

Final Report

for Python programming



班級：電機二乙

學號：B1141065

姓名：陳昱安

Problem 1：

請嘗試撰寫一個有關於預測大樂透開獎結果的模擬器之程式，其中抽號規則如下：

1. 共有 01，02...49 號等 49 顆號碼球。
2. 從中優先去選擇 6 顆球作為抽球號碼，另外再選一顆作為特別號（不能重複）
3. 最後在開獎之時，如有順序和相關的數字號碼相同即有機會得獎

（※可將其預測結果匯入一個 csv 檔案之中，以進行資料儲存。）



（圖一）大樂透活動宣傳

【解】：

〈法一〉：統計和隨機分析法

〈主要程式〉

```
import random
import pandas as pd
from collections import Counter

# 讀取CSV文件
file_path = '/content/drive/MyDrive/大樂透/大樂透歷史記錄 .csv' # 確保檔案路徑正確
afile = pd.read_csv(file_path)
file_path2 = '/content/drive/MyDrive/大樂透/guojing0805.csv'

# 提取中獎號碼和特別號
draws = []
special_numbers = []
# 使用 iloc 從索引 0 開始遍歷
for i in range(len(afile)):
    # 提取每行的中獎號碼和特別號
    numbers = [afile.iloc[i]['num1'], afile.iloc[i]['num2'], afile.iloc[i]['num3'],
               afile.iloc[i]['num4'], afile.iloc[i]['num5'], afile.iloc[i]['num6']]
    special_number = afile.iloc[i]['特別號']
    draws.append(numbers)
    special_numbers.append(special_number)

# 轉換成DataFrame
df = pd.DataFrame(draws, columns=["Num1", "Num2", "Num3", "Num4", "Num5", "Num6"])
df["Special"] = special_numbers

# 找出每個號碼出現的頻率
all_numbers = [num for draw in draws for num in draw]
number_frequency = Counter(all_numbers)
special_number_frequency = Counter(special_numbers)

# 根據頻率選擇前6個最常出現的號碼作為預測
most_common_numbers = [num for num, count in number_frequency.most_common(6)]
most_common_special = special_number_frequency.most_common(1)[0][0]
```

(圖二)模擬大樂透抽號預測系統程式示意圖 (1)


```

# 另外一種策略是隨機選擇未出現在前6個最常見號碼的號碼
remaining_numbers = [num for num in range(1, 50) if num not in most_common_numbers]
random_choice = random.sample(remaining_numbers, 6)
remaining_special_numbers = [num for num in range(1, 50) if num not in special_numbers]

# 如果 remaining_special_numbers 是空的, 從所有可能的號碼中選擇
if not remaining_special_numbers:
    remaining_special_numbers = [num for num in range(1, 50)]

random_special_choice = random.choice(remaining_special_numbers)

# 完全隨機生成 6 個普通號碼和一個特別號
def generate_random_numbers():
    normal_numbers = random.sample(range(1, 50), 6)
    special_number = random.choice(range(1, 50))
    return normal_numbers, special_number

random_prediction_numbers, random_prediction_special = generate_random_numbers()

def compare_to_actual(predicted_numbers, predicted_special, actual_numbers, actual_special):
    correct_numbers = set(predicted_numbers) & set(actual_numbers)
    correct_special = 1 if predicted_special in actual_special else 0
    return len(correct_numbers), correct_special

def calculate_accuracy(predicted_numbers, predicted_special, actual_numbers, actual_special):
    correct_count_numbers, correct_special = compare_to_actual(predicted_numbers, predicted_special, actual_numbers, actual_special)
    accuracy_numbers = (correct_count_numbers / 6) * 100
    accuracy_special = int(correct_special) * 100
    return accuracy_numbers, accuracy_special

# 歷史中獎號碼
historical_numbers = [num for draw in draws for num in draw]
historical_special_numbers = special_numbers

