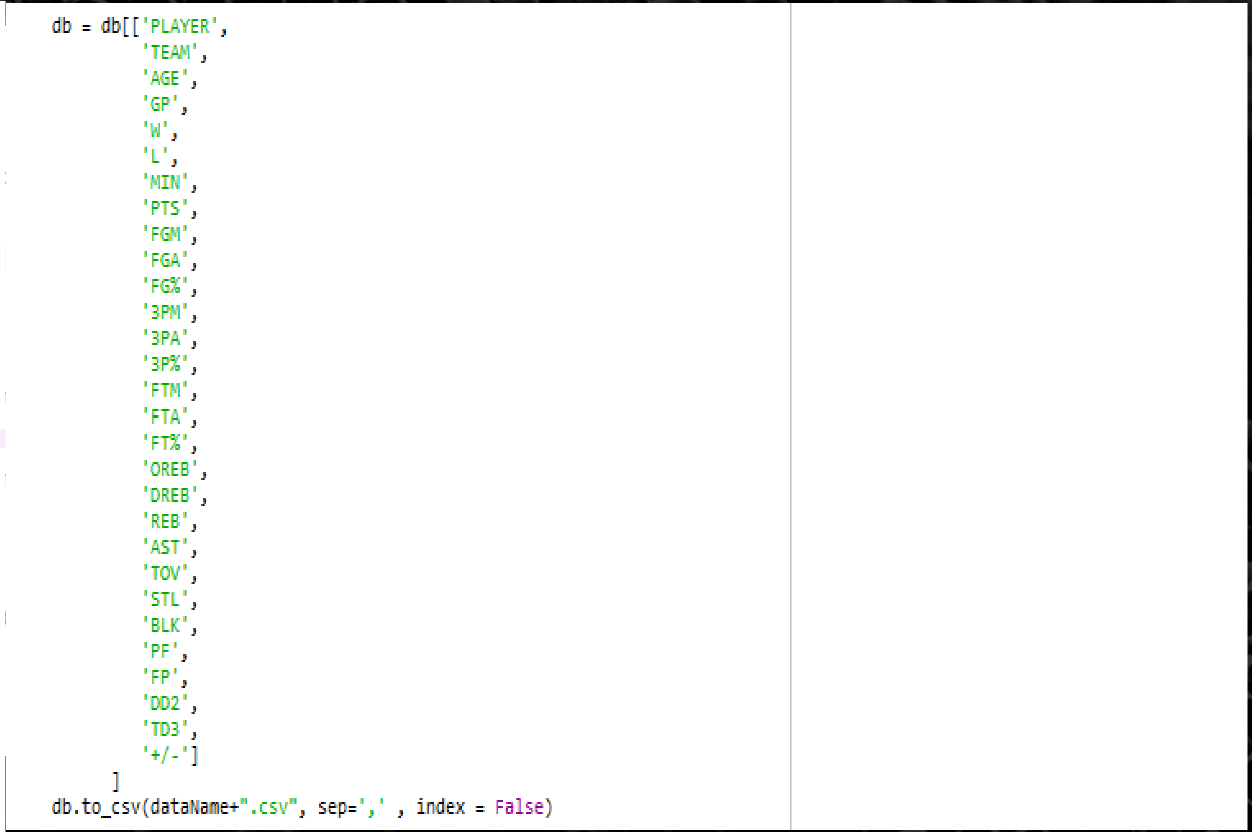
擷取資料

擷取網頁上的數據後存入本機

程式碼



```
def getAndSave_info(table,dataName):
    player_stats = []
    player = []
    colnames = []

    for line_id, lines in enumerate(table.text.split('\n')):
            if line_id == 0:
                colnames.append(lines)
            elif line_id % 3 == 2:
                player.append(lines)
            elif line_id % 3 == 0:
                player_stats.append(lines.split(' '))



    db = pandas.DataFrame({'PLAYER': [i for i in player],
                           'TEAM': [i[0] for i in player_stats],
                           'AGE': [i[1] for i in player_stats],
                           'GP': [i[2] for i in player_stats],
                           'W': [i[3] for i in player_stats],
                           'L': [i[4] for i in player_stats],
                           'MIN': [i[5] for i in player_stats],
                           'PTS': [i[6] for i in player_stats],
                           'FGM': [i[7] for i in player_stats],
