

財經資料分析工具開發 期末報告書

目錄

HW-1 配對交易策略/Hightcharts 工具

- 1.1 作業目標
- 1.2 實作中學習到的知識與遭遇的困難
- 1.3 結果截圖
- 1.4 程式使用方式

HW-2 建構簡單進出場策略+ backtrader 實作

- 2.1 作業目標
- 2.2 實作中學習到的知識與遭遇的困難
- 2.3 結果截圖
- 2.4 程式使用方式

HW-3 Django Rest Framework API

- 3.1 作業目標
- 3.2 實作中學習到的知識與遭遇的困難
- 3.3 結果截圖
- 3.4 程式使用方式

HW-4 Secured Function API/DATA API

4.1 作業目標

4.2 實作中學習到的知識與遭遇的困難

4.3 結果截圖

4.4 程式使用方式

HW-5 Tracker

5.1 作業目標

5.2 實作中學習到的知識與遭遇的困難

5.3 結果截圖

5.4 程式使用方式

HW-6 Docker

6.1 作業目標

6.2 實作中學習到的知識與遭遇的困難

6.3 結果截圖

6.4 程式使用方式

HW-7 選股 fin 模組

7.1 作業目標

7.2 實作中學習到的知識與遭遇的困難

7.3 結果截圖

7.4 程式使用方式

HW-8 選股 fin 網站

8.1 作業目標

8.2 實作中學習到的知識與遭遇的困難

8.3 結果截圖

8.4 程式使用方式

HW-9 股票定價策略

9.1 作業目標

9.2 實作中學習到的知識與遭遇的困難

9.3 結果截圖

9.4 程式使用方式

HW-10 本益比河流圖

10.1 作業目標

10.2 實作中學習到的知識與遭遇的困難

10.3 結果截圖

10.4 程式使用方式

HW-11 天花板地板線

11.1 作業目標

11.2 實作中學習到的知識與遭遇的困難

11.3 結果截圖

11.4 程式使用方式

HW-12 KD 指標、MACD 指標、布林通道

12.1 作業目標

12.2 實作中學習到的知識與遭遇的困難

12.3 結果截圖

12.4 程式使用方式

HW-13 RSI 指標、ADX 指標、DMI 指標

13.1 作業目標

13.2 實作中學習到的知識與遭遇的困難

13.3 結果截圖

13.4 程式使用方式

HW-14 FinalProject、K 線型態

14.1 作業目標

14.2 實作中學習到的知識與遭遇的困難

14.3 結果截圖

14.4 程式使用方式

修課心得

GitHub: https://github.com/Kane-ouvic/Eco_Web

HW-1 配對交易策略/Hightcharts 工具

1.1 作業目標

配對交易策略實作 (Distance Method)

- 用戶輸入股票代號、起始日、結束日、參數。
- 使用 Yahoo Finance API 抓取股價數據 (開高低收量)。
- 計算 Spread (兩檔股票對數價格差異)、移動平均與標準差 (window size = 200)。
- 根據上下線進行開倉與關倉操作 (上線、下線以 1 至 3 倍標準差為門檻)。
- 計算損益。

結果視覺化

- Highchart Stock：繪製股價與進出場訊號。
- Datatable：顯示進出場訊號與損益資訊。

網站整合

- 前端顯示圖形與表格。
- 後端整合 Yahoo Finance API 與策略計算。

1.2 實作中學習到的知識與遭遇的困難

雖然之前有碰過網頁前後端的經驗，但其實我之前沒有使用過 Highcharts，所以花了蠻多時間去查資料，看 highcharts 這個 javascript 的套件可以做出哪些圖表，這次的作業讓我對 highcharts 有基本的瞭解，以及如何畫出專業分析

股市的圖表。

1.3 結果截圖

HomeHW1HW2_1HW2_2HW8HW9HW10HW11HW12HW13HW14_1HW14_2K-LineWelcome, n26130655!MonitorLogout