```

(圖三) 模擬大樂透抽號預測系統程式示意圖 (2)

```

# 計算三種方式的準確率
accuracy_most_common_numbers, accuracy_most_common_special = calculate_accuracy(most_common_numbers, most_common_special, historical_numbers, historical_special_numbers)
accuracy_random_choice_numbers, accuracy_random_choice_special = calculate_accuracy(random_choice, random_special_choice, historical_numbers, historical_special_numbers)
accuracy_completely_random_numbers, accuracy_completely_random_special = calculate_accuracy(random_prediction_numbers, random_prediction_special, historical_numbers, historical_special_numbers)

# 建立預測結果的DataFrame
predicted_df = pd.DataFrame({
    "Method": ["熱門抽號法", "過去隨機法", "完全隨機法"],
    "Num1": [sorted(most_common_numbers)[0], sorted(random_choice)[0], sorted(random_prediction_numbers)[0]],
    "Num2": [sorted(most_common_numbers)[1], sorted(random_choice)[1], sorted(random_prediction_numbers)[1]],
    "Num3": [sorted(most_common_numbers)[2], sorted(random_choice)[2], sorted(random_prediction_numbers)[2]],
    "Num4": [sorted(most_common_numbers)[3], sorted(random_choice)[3], sorted(random_prediction_numbers)[3]],
    "Num5": [sorted(most_common_numbers)[4], sorted(random_choice)[4], sorted(random_prediction_numbers)[4]],
    "Num6": [sorted(most_common_numbers)[5], sorted(random_choice)[5], sorted(random_prediction_numbers)[5]],
    "Special": [most_common_special, random_special_choice, random_prediction_special],
    "Accuracy Numbers (%)": [accuracy_most_common_numbers, accuracy_random_choice_numbers, accuracy_completely_random_numbers],
    "Accuracy Special (%)": [accuracy_most_common_special, accuracy_random_choice_special, accuracy_completely_random_special]
})

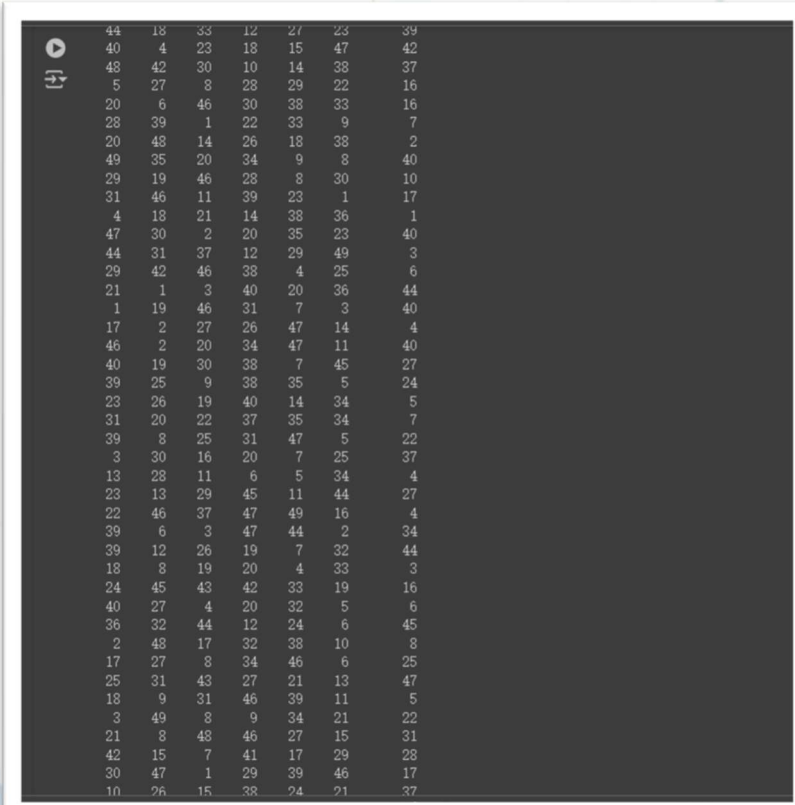
# 列印歷史資料表格
print("歷史中獎數據:")
print(df.to_string(index=False))

# 列印預測結果表格
print("\n預測結果:")
print(predicted_df.to_string(index=False))
# 將預測結果寫入新的CSV文件
predicted_df.to_csv(file_path2, index=False)

```

(圖四) 模擬大樂透抽號預測系統程式示意圖 (3)