                           'FGA': [i[8] for i in player_stats],
                           'FG%': [i[9] for i in player_stats],
                           '3PM': [i[10] for i in player_stats],
                           '3PA': [i[11] for i in player_stats],
                           '3P%': [i[12] for i in player_stats],
                           'FTM': [i[13] for i in player_stats],
                           'FTA': [i[14] for i in player_stats],
                           'FT%': [i[15] for i in player_stats],
                           'OREB': [i[16] for i in player_stats],
                           'DREB': [i[17] for i in player_stats],
                           'REB': [i[18] for i in player_stats],
                           'AST': [i[19] for i in player_stats],
                           'TOV': [i[20] for i in player_stats],
                           'STL': [i[21] for i in player_stats],
                           'BLK': [i[22] for i in player_stats],
                           'PF': [i[23] for i in player_stats],
                           'FP': [i[24] for i in player_stats],
                           'DD2': [i[25] for i in player_stats],
                           'TD3': [i[26] for i in player_stats],
                           '+/-': [i[27] for i in player_stats],

                           }
                         )
```

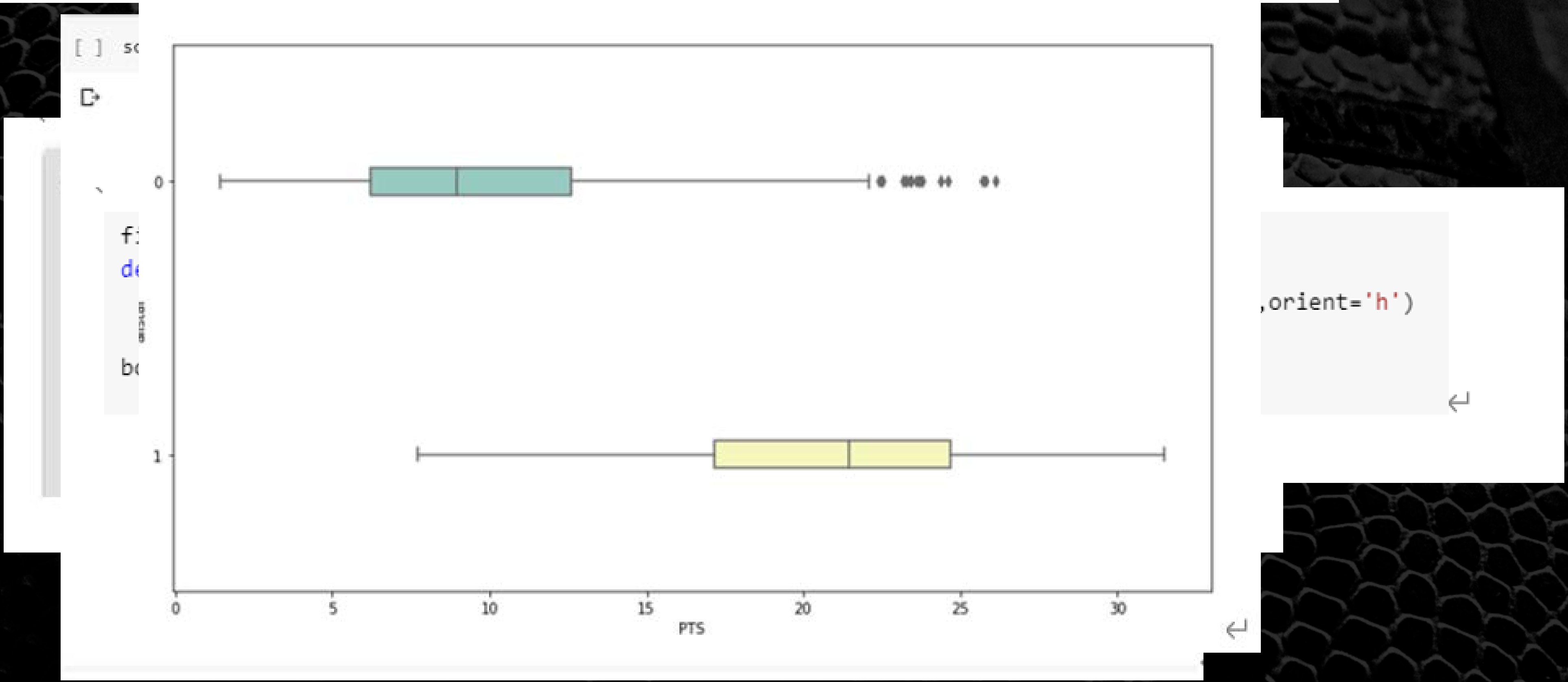


```
    db = db[['PLAYER',
             'TEAM',
             'AGE',
             'GP',
             'W',
             'L',
             'MIN',
             'PTS',
             'FGM',
             'FGA',
             'FG%',
             '3PM',
             '3PA',
             '3P%',
             'FTM',
             'FTA',
             'FT%',
             'OREB',
             'DREB',
             'REB',
             'AST',
             'TOV',
             'STL',
             'BLK',
             'PF',
             'FP',
             'DD2',
             'TD3',
             '+/-']
            ]
    db.to_csv(dataName+".csv", sep=',' , index = False)
```