Strategy Parameters

Stock 1

AAPL

Stock 2

GLD

Start Date

2021/01/01

End Date

2024/01/01

n * std

2

Window Size

200

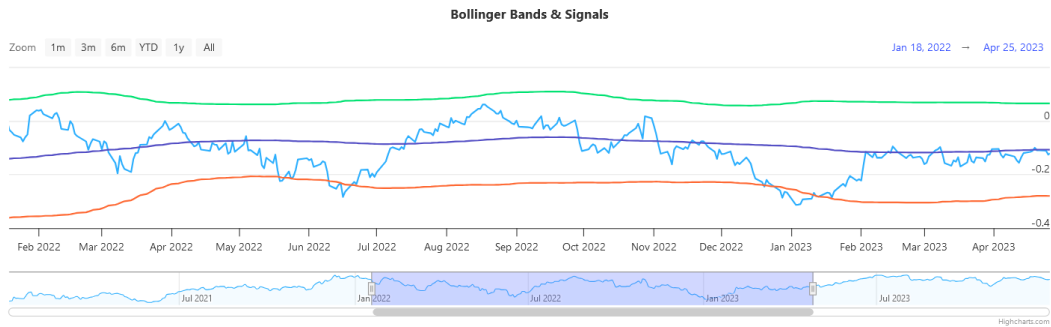
Submit

Add Track

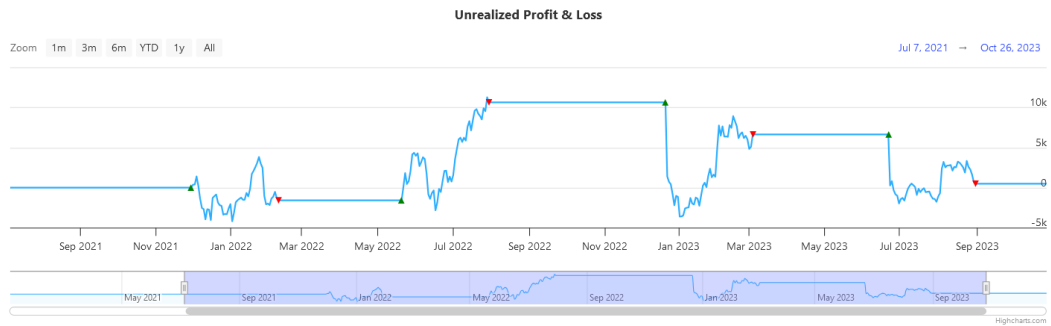
Strategy Results



Bollinger Bands & Signals



Profits & Loss



Details

Show entries

Search:

Date	Type	Action of AAPL	Price of AAPL	Action of GLD	Price of GLD
2021-11-30	OPEN	SELL	165.3000030517578	BUY	165.5
2022-02-09	CLOSE	SELL	176.27999877929688	BUY	171.2100067138672
2022-05-19	OPEN	BUY	137.35000610351562	SELL	171.91000366210938
2022-08-01	CLOSE	SELL	161.50999450683594	BUY	165.02999877929688
2022-12-20	OPEN	BUY	132.3000030517578	SELL	169.0800018310547

Showing 1 to 5 of 8 entries

Previous 2 Next

1.4 程式使用方式

Step1. 前端網頁輸入兩檔股票、初始日期、結束日期、標準差、窗口大小

Step2. 後端計算前端發送 Post 請求並回傳結果

Step3. 前端頁面渲染圖表以及計算結果

HW-2 建構簡單進出場策略+ backtrader 實作

2.1 作業目標

簡單進出場策略建構

- 建立簡單的技術分析交易策略（例如：RSI 黃金交叉策略、均線策略）。
- 可設定參數（如 RSI 短周期與長周期）。

回測實作（Backtrader）

- 使用 backtrader 套件回測進出場策略。
- 記錄進出場紀錄，包含：股票代碼、買賣別

、買進日期、價格、賣出日期、價格、買賣單位

結果視覺化

- Datatable：顯示進出場紀錄表格。

- Highchart：繪製股價走勢圖與進出場點位。
- 報酬率 & MDD 走勢圖：展示風險與報酬。
- 風險報酬分析：呈現交易績效指標（如總績效、勝率、賺賠比、最大回撤 MDD）。

功能整合與展示

- 輸入：股票代碼、參數、開始與結束日期。
- 輸出：圖表與表格展示於網站（用 Highchart 與 Datatable）。

2.2 實作中學習到的知識與遭遇的困難

這次的作業主要是讓我們找一個書上的分析方法實作，另外還有怎麼使用 backtrader 這個 python 套件來做回測分析，主要遇到的問題是在計算各個結果指標花了蠻多時間理解並把它實作出來，像是進出場點以及回報率等等。

