

資料處理 - 針對網路遊戲「新楓之谷」中自架賭場的結果預測

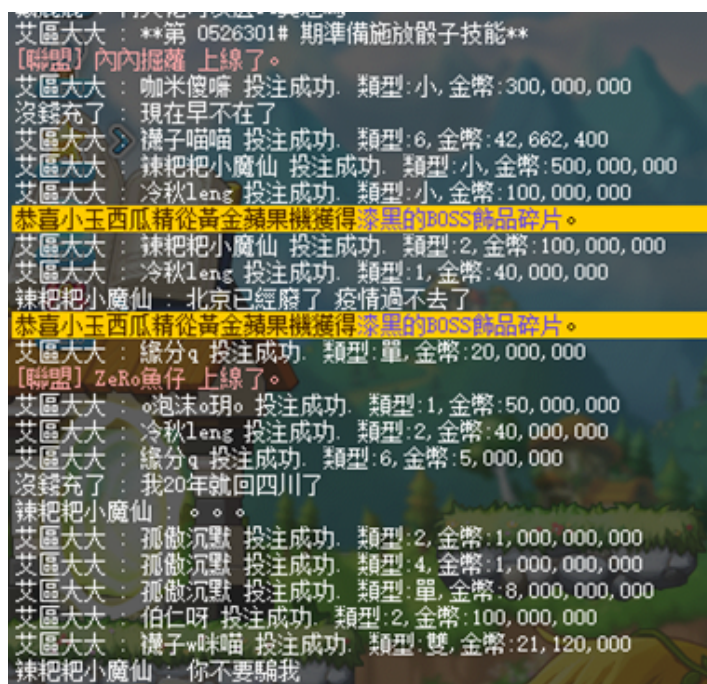
初步構想

在「新楓之谷」這個遊戲當中，其實充斥著賭博的成分，諸如像是遊戲本身的時裝、髮型、臉型、現金道具、道具強化等等都是有機率性的，可以說玩這個遊戲的很多玩家都有一點賭博心態。而正因如此，或許是有人看中這一點，而在這遊戲中開始架設賭場，透過施放技能「幸運骰子」來擲出1~6的數字，讓玩家透過下注單、雙、大、小、數字等方式來進行賭博，正常的骰子看似機率都為六分之一，然而畢竟是遊戲，會有所謂外掛的可能性，因此當歷史數據有點誇張時，像是連續出現十幾次連續開出單號的情況，常常會被玩家詬病賭場「控骰」，也正因如此我才思考或許能夠去分析資料而掌握控骰時機，並且進行一定的預測。

預期成果與效益

在這次的嘗試中，主要是想要去釐清賭場是否真的有可能有控骰的可能性，因為總是有玩家抱怨賭場機制有問題，當有人下注鉅額賭注時，只要被咬就覺得是被控骰，因此我想要去探討控骰的真實性，看看是賭場不公還是人心態上總是會覺得自己被擺了一道。另外如果真的有控骰的情況發生，我希望模型能夠在投注大、小、單、雙這種勝率正常為50%的情況下，能夠有55%以上的勝率，來證實控骰是真實存在的，以此告誡長期去賭場的玩家遠離賭博。