PART TWO

RAW ANALYSIS

初步分析

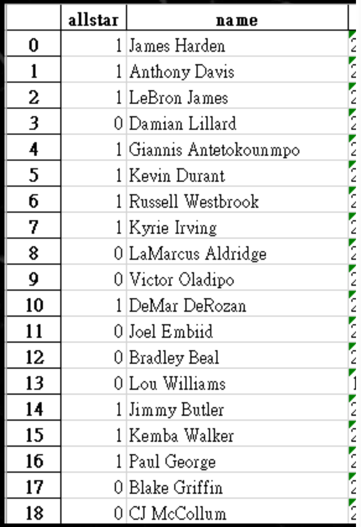


```
[ ] sc
fi
de
bo
,orient='h')
```

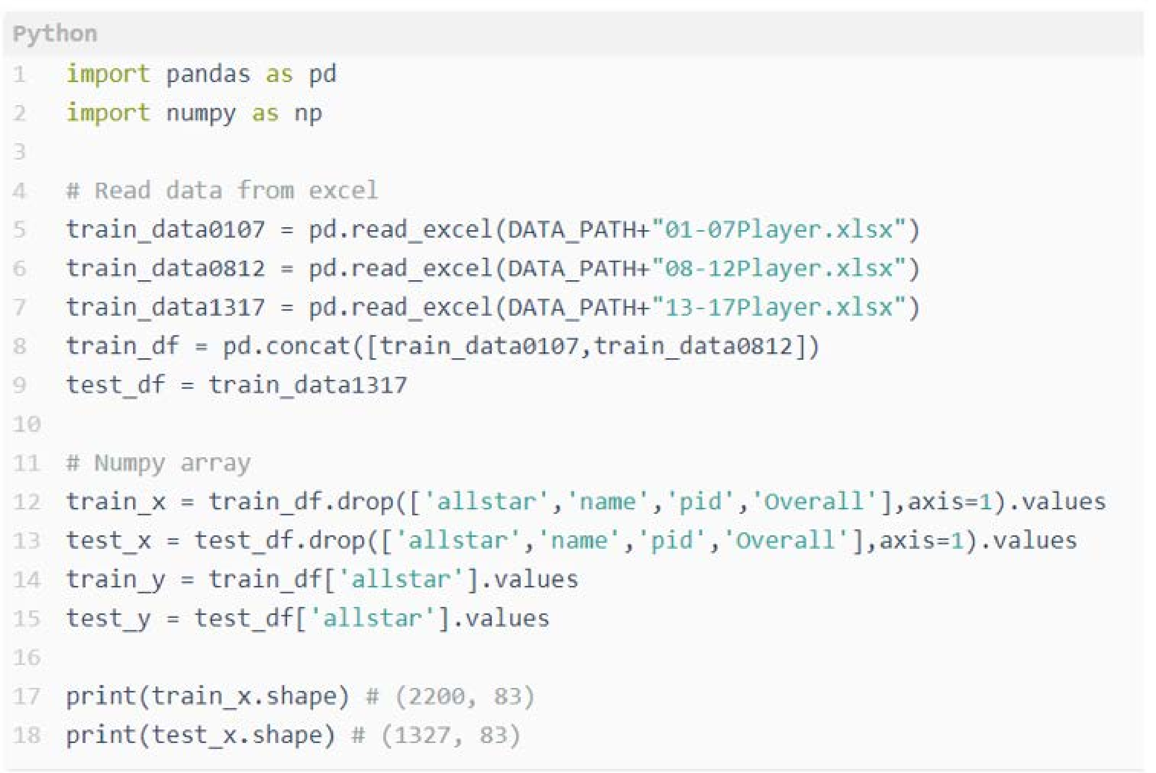
PART THREE

DNN MODEL

資料輸入



	allstar	name
0	1	James Harden
1	1	Anthony Davis
2	1	LeBron James
3	0	Damian Lillard
4	1	Giannis Antetokounmpo
5	1	Kevin Durant
6	1	Russell Westbrook
7	1	Kyrie Irving
8	0	LaMarcus Aldridge
9	0	Victor Oladipo
10	1	DeMar DeRozan
11	0	Joel Embiid
12	0	Bradley Beal
13	0	Lou Williams
14	1	Jimmy Butler
15	1	Kemba Walker
16	1	Paul George
17	0	Blake Griffin
18	0	CJ McCollum