〈呈現結果〉



44	18	33	12	27	23	39
40	4	23	18	15	47	42
48	42	30	10	14	38	37
5	27	8	28	29	22	16
20	6	46	30	38	33	16
28	39	1	22	33	9	7
20	48	14	26	18	38	2
49	35	20	34	9	8	40
29	19	46	28	8	30	10
31	46	11	39	23	1	17
4	18	21	14	38	36	1
47	30	2	20	35	23	40
44	31	37	12	29	49	3
29	42	46	38	4	25	6
21	1	3	40	20	36	44
1	19	46	31	7	3	40
17	2	27	26	47	14	4
46	2	20	34	47	11	40
40	19	30	38	7	45	27
39	25	9	38	35	5	24
23	26	19	40	14	34	5
31	20	22	37	35	34	7
39	8	25	31	47	5	22
3	30	16	20	7	25	37
13	28	11	6	5	34	4
23	13	29	45	11	44	27
22	46	37	47	49	16	4
39	6	3	47	44	2	34
39	12	26	19	7	32	44
18	8	19	20	4	33	3
24	45	43	42	33	19	16
40	27	4	20	32	5	6
36	32	44	12	24	6	45
2	48	17	32	38	10	8
17	27	8	34	46	6	25
25	31	43	27	21	13	47
18	9	31	46	39	11	5
3	49	8	9	34	21	22
21	8	48	46	27	15	31
42	15	7	41	17	29	28
30	47	1	29	39	46	17
10	26	15	38	24	21	37

(圖五) 模擬大樂透抽號預測系統結果呈現圖 (1)

11

39	25	9	38	35	5	24
23	26	19	40	14	34	5
31	20	22	37	35	34	7
39	8	25	31	47	5	22
3	30	16	20	7	25	37
13	28	11	6	5	34	4
23	13	29	45	11	44	27
22	46	37	47	49	16	4
39	6	3	47	44	2	34
39	12	26	19	7	32	44
18	8	19	20	4	33	3
24	45	43	42	33	19	16
40	27	4	20	32	5	6
36	32	44	12	24	6	45
2	48	17	32	38	10	8
17	27	8	34	46	6	25
Method	31	43	27	21	13	47
18	9	31	46	39	11	5
3	49	8	9	34	21	22
21	8	48	46	27	15	31
42	15	7	41	17	29	28
30	47	1	29	39	46	17
10	26	15	38	24	21	37
40	26	29	27	7	36	32
22	3	49	45	27	13	17
37	43	18	40	33	8	31
21	19	36	14	8	20	28
48	4	38	40	41	16	35
6	21	46	19	32	29	11
13	3	43	15	48	2	1
46	49	38	39	27	32	24
4	31	48	38	40	35	18
22	26	3	17	16	19	1

預測結果:

Method	Num1	Num2	Num3	Num4	Num5	Num6	Special	Accuracy	Numbers (%)	Accuracy Special (%)
熱門抽籤法	1	2	8	15	23	43	41	100.0	100	
過去隨機法	4	32	36	38	41	48	11	100.0	100	
完全隨機法	6	7	11	13	23	28	21	100.0	100	

(圖六) 模擬大樂透抽號預測系統結果呈現圖 (2)

筆記本 guojing0805.csv X

1 to 3 of 3 entries Filter

Method	Num1	Num2	Num3	Num4	Num5	Num6	Special	Accuracy Numbers (%)	Accuracy Special (%)
熱門抽號法	1	2	8	15	23	43	41	100.0	100
過去隨機法	4	32	36	38	41	48	11	100.0	100
完全隨機法	6	7	11	13	23	28	21	100.0	100

Show 10 per page

(圖七) 模擬大樂透抽號預測系統結果呈現圖 (另存 csv 新檔)