2.3 結果截圖

HW2_1

Home HW1 HW2_1 HW2_2 HW8 HW9 HW10 HW11 HW12 HW13 HW14_1 HW14_2 K-Line Welcome, n26130655! Monitor Logout

RSI Crossover Strategy Backtest

Strategy Parameters

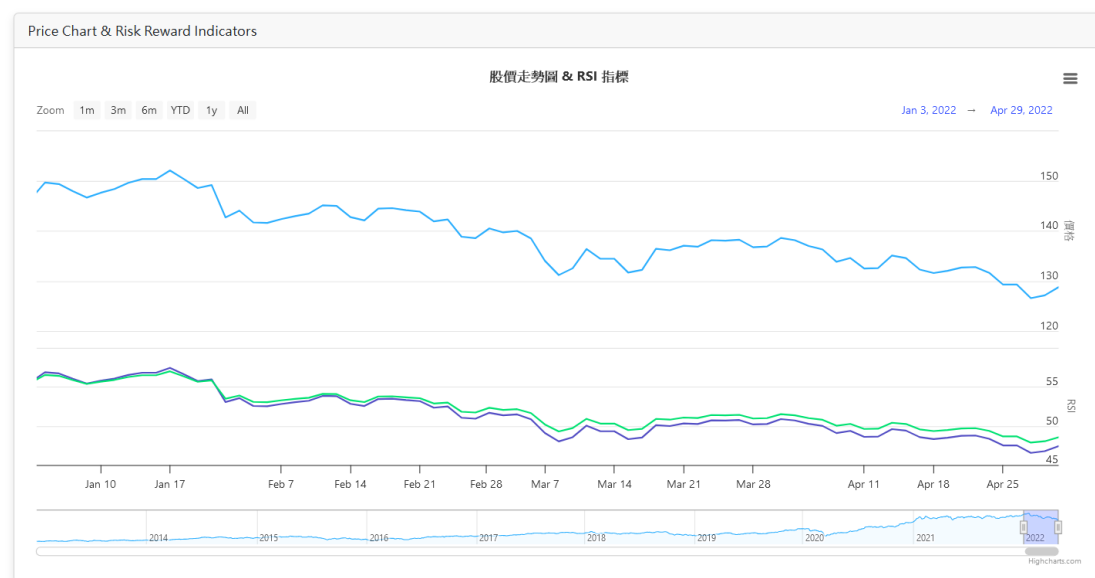
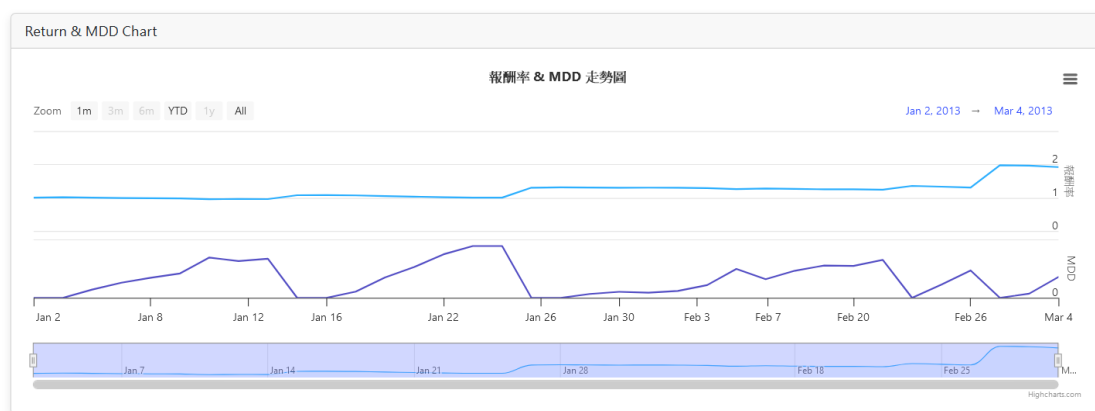
Stock Symbol	Start Date	End Date
0050.TW	2013/01/01	2022/05/01
Short RSI Period	Long RSI Period	Exit Threshold
120	150	0.999