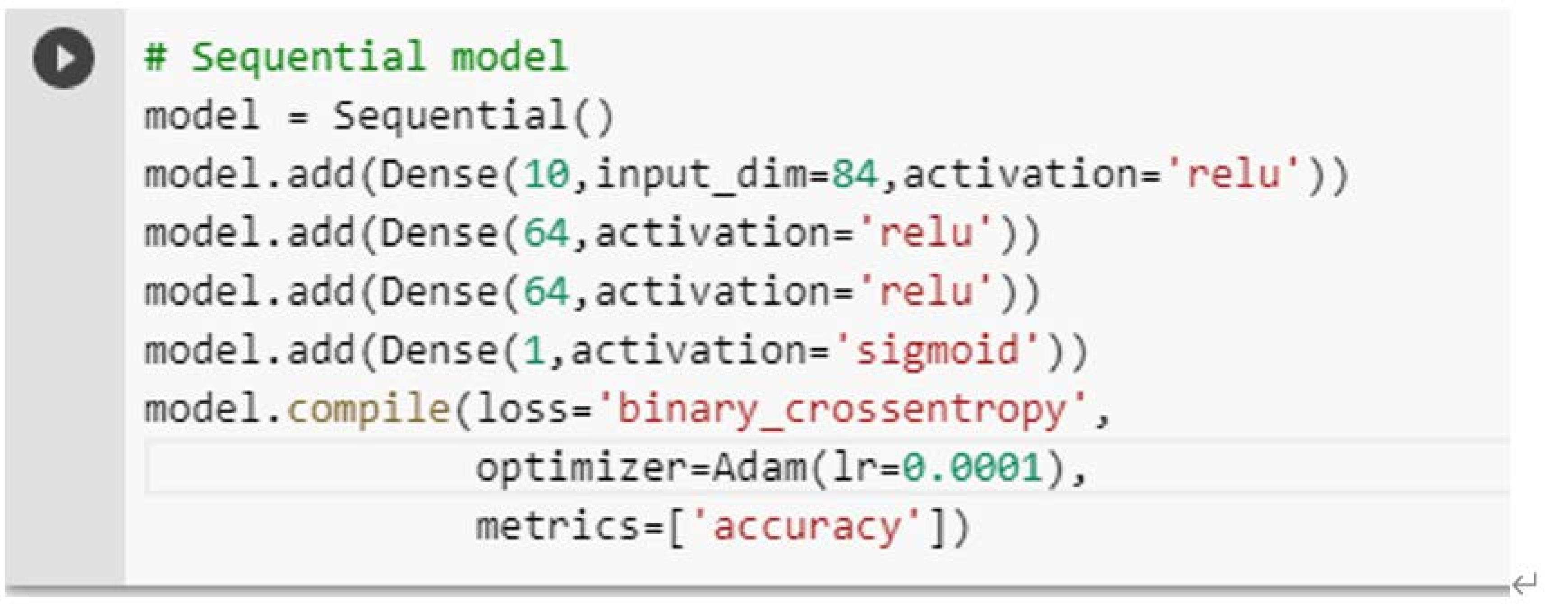


Python

```python
import pandas as pd
import numpy as np

# Read data from excel
train_data0107 = pd.read_excel(DATA_PATH+"01-07Player.xlsx")
train_data0812 = pd.read_excel(DATA_PATH+"08-12Player.xlsx")
train_data1317 = pd.read_excel(DATA_PATH+"13-17Player.xlsx")
train_df = pd.concat([train_data0107,train_data0812])
test_df = train_data1317

# Numpy array
train_x = train_df.drop(['allstar','name','pid','Overall'],axis=1).values
test_x = test_df.drop(['allstar','name','pid','Overall'],axis=1).values
train_y = train_df['allstar'].values
test_y = test_df['allstar'].values

print(train_x.shape) # (2200, 83)
print(test_x.shape) # (1327, 83)
```