程式碼 文字 已儲存所有變更

筆記本 大樂透歷史記錄.csv X guojing0805.csv X

1 to 100 of 1888 entries Filter

期別	num1	num2	num3	num4	num5	num6	特別號
1	27	31	13	49	23	21	19
2	19	44	42	48	23	12	33
3	28	39	35	45	26	44	18
4	26	31	33	10	28	16	44
5	28	38	13	33	43	48	4
6	46	45	44	26	6	27	2
7	44	15	7	31	36	6	29
8	36	34	8	26	18	13	17
9	45	6	26	18	39	42	14
10	48	10	12	7	8	47	38
11	16	26	32	48	12	41	1
12	45	7	28	41	48	30	44
13	2	46	25	39	8	43	13
14	34	48	44	19	27	18	37
15	1	19	45	38	33	13	36
16	13	39	48	25	5	30	36
17	49	39	6	36	48	22	28
18	15	39	49	32	12	24	6
19	33	2	15	46	16	31	40
20	21	14	23	31	20	12	45
21	41	4	22	13	24	26	28
22	35	43	28	49	36	38	47
23	47	37	14	15	25	49	13
24	19	32	17	20	5	6	18
25	39	11	21	42	43	4	10
26	14	45	46	40	33	31	49
27	39	49	45	4	10	43	33
28	18	22	37	34	35	10	41
29	20	6	42	2	47	4	39

(圖八) 模擬大樂透抽號預測系統結果呈現圖 (原 csv 檔)

<程式解釋>

這程式主要是利用機率與統計以及隨機分析之方式來歷史中獎號碼並嘗試預測大樂透的中獎號碼。

(1) 首先，程式導入必要的模組，包括 random 用於隨機數生成、pandas 用於資料操作、Counter 用於計算頻率。

(2) 從指定的 CSV 文件中讀取歷史大樂透中獎號碼。其中包含六

個常規號碼和一個特別號碼。讀取後將它們存儲在 `draws` 和 `special_numbers` 中。

(3) 將這些號碼轉換為 `pandas` 的 `DataFrame`，方便後續操作。

(4) 使用 `Counter` 對所有號碼以及特別號碼進行計數，以獲取它們的出現頻率。

(5) 採用兩種策略來預測下一期的大樂透號碼：

第一種策略是選擇歷史上出現頻率最高的六個號碼作為預測。

第二種策略是從未出現在前六個最常見號碼中的號碼中隨機選取六個作為預測。另外再將這兩種策略分別存儲在 `most_common_numbers` 和 `random_choice` 中。

(6) 完全隨機生成一組預測號碼和一個特別號碼，這是第三種預測策略。

(7) 定義了兩個函式 `compare_to_actual` 和 `calculate_accuracy` 來計算預測準確率。前者比較預測號碼和實際號碼，後者計算準確率。

(8) 使用歷史中獎號碼計算三種預測策略的準確率。

創建一個 `DataFrame`，顯示三種預測方法的結果以及它們的準確

率。

(9) 將歷史記錄的中獎號碼和預測結果以表格形式打印出來。

最後，將預測結果存儲到一個新的 CSV 文件中，以便稍後進一步分析。



<法二>：KMeans 演算法

<主要程式>

```
import pandas as pd
from sklearn.cluster import KMeans
import numpy as np

# 讀取CSV文件
file_path = '/content/drive/MyDrive/大樂透/大樂透歷史記錄 .csv' # 確保檔案路徑正確
afile = pd.read_csv(file_path)
file_path2 = '/content/drive/MyDrive/大樂透/guojing0805.csv' # 新的 CSV 文件路徑

# 提取中獎號碼
draws = []
for i in range(len(afile)):
    numbers = [afile.iloc[i]['num1'], afile.iloc[i]['num2'], afile.iloc[i]['num3'],
               afile.iloc[i]['num4'], afile.iloc[i]['num5'], afile.iloc[i]['num6']]
    draws.append(numbers)

# 計算每個號碼的出現次數
all_numbers = [num for draw in draws for num in draw]
number_frequency = pd.Series(all_numbers).value_counts().sort_index()

# 轉換成DataFrame
df = pd.DataFrame({"Number": number_frequency.index, "Frequency": number_frequency.values})

# 使用K-means聚類算法將號碼分為兩組
kmeans = KMeans(n_clusters=2, n_init=10, random_state=42) # 修改默認參數
kmeans.fit(df[['Frequency']])
df['Cluster'] = kmeans.labels_

# 分別選擇兩組號碼中的一組作為預測
cluster_0_numbers = df[df['Cluster'] == 0]['Number'].tolist()
cluster_1_numbers = df[df['Cluster'] == 1]['Number'].tolist()

# 隨機選擇一組號碼作為預測
prediction_numbers = np.random.choice(cluster_0_numbers + cluster_1_numbers, size=6, replace=False)
```