Execute Backtest

Trade Record				
Show 10 entries		Search: <input type="text"/>		
進場時間	進場價格	出場時間	出場價格	報酬率
2013-09-04	56.5	2013-10-28	57.150001525878906	1.15%
2013-10-29	57.45000076293945	2013-11-06	56.79999923706055	-1.13%
2013-12-02	57.599998474121094	2013-12-06	57.04999923706055	-0.95%
2013-12-10	57.54999923706055	2013-12-17	57.150001525878906	-0.70%
2013-12-24	57.849998474121094	2014-01-10	57.5	-0.61%
2014-01-14	57.70000076293945	2014-02-05	56.400001525878906	-2.25%
2014-02-19	58.099998474121094	2014-02-21	58.400001525878906	0.52%
2014-02-24	58.400001525878906	2014-03-05	58.20000076293945	-0.34%
2014-03-06	58.349998474121094	2014-09-26	65.5	12.25%
2015-01-26	68.55000305175781	2015-03-30	68.75	0.29%

Showing 1 to 10 of 35 entries

Previous 1 2 3 4 Next



Total Return	91.91%	Average Return	2.29%
Win Rate	31.43%	Average Profit	9.74%
Average Loss	-1.17%	Profit-Loss Ratio	8.33
Expectancy	2.26%	Profitable Average Holding Days	971
Loss Average Holding Days	208	Max Consecutive Loss	7.3%
Max Consecutive Profit	51.54%		

HW2_2

HomeHW1HW2_1HW2_2HW8HW9HW10HW11HW12HW13HW14_1HW14_2K-Line

Welcome: n26130655!MonitorLogout

RSI Backtrader

Strategy Parameters

股票代號:
2330.TW

起始日期:
2020/01/01

結束日期:
2025/01/01

短期 RSI 週期:
14

長期 RSI 週期:
30

超買區:
70

超賣區:
30

單筆下單股數:
1000

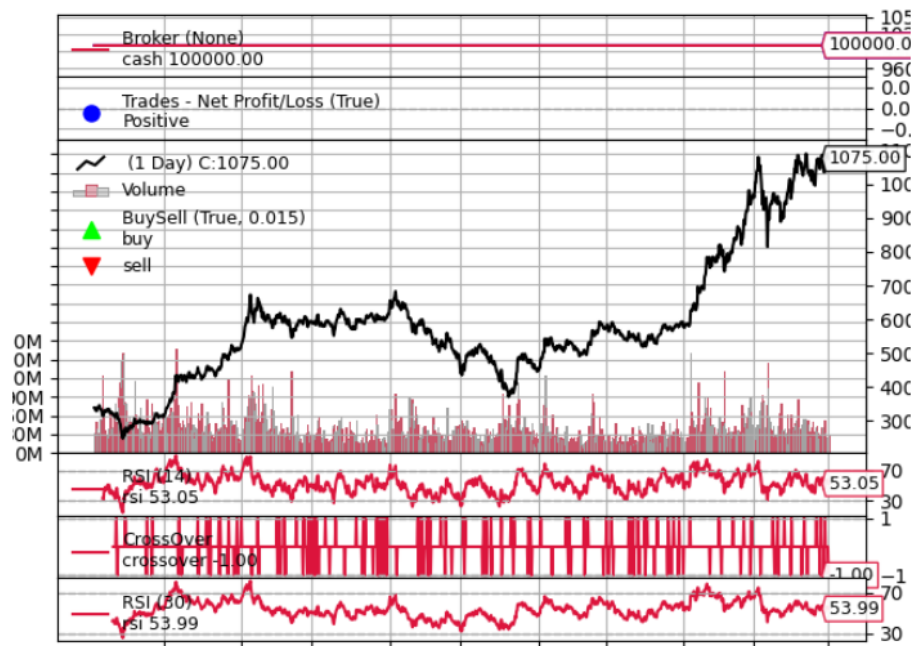
初始資金:
100000

手續費比率:
0.001425

執行回測

0.001425

執行回測



2.4 程式使用方式

HW2_1

Step1. 前端網頁輸入股票代碼、初始日期、結束日期、短週期 RSI、長週期 RSI、Exit Threshold

Step2. 後端計算前端發送 Post 請求並回傳結果

Step3. 前端頁面渲染圖表以及計算結果

HW2_2

Step1. 前端網頁輸入股票代碼、初始日期、結束日期、短週期 RSI、長週期 RSI、超買區、超賣區、單筆下單股數、初始資金、手續費比率

Step2. 後端計算前端發送 Post 請求並回傳結果

Step3. 前端頁面回傳 backtrader 產生的圖片

HW-3 Django Rest Framework API

3.1 作業目標

建構 Django Rest Framework (DRF) API

- 使用 DRF 建立 Web API，簡化後端與前端資料交換。

API 設計與開發

- 視圖 (Views)：使用 ModelViewSet 建立 API 行為（如 BacktestViewSet 與 TrackViewSet）。
- 路由 (Router)：將 API 註冊到 urls.py，設定可呼叫的路徑。