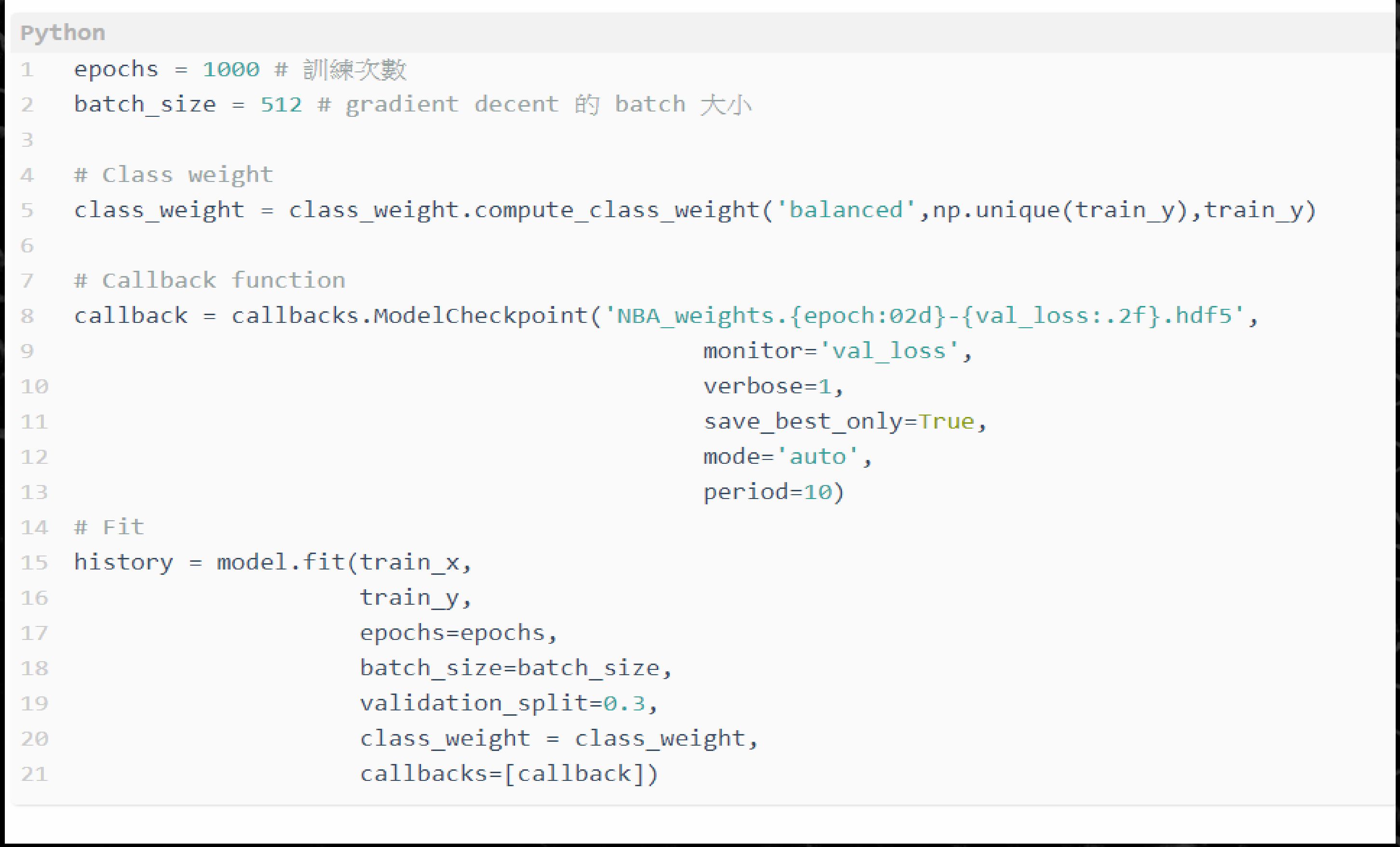
數據

DNN MODEL



```
# Sequential model
model = Sequential()
model.add(Dense(10,input_dim=84,activation='relu'))
model.add(Dense(64,activation='relu'))
model.add(Dense(64,activation='relu'))
model.add(Dense(1,activation='sigmoid'))
model.compile(loss='binary_crossentropy',
              optimizer=Adam(lr=0.0001),
              metrics=['accuracy'])
```