(圖九) 模擬大樂透抽號預測系統程式示意圖 (4)

```
# 分析特別號碼
special_numbers = afile['特別號'].tolist()
special_frequency = pd.Series(special_numbers).value_counts().sort_index()
df_special = pd.DataFrame({"Special Number": special_frequency.index, "Frequency": special_frequency.values})

# 使用K-means聚類算法將特別號碼分為兩組
kmeans_special = KMeans(n_clusters=2, n_init=10, random_state=42) # 修改默認參數
kmeans_special.fit(df_special[['Frequency']])
df_special['Cluster'] = kmeans_special.labels_

# 分別選擇兩組特別號碼中的一組作為預測
cluster_0_special = df_special[df_special['Cluster'] == 0]['Special Number'].tolist()
cluster_1_special = df_special[df_special['Cluster'] == 1]['Special Number'].tolist()

# 隨機選擇一組特別號碼作為預測
prediction_special = np.random.choice(cluster_0_special + cluster_1_special, size=1, replace=False)

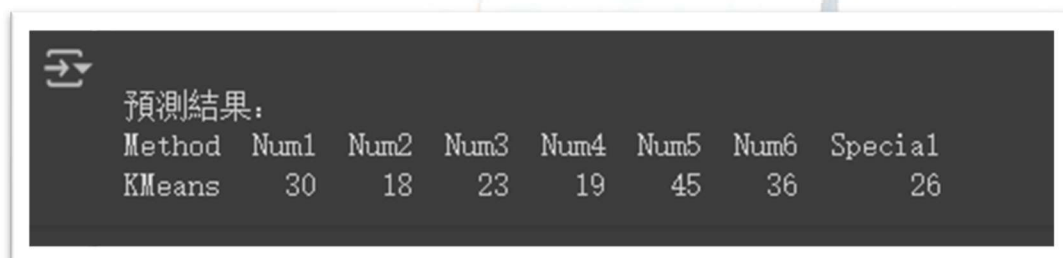
# 建立預測結果的 DataFrame
predicted_df = pd.DataFrame({
    "Method": ["KMeans"], # 預測方法
    "Num1": [prediction_numbers[0]], # 預測的號碼1
    "Num2": [prediction_numbers[1]], # 預測的號碼2
    "Num3": [prediction_numbers[2]], # 預測的號碼3
    "Num4": [prediction_numbers[3]], # 預測的號碼4
    "Num5": [prediction_numbers[4]], # 預測的號碼5
    "Num6": [prediction_numbers[5]], # 預測的號碼6
    "Special": [prediction_special[0]], # 預測的特別號
})

# 列印預測結果表格
print("\n預測結果:")
print(predicted_df.to_string(index=False))

# 將預測結果寫入新的CSV文件
predicted_df.to_csv(file_path2, index=False)
```

(圖十) 模擬大樂透抽號預測系統程式示意圖 (5)

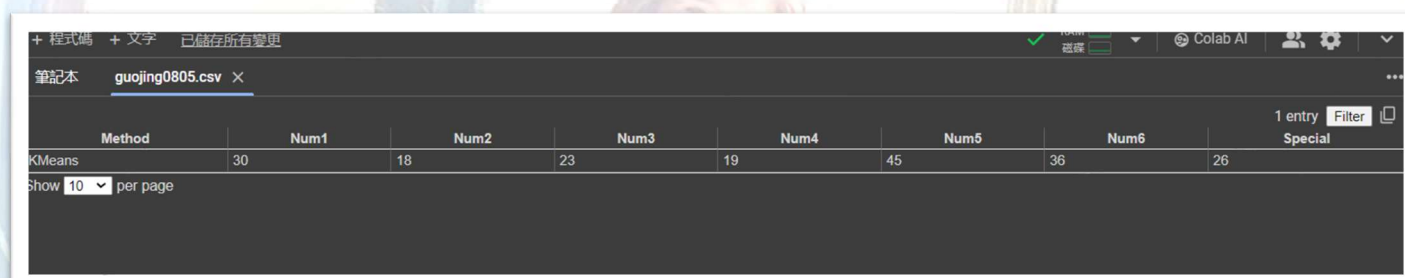
〈呈現結果〉



A terminal window with a dark background and a light border. It contains the text '預測結果:' followed by a table of prediction results. The table has columns for Method, Num1, Num2, Num3, Num4, Num5, Num6, and Special. The row for KMeans shows values 30, 18, 23, 19, 45, 36, and 26.