資料收集與取得



艾區大大 : **第 0526301# 期準備施放骰子技能**
[聯盟] 內內掘窿 上線了。
艾區大大 : 咖米傻嘴 投注成功. 類型:小, 金幣:300,000,000
沒錢充了 : 現在早不在了
艾區大大 : 襪子喵喵 投注成功. 類型:6, 金幣:42,662,400
艾區大大 : 辣把把小魔仙 投注成功. 類型:小, 金幣:500,000,000
艾區大大 : 冷秋1eng 投注成功. 類型:小, 金幣:100,000,000
恭喜小王西瓜精從黃金蘋果機獲得遠東的BOSS飾品碎片。
艾區大大 : 辣把把小魔仙 投注成功. 類型:2, 金幣:100,000,000
艾區大大 : 冷秋1eng 投注成功. 類型:1, 金幣:40,000,000
辣把把小魔仙 : 北京已經廢了 疫情過不去了
恭喜小王西瓜精從黃金蘋果機獲得遠東的BOSS飾品碎片。
艾區大大 : 緣分q 投注成功. 類型:單, 金幣:20,000,000
[聯盟] ZeRo魚仔 上線了。
艾區大大 : o泡沫o玥o 投注成功. 類型:1, 金幣:50,000,000
艾區大大 : 冷秋1eng 投注成功. 類型:2, 金幣:40,000,000
艾區大大 : 緣分q 投注成功. 類型:6, 金幣:5,000,000
沒錢充了 : 我20年就回四川了
辣把把小魔仙 : 。。。
艾區大大 : 孤傲沉默 投注成功. 類型:2, 金幣:1,000,000,000
艾區大大 : 孤傲沉默 投注成功. 類型:4, 金幣:1,000,000,000
艾區大大 : 孤傲沉默 投注成功. 類型:單, 金幣:8,000,000,000
艾區大大 : 伯仁呀 投注成功. 類型:2, 金幣:100,000,000
艾區大大 : 襪子*喵喵 投注成功. 類型:雙, 金幣:21,120,000
辣把把小魔仙 : 你不要騙我

如圖，以上為其中一期的投注信息，「艾區大大」這個遊戲角色是遊戲內眾多賭場之一，透過腳本自動與玩家進行互動並抓取遊戲內訊息，每個玩家先在遊戲中密語艾區大大來儲值賭金(賭金會放入獎池中)：

儲值、領取獎池指令：

艾區大大：領幣請密語：a 金額 儲值請密語：ch

投注指令：

艾區大大：密語下注格式：下注類型（空格）下注金額 骰手下注上
限為單局1000e

其他指令：

艾區大大：G=轉移獎池，A=指定獎池領取，QA=類型=取消投注，bi=隱藏
投注信息

賠率：

艾區大大：**賠率：大雙-2.02，小單-1.9，數字1到5-5.2 6-6 **



透過下注單、雙、大、小、數字及金額來進行投注(ex:雙 1e、大 1e、6 1e)，並且每隔180秒會進行一次擲骰，因此資料的收集方式就是透過觀察每次擲骰區間的投注情況，來觀察是否有控骰的可能性並且去預測下一次的擲骰情況。目前想到能夠納入考慮的資料有以下：

- 單場參與人數
- 歷史數據
- 單場押單次數、金額
- 單場押雙次數、金額
- 單場押大次數、金額
- 單場押小次數、金額
- 單場押1~6次數、金額

- 單場單筆最大投注金額
- 目前連續出現大、小、單、雙的次數

初步想法

從以上其實不難想到，因為獎池為所有玩家共同投入，並且賭場玩家間會有所謂「賣獎池」這個行為，也就是上面指令有一個G的指令，可以將自己的獎池轉移給其他玩家，而賭場也會將獎池內的錢賣給玩家，也就是說其實賭場幾乎沒有運營成本，只要一直有玩家，獎池內就不會沒錢，有點類似龐式騙局，用玩家的錢補給玩家，因此雖然以期望值來看賭場在長崎下本來就會贏，但在這種手法之下賭場更是可以大撈一筆，個人認為賭場在這樣的條件下似乎沒有控股的必要。

資料蒐集的困難性及探討

在上面我列出了許多用以判斷是否有控股可能的資料，像是參與人數、單場單筆最大投注金額等等。然而我發現了一個困難之處，也就是上面「其他指令」的地方裡有個指令是「bi」，這個指令可以讓玩家隱藏投注信息，也就是玩家的投注情況不會公布在聊天室中，這也讓資料的蒐集有困難，甚至讓這些數據有可能去誤導模型判斷其為控股或不控股的情況，因此無奈之下必須刪除以下資料：