FIT



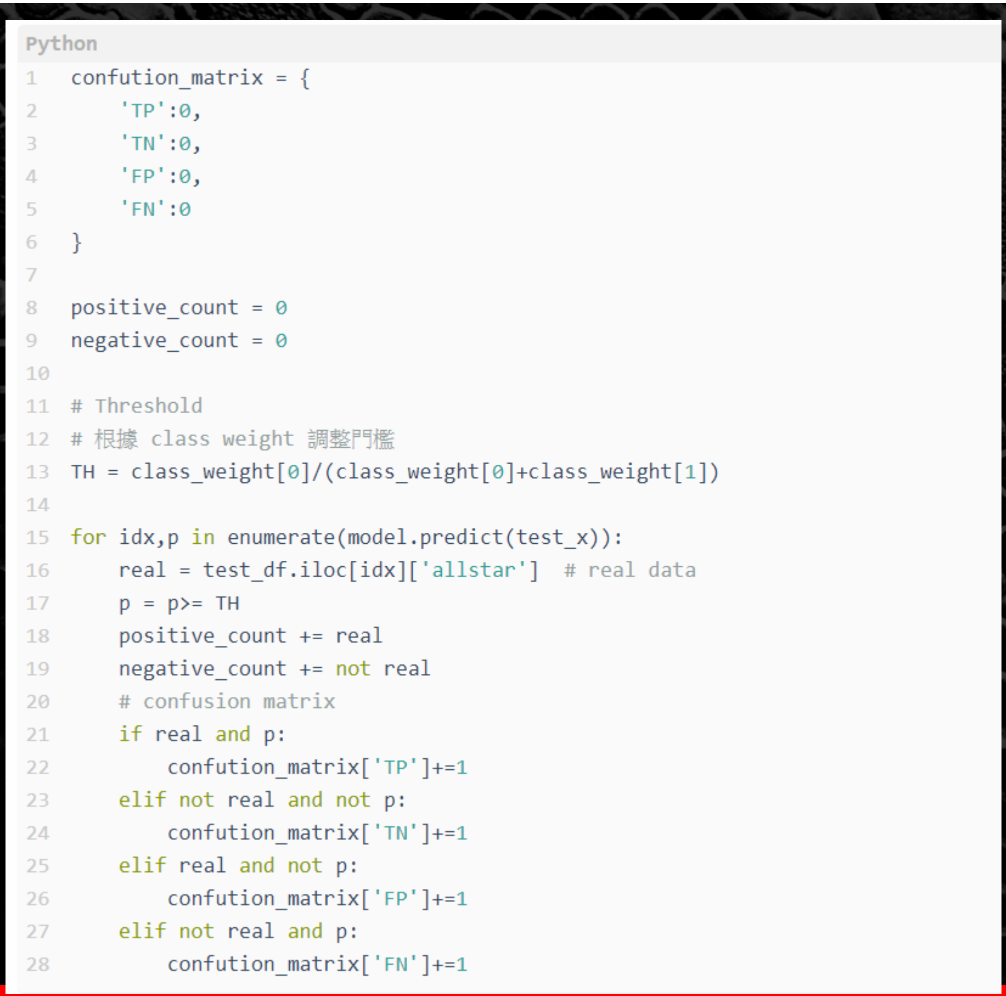
Python

```python
epochs = 1000 # 訓練次數
batch_size = 512 # gradient decent 的 batch 大小

# Class weight
class_weight = class_weight.compute_class_weight('balanced',np.unique(train_y),train_y)

# Callback function
callback = callbacks.ModelCheckpoint('NBA_weights.{epoch:02d}-{val_loss:.2f}.hdf5',
                                            monitor='val_loss',
                                            verbose=1,
                                            save_best_only=True,
                                            mode='auto',
                                            period=10)
# Fit
history = model.fit(train_x,
                    train_y,
                    epochs=epochs,
                    batch_size=batch_size,
                    validation_split=0.3,
                    class_weight = class_weight,
                    callbacks=[callback])
```