Method	Num1	Num2	Num3	Num4	Num5	Num6	Special
KMeans	30	18	23	19	45	36	26

(圖十一) 模擬大樂透抽號預測系統結果呈現圖 (3)



A screenshot of a Jupyter Notebook interface. The top bar shows '+ 程式碼 + 文字' and '已儲存所有變更'. The notebook is titled 'guojing0805.csv'. The table content is the same as in Figure 11. The bottom of the interface shows 'Show 10 per page'.

Method	Num1	Num2	Num3	Num4	Num5	Num6	Special
KMeans	30	18	23	19	45	36	26

(圖十二) 模擬大樂透抽號預測系統結果呈現圖 (另存 csv 新檔)

〈程式解釋〉

這段程式碼是使用 K-means 演算法來分析大樂透歷史中獎號碼的出現模式，並以此來預測下一期的抽號結果。

(1) 導入所需的模組，包括 pandas 用於資料操作，KMeans 用於 K-means 用於數據整理及分類（機器學習），以及 numpy 用於數學運算。

(2) 從指定的 CSV 文件中讀取歷史大樂透中獎號碼，並將每一期的號碼提取出來。

(3) 計算每個號碼的出現次數，並將其轉換為 DataFrame。

(4) 使用 K-means 聚類算法將號碼分為兩組。這裡使用的是兩個簇的 K-means 模型。

(5) 從每個簇中分別選擇一組號碼作為預測，或者隨機選擇一組號碼作為預測。

(6) 分析特別號碼的出現模式，並使用 K-means 聚類算法將特別號碼分為兩組。

(7) 從每個簇中分別選擇一個特別號碼作為預測，或者隨機選擇一個特別號碼作為預測。

(8) 創建一個 DataFrame 來存儲預測結果，包括預測的號碼和特別號碼。

(9) 將預測結果打印出來，並將其存儲到新的 CSV 文件中。

總而言之，主要就是使用了 K-means 算法將號碼分為兩組（普通和特別號），然後從每個組中選擇一組號碼作為預測結果進行呈現。

※ K 均值 (K-means) 法

是一種常見的無監督學習算法，主要用於集群分析。它的目標是將一組數據點分為幾個不同的群集，使得每個數據點都屬於與其最近的均值（中心）所代表的群集。

這算法的基本思想是通過迭代地調整群集的中心位置來最小化群內數據點與其所屬群集中心的距離之和（通常是歐式距離的平方）。



〈法三〉：決策樹演算法

〈主要程式〉

```
import pandas as pd
from sklearn.tree import DecisionTreeClassifier
from sklearn.model_selection import train_test_split

# 讀取CSV文件
file_path = '/content/drive/MyDrive/大樂透/大樂透歷史記錄 .csv'
afile = pd.read_csv(file_path)

# 提取中獎號碼和特別號
draws = []
special_numbers = []
# 使用 iloc 從索引 0 開始遍歷
for i in range(len(afile)):
    # 提取每行的中獎號碼和特別號
    numbers = [afile.iloc[i]['num1'], afile.iloc[i]['num2'], afile.iloc[i]['num3'],
               afile.iloc[i]['num4'], afile.iloc[i]['num5'], afile.iloc[i]['num6']]
    special_number = afile.iloc[i]['特別號']
    draws.append(numbers)
    special_numbers.append(special_number)

# 轉換成DataFrame
df = pd.DataFrame(draws, columns=["Num1", "Num2", "Num3", "Num4", "Num5", "Num6"])
df["Special"] = special_numbers

# 打亂數據順序
afile = afile.sample(frac=1, random_state=42).reset_index(drop=True)

# 提取最後一期的中獎號碼
last_draw = afile.iloc[-1]
last_draw_numbers = [last_draw['num1'], last_draw['num2'], last_draw['num3'],
                     last_draw['num4'], last_draw['num5'], last_draw['num6']]
last_draw_numbers_df = pd.DataFrame([last_draw_numbers], columns=['num1', 'num2', 'num3', 'num4', 'num5', 'num6'])

# 將歷史數據中的普通號碼作為特徵數據
X = afile[['num1', 'num2', 'num3', 'num4', 'num5', 'num6']]
# 使用特別號碼作為目標值
y_special = afile['特別號']
```