- 單場參與人數
- 單場押單次數、金額
- 單場押雙次數、金額
- 單場押大次數、金額
- 單場押小次數、金額
- 單場押1~6次數、金額
- 單場單筆最大投注金額

也因此剩下能用的資料只有：

- 歷史數據
- 目前連續出現大、小、單、雙的次數

思考到這邊頓時覺得賭場真的很聰明，透過這種方式讓玩家沒辦法對其進行準確的預測，但這反而又讓我覺得賭場真的有可能有控股的情況，只不過要單以上面這兩種資料進行預測，所作出的模型預測成果可想而知會非常有限，不過既然決定要做了那就還是嘗試看看，稍微思考過後還能加入以下資料：

- 大、小、單、雙、各數字在歷史數據(最近15次)中的出現次數
- 最近一期開獎結果

資料處理

首先，我們唯一能得到的就是歷史數據，因此我會把歷史數據整理成maplestory.csv，然後建立成DataFrame，如下圖：

nber	history1	history2	history3	history4	history5	history6	history7	history8	history9	history10	history11	history12	history13	history14	history15	result
1	3	2	6	2	5	4	4	4	6	5	5	6	6	2	6	4
2	3	4	6	2	4	1	5	1	6	5	4	4	6	5	3	5
3	4	6	2	4	1	5	1	6	5	4	4	6	5	3	5	3
4	6	2	4	1	5	1	6	5	4	4	6	5	3	5	3	1
5	2	4	1	5	1	6	5	4	4	6	5	3	5	3	1	1
...
71	1	3	6	5	4	4	6	6	4	2	5	1	2	3	2	3
72	3	6	5	4	4	6	6	4	2	5	1	2	3	2	3	3
73	6	5	4	4	6	6	4	2	5	1	2	3	2	3	3	6
74	5	4	4	6	6	4	2	5	1	2	3	2	3	3	6	4
75	4	4	6	6	4	2	5	1	2	3	2	3	3	6	4	2

再來就開始處理資料，首先先把每筆數據出現大、小、單、雙的次數都整理出來，並且加入資料集當中，用以下方法把history1~15都統計一次，就會得到下圖：

```
for num in data['history2'] :
    if(num == 1) :
        small_num += 1
        single_num += 1
    if(num == 2) :
        small_num += 1
        pair_num += 1
    if(num == 3) :
        small_num += 1
        single_num += 1
    if(num == 4) :
        big_num += 1
        pair_num += 1
    if(num == 5) :
        big_num += 1
        single_num += 1
    if(num == 6) :
        big_num += 1
        pair_num += 1

    big.at[i, 'big'] += big_num
    small.at[i, 'small'] += small_num
    single.at[i, 'single'] += single_num
    pair.at[i, 'pair'] += pair_num

    i += 1
    big_num = 0
    small_num = 0
    single_num = 0
    pair_num = 0
i = 0
```

```
data = pd.concat([data, big, small, single, pair], axis=1)
data
```

story2	history3	history4	history5	history6	history7	history8	history9	...	history11	history12	history13	history14	history15	result	big	small	single	pair
2	6	2	5	4	4	4	6	...	5	6	6	2	6	4	11	4	4	11
4	6	2	4	1	5	1	6	...	4	4	6	5	3	5	10	5	7	8
6	2	4	1	5	1	6	5	...	4	6	5	3	5	3	11	4	7	8
2	4	1	5	1	6	5	4	...	6	5	3	5	3	1	10	5	8	7
4	1	5	1	6	5	4	4	...	5	3	5	3	1	1	9	6	9	6
...
3	6	5	4	4	6	6	4	...	5	1	2	3	2	3	8	7	6	9
6	5	4	4	6	6	4	2	...	1	2	3	2	3	3	8	7	6	9
5	4	4	6	6	4	2	5	...	2	3	2	3	3	6	8	7	6	9
4	4	6	6	4	2	5	1	...	3	2	3	3	6	4	8	7	6	9
4	6	6	4	2	5	1	2	...	2	3	3	6	4	2	8	7	5	10