API 測試與整合

- 使用 Postman 測試 API 的正確性（狀態碼 200 成功、404 失敗）。

- 使用 Function API 整合回測與追蹤功能。
- 結合 網站前端 (Highchart、Datatable) 展示結果。

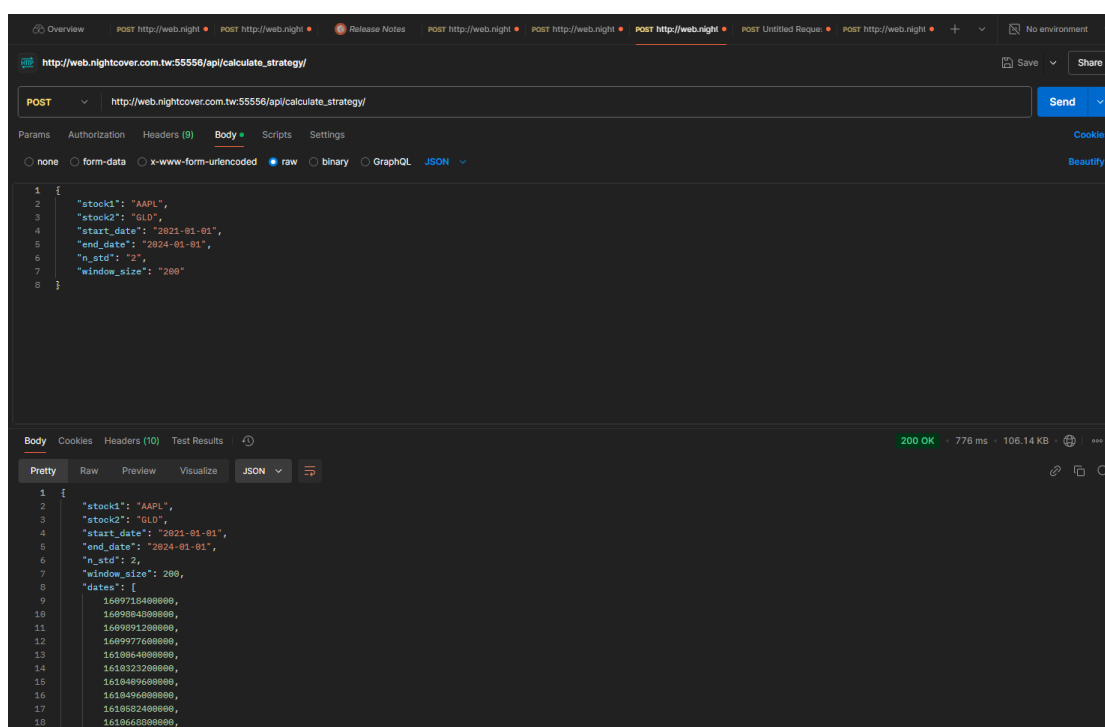
驗證與權限控制 (選配)

- 使用 Django Auth 系統進行 API 權限管理。
- 採用 Token 或 Session 驗證方式。

3.2 實作中學習到的知識與遭遇的困難

從這次作業開始就有感受到難度上升，因為感覺得出來老師和助教希望我們去架一套完整的系統，所以就必須將之前的架構整個大改，要能把 API 計算的部分另外切出去變成一個獨立的 API server，計算不會是網頁後端幫忙做，但我到 HW5 才理解是要這樣做，不然我都是把它放在同一個 Django 底下然後用不一樣的 route 設置。

3.3 結果截圖



3.4 程式使用方式

透過 Postman 發送 request 來測試 API

HW-4 Secured Function API/DATA API

4.1 作業目標

實作 Secured Function API

- 強化 API 安全性，透過 Token 認證機制限制 API 訪問。
- 使用者需先登入獲取 Access Token，之後透過 Token 使用 API。
- 支援 Token 更新（Refresh Token）機制，定時自動更新 Token。

