

# 期末報告

1. 組員：409170517 韋重宇、409170581 劉睿麒、409170567 楊勝傑。
2. 目標：賺大錢
3. 商品：老爺知 5704.TWO
4. 功能：可以查找特定股票的買賣訊號以及技術指標。例如：MACD、KD、三烏鴉訊號、吞噬訊號。
5. 程式說明：

```
import yfinance as yf
import pandas as pd
...

# 分析0050
df = yf.download('5704.TWO', period='120d', interval='1d')
df
```

1. 匯入必要的函式庫
2. 下載股票資料：

使用 yfinance 函式庫的 download 函數下載了股票代號為 '5704.TWO' 的台灣證券交易所上市公司的資料，設定了下載的時間範圍為最近 120 天，K 棒設置為日 K。最後輸出資料。

```
import mplfinance as mpf

#上漲為綠色，下跌為紅色
#5日橘色，10日黃色，20日紫色
mpf.plot(df, type='candle', style='yahoo', volume=True, figratio=(20, 10))
```

1. 匯入函式庫

2. 繪製蠟燭圖：

使用 mpf.plot 函式繪製蠟燭圖。其中參數含義：

- type='candle'：指定繪製蠟燭圖。
- style='yahoo'：指定使用 Yahoo Finance 風格的圖表樣式。
- volume=True：顯示交易量的圖表。
- figratio=(20, 10)：設定圖表的寬高比為 20:10。

```

from talib.abstract import MACD

# OSI>0 買進
K_Data=df
K_Data.columns=[ i.lower() for i in K_Data.columns ]
macdData=MACD(K_Data,fastperiod=12,slowperiod=26,signalperiod=9)
K_Data = K_Data.join(macdData)
adplot = []
adplot.append(mpf.make_addplot(K_Data['macd'], panel=2, type='line', marker='.', color='red', \
scatter=False, ylabel='DIF & MACD', secondary_y=False))
adplot.append(mpf.make_addplot(K_Data['macdsignal'], panel=2, type='line', marker='.', color='green', \
scatter=False, ylabel='DIF & MACD', secondary_y=False))
adplot.append(mpf.make_addplot(K_Data['macdhist'], panel=2, type='bar', marker='.', color='blue', \
scatter=False, ylabel='OSC', secondary_y=False))
K_Data_color = mpf.make_marketcolors(up='green', down='red', inherit=True)
K_Data_style = mpf.make_mpf_style(base_mpf_style='yahoo', edgecolor='black', marketcolors=K_Data_color)
mpf.plot(K_Data, type='candle', style=K_Data_style, volume=True, addplot=adplot)

```

1. 匯入函式庫
2. 使用 MACD 函數計算 K\_Data 的 MACD 指標，並指定快線週期為 12，慢線週期為 26，訊號線週期為 9。
3. 使用 make\_addplot 函數來創建額外的指標圖表。指標包括 MACD 線（DIF）、訊號線（MACD Signal）和 MACD 柱（MACD Histogram）。
4. 使用 make\_mpf\_style 函數創建了一個圖表樣式配置，基於 Yahoo Finance 的風格，設定了邊線顏色為黑色，並應用之前創建的顏色配置。
5. 使用 plot 函數繪製蠟燭圖，並將之前創建的圖表樣式 K\_Data\_style 應用於圖表。同時，設定顯示交易量的圖表，並添加了之前創建的指標圖表 adplot。最終呈現的圖表將包含蠟燭圖、交易量、MACD 線、訊號線和 MACD 柱。

同樣使用了 talib 函式庫，但這次是用來計算和繪製股票的 KD 指標（隨機指標）

```
from talib.abstract import STOCH
import yfinance as yf
import mplfinance as mpf

# K>D 買進
stochData=STOCH(K_Data,fastk_period=9,slowk_period=3,slowd_period=3)
K_Data = K_Data.join(stochData)
adplot = []
adplot.append(mpf.make_addplot(K_Data['slowk'], panel=2, type='line', marker='.', \
color='red', scatter=False, ylabel='KD', secondary_y=False))
adplot.append(mpf.make_addplot(K_Data['slowd'], panel=2, type='line', marker='.', \
color='blue', scatter=False, ylabel='KD', secondary_y=False))
K_Data_color = mpf.make_marketcolors(up='red', down='green', inherit=True)
K_Data_style = mpf.make_mpf_style(base_mpf_style='yahoo', edgecolor='black', marketcolors=K_Data_color)
mpf.plot(K_Data, type='candle', style=K_Data_style, volume=True, addplot=adplot)
```

1. 匯入函式庫
2. 使用 STOCH 函數計算 K\_Data 的 KD 指標，並指定快速隨機指標的週期為 9，慢速隨機指標的週期為 3。接著將 K\_Data 和 stochData 進行合併。
3. 使用 make\_addplot 函數創建了額外的指標圖表。其中包括 KD 指標的快速隨機指標（%K）和慢速隨機指標（%D）。
4. 使用 plot 函數繪製蠟燭圖，並將之前創建的圖表樣式 K\_Data\_style 應用於圖表。同時，設定顯示交易量的圖表，並添加了之前創建的指標圖表 adplot。最終呈現的圖表將包含蠟燭圖、交易量、KD 指標的 %K 線和 %D 線。

使用 talib 函式庫的 CDL3BLACKCROWS 函數，該函數用於識別股票中的三烏鴉（Three Black Crows）型態。

```
from talib.abstract import CDL3BLACKCROWS
import yfinance as yf

Pattern = CDL3BLACKCROWS(K_Data)
Result=Pattern[Pattern!=0]
if len(Result)!=0:
    print('這個期間共有 {} 筆三烏鴉型態訊號:'.format(len(Result)))
    for i in range(len(Result)):
        print(Result.index[i],Result[i])
else:
    print('這個期間沒有三烏鴉型態訊號')
```

1. 匯入函式庫
2. 使用 CDL3BLACKCROWS 函數來分析 K\_Data，並識別其中的三烏鴉型態。
3. 過濾掉 Pattern 中值為零的項目，只保留非零值的三烏鴉型態。
4. 判斷是否有識別到三烏鴉型態。如果 Result 的長度不為零，即存在三烏鴉型態，則輸出相應的訊息。首先，輸出有多少筆三烏鴉型態訊號。使用迴圈遍歷 Result，並輸出每個型態訊號的日期和該型態的數值。如果 Result 的長度為零，則輸出沒有三烏鴉型態訊號的訊息。

使用 talib 函式庫的 CDLENGULFING 函數，用於識別股票中的吞噬型態（Engulfing Pattern）

```
from talib.abstract import CDLENGULFING
import yfinance as yf

Pattern = CDLENGULFING(K_Data)
Result=Pattern[Pattern!=0]
if len(Result)!=0:
    print('這個期間共有 {} 筆吞噬型態訊號:'.format(len(Result)))
    for i in range(len(Result)):
        print(Result.index[i],Result[i])
else:
    print('這個期間沒有吞噬型態訊號')
```

1. 匯入函式庫
2. 使用 CDLENGULFING 函數來分析 K\_Data，並識別其中的吞噬型態。
3. 過濾掉 Pattern 中值為零的項目，只保留非零值的吞噬型態。
4. 判斷是否有識別到吞噬型態。如果 Result 的長度不為零，即存在吞噬型態，則輸出相應的訊息。首先，輸出有多少筆吞噬型態訊號。使用迴圈遍歷 Result，並輸出每個型態訊號的日期和該型態的數值。如果 Result 的長度為零，則輸出沒有吞噬型態訊號的訊息。