(圖十三) 模擬大樂透抽號預測系統程式示意圖 (6)

```

# 預測普通號碼：以多個決策樹模型，分別對每一個號碼進行預測
predicted_numbers = []
for i in range(1, 7):
    # 創建一個臨時模型來預測單個號碼
    temp_clf = DecisionTreeClassifier()
    temp_y = afile[f'num{i}']
    temp_clf.fit(X, temp_y)
    predicted_number = temp_clf.predict(last_draw_numbers_df)
    predicted_numbers.append(predicted_number[0])

# 創建決策樹分類器來預測特別號碼
clf_special = DecisionTreeClassifier()
# 訓練特別號碼的模型
clf_special.fit(X, y_special)

# 進行特別號碼預測
predicted_special_number = clf_special.predict(last_draw_numbers_df)

# 建立預測結果的 DataFrame
predicted_df = pd.DataFrame({
    "Method": ["Random Choice"], # 預測方法
    "Num1": [predicted_numbers[0]], # 預測的號碼1
    "Num2": [predicted_numbers[1]], # 預測的號碼2
    "Num3": [predicted_numbers[2]], # 預測的號碼3
    "Num4": [predicted_numbers[3]], # 預測的號碼4
    "Num5": [predicted_numbers[4]], # 預測的號碼5
    "Num6": [predicted_numbers[5]], # 預測的號碼6
    "Special": [predicted_special_number[0]], # 預測的特別號
})

# 列印預測結果表格
print("\n預測結果表格:")
print(predicted_df.to_string(index=False))
# 將預測結果寫入新的 CSV 文件
predicted_df.to_csv(file_path2, index=False)

```

(圖十四) 模擬大樂透抽號預測系統程式示意圖 (7)

〈呈現結果〉

預測結果表格:							
Method	Num1	Num2	Num3	Num4	Num5	Num6	Special
DecisionTree	30	47	2	21	42	12	31

(圖十五) 模擬大樂透抽號預測系統結果呈現圖 (4)

筆記本 guojing0805.csv ×

Method	Num1	Num2	Num3	Num4	Num5	Num6	Special
DecisionTree	30	47	2	21	42	12	31

show 10 per page

1 entry Filter

(圖十六) 模擬大樂透抽號預測系統結果呈現圖 (另存 csv 新檔)

<程式解釋>

這段程式碼是使用決策樹分類器來預測大樂透的中獎號碼。以下是程式碼的解釋：

- (1) 導入必要的模組，包括 pandas 用於資料操作、DecisionTreeClassifier 用於構建決策樹模型、以及 train_test_split 用於數據集的分割。
 - (2) 從指定的 CSV 文件中讀取歷史大樂透中獎號碼，並提取每一期的中獎號碼和特別號。
 - (3) 將數據轉換成 DataFrame 格式，方便後續處理。
- 使用 sample 方法將數據打亂順序，確保模型訓練的隨機性。
- 並提取最後一期的中獎號碼作為測試數據。
- (4) 將歷史數據中的普通號碼作為特徵數據 X，將特別號碼作為目標值 y_special。
 - (5) 使用多個決策樹模型對每一個號碼進行預測，並將預測結果存

儲在 `predicted_numbers` 中。再從中使用單一的決策樹模型來預測特別號碼。並將預測結果存儲在 `DataFrame` 中。

(6) 將預測結果打印出來，並將其寫入新的 CSV 文件中。

總之主要就是使用了監督式學習中的決策樹分類器，以歷史數據來訓練模型，並對最後一期的中獎號碼進行預測。

※決策樹法

是一種常見的監督式學習算法，用於分類和回歸任務。它通過對數據集中的特徵進行分割，建立一個樹狀結構，以解釋和預測目標變量的值。

那其主要的思想就是通過一系列的條件分支來將數據集分割成不同的子集，每個子集中的數據點都具有相似的特徵或目標變量值。這種分割是根據特徵的值來進行的，而每個分割點對應著一個決策規則。