PART FOUR

PREDICTION

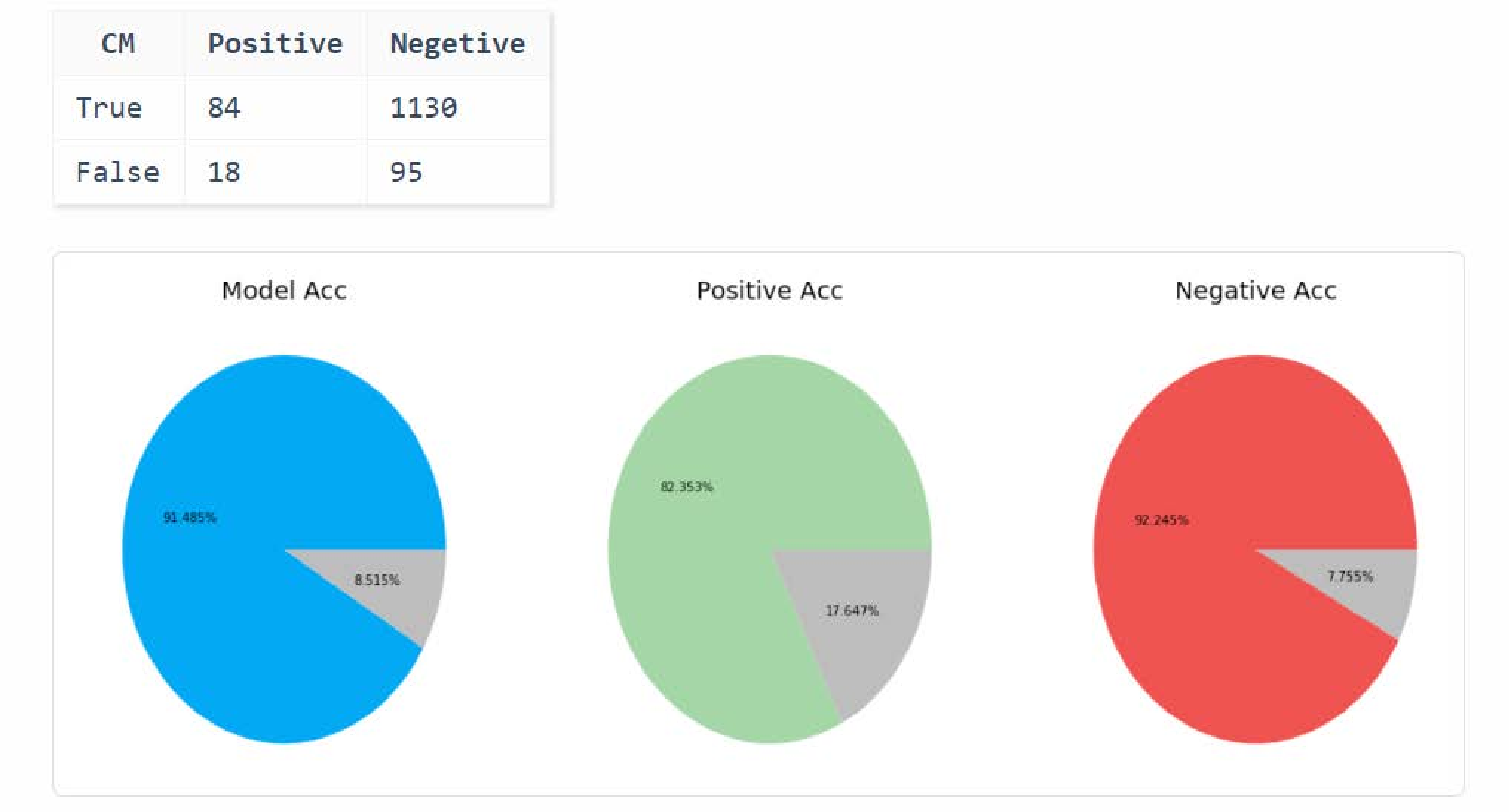
預測



Python

```python
confution_matrix = {
    'TP':0,
    'TN':0,
    'FP':0,
    'FN':0
}

positive_count = 0
negative_count = 0

# Threshold
# 根據 class weight 調整門檻
TH = class_weight[0]/(class_weight[0]+class_weight[1])

for idx,p in enumerate(model.predict(test_x)):
    real = test_df.iloc[idx]['allstar']  # real data
    p = p>= TH
    positive_count += real
    negative_count += not real
    # confusion matrix
    if real and p:
        confution_matrix['TP']+=1
    elif not real and not p:
        confution_matrix['TN']+=1
    elif real and not p:
        confution_matrix['FP']+=1
    elif not real and p:
        confution_matrix['FN']+=1
```

結果



CM	Positive	Negetive
True	84	1130
False	18	95

Model Acc

Positive Acc

Negative Acc

PART FIVE

CONCLUSION

結論

- 投票意向和球員場上成績有關
- 有些無法從球員成績看出的因素，使得準確度不容易達到非常高
  - ■ 球員的名氣（老將可能成績較差但仍受觀眾喜愛）
  - ■ 球員私人行為（發表歧視言論等等）
  - ■ 作弊手段得到數據
  - ■ 球員所在的隊伍

PART SIX

Future Work

Future Work

THANKS