再來就是考慮連續出現的情況，判斷的方法為，先去看歷史數據最後一個數字，看是大小單雙哪一種，假如是3那就是小單，再往前看一個數字，假如是4，這樣就沒有連續，那如果是2的話，那就又是小，所以再往前看一個數字，假如是3，則目前最後三個數字為323，這樣連續出現3個我就視為他正在連續並把他設成3，那如果再往前看是1的話就會是1323，我就會把他+1變成4，以此來表示連續的程度：

story4	history5	history6	history7	history8	history9	...	history15	big_continue	small_continue	single_continue	pair_continue	result	big	small	single	pair
2	5	4	4	4	6	...	6	0	0	0	4	4	11	4	4	11
2	4	1	5	1	6	...	3	0	0	0	0	5	10	5	7	8
4	1	5	1	6	5	...	5	0	0	3	0	3	11	4	7	8
1	5	1	6	5	4	...	3	0	0	4	0	1	10	5	8	7
5	1	6	5	4	4	...	1	0	0	5	0	1	9	6	9	6
...
5	4	4	6	6	4	...	2	0	4	0	0	3	8	7	6	9
4	4	6	6	4	2	...	3	0	5	0	0	3	8	7	6	9
4	6	6	4	2	5	...	3	0	6	0	0	6	8	7	6	9
6	6	4	2	5	1	...	6	0	0	0	0	4	8	7	6	9
6	4	2	5	1	2	...	4	0	0	0	0	2	8	7	5	10

模型的選用

資料處理完後，在挑選模型時考慮到我自己的構想就是像例如現在正在連哪個數字、正在面臨怎樣的情況來進行預測，因此感覺可以採用Decision tree作為模型，來區分應該會是哪個數字。

另外，由於在做預測時要命中單一數字感覺實在很困難，因此我決定讓模型判斷完數字後，以大小來區分，1~3我會output 0、4~6我會output 1，如果是0我就會猜是小、是1就會猜是大，以這樣的區分方式來進行驗證：

```
from sklearn import tree
from sklearn.model_selection import train_test_split

X = data.drop('result', axis=1)
y = data['result']

train_X, test_X, train_y, test_y = train_test_split(X, y, test_size = 0.3)

model = tree.DecisionTreeClassifier()
model = model.fit(train_X, train_y)
prediction = model.predict(test_X)
```

```
i = 0
predict = np.zeros(shape = len(prediction)).astype(int)
for num in prediction :
    if(num <= 3) :
        predict[i] = 0
    elif(num >= 4) :
        predict[i] = 1
    i+=1

i = 0
test_Y = np.zeros(shape = len(test_y)).astype(int)
for num in test_y :
    if(num <= 3) :
        test_Y[i] = 0
    elif(num >= 4) :
        test_Y[i] = 1
    i+=1
```

成果&結論

```
In [6]: i = 0
correct = 0

for num in predict :
    if(num == test_Y[i]) :
        correct += 1
    i+=1

accuracy = correct / i
accuracy
```

Out[6]: 0.5217391304347826

```
In [7]: accuracy
```

Out[7]: 0.5217391304347826

由上圖可看出準確率大概落在50%上下，有時甚至能達到70%：

```
In [7]: accuracy
```

Out[7]: 0.7391304347826086

當中是有考慮到控骰可能性的，但因為資料不多，所以準確率也不高，但至少能確保大概有50%的勝率，只能說賭場的機制真的做得很聰明，透過讓玩家可以隱藏訊息來讓玩家難以猜測，表面上看似是保護玩家的隱私，實際上是讓玩家沒有破解的可能，因此老話一句，遠離賭場才是明智之舉！