實作 Secured Data API

- 使用 Token 認證限制資料存取。
- API 從資料庫中讀取資料並回傳。
- 需處理資料庫連線與 SQL 查詢，並防止 SQL Injection。

API 權限與驗證設計

- 使用 Django Rest Framework (DRF) 的權限模組來管理 API 存取權限。
- 實作 Token 驗證流程：
 - 登入取得 Token : `http://127.0.0.1:8000/token/`
 - 更新 Token : `http://127.0.0.1:8000/token/refresh/`

整合 APIConnector

- 使用 APIConnector 自動化 Token 獲取與更新，每 15 分鐘更新 Token。
- 在 Function API 與 Data API 請求時自動附帶有效 Token。

錯誤處理與安全性加強

- 處理 Token 過期、自動更新、錯誤 Token、未登入等狀況。
- 設定 API 回傳的錯誤碼（200、400、401、404）與相對應的錯誤訊息。

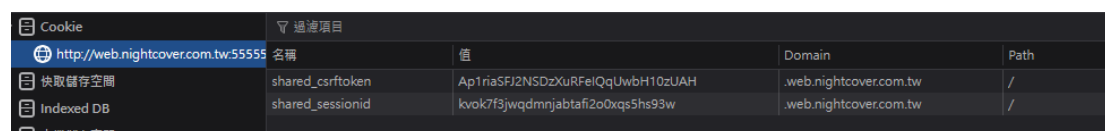
修改既有 API 為 Secured API

- 將前次作業（HW-3）中的 API 改為 Secured API。

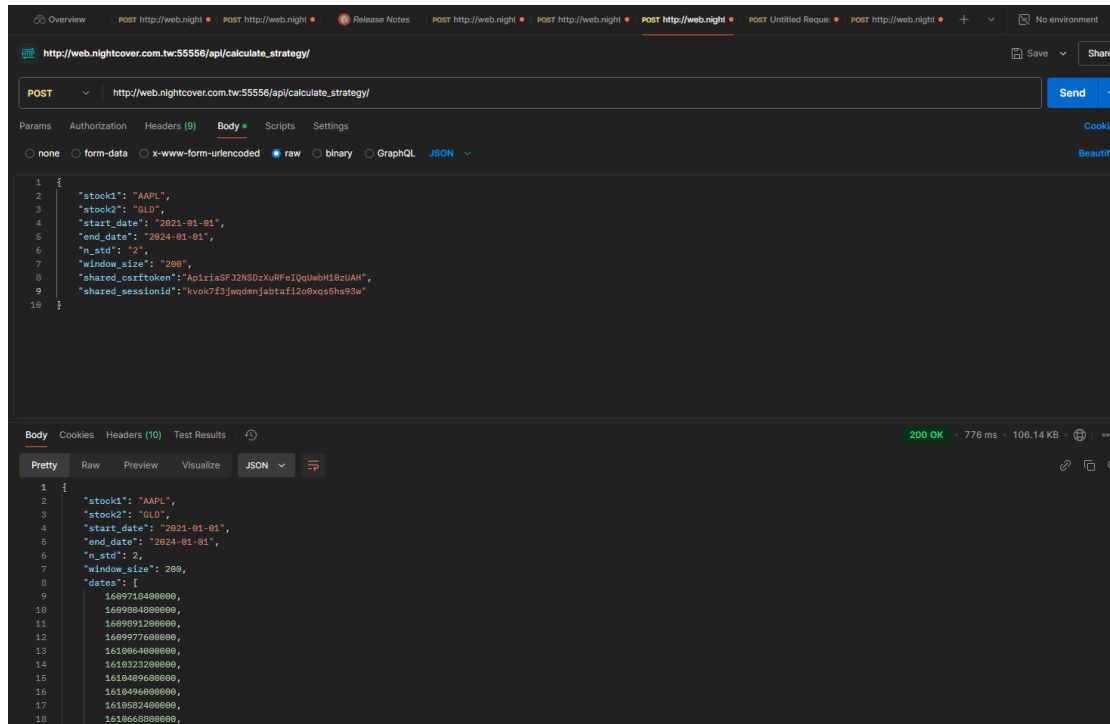
4.2 實作中學習到的知識與遭遇的困難

這次的作業開始就有一點沒有頭緒，尤其是那個定時要去更新 token，我查找了蠻多資料和問 ChatGPT 才知道該如何實作，另外我也在這個作業將自己的 PostgreSQL 給裝起來，老實說裝這個的過程還蠻費時的，加上我的程式是運行在 Linux 系統上，所以在設定上花了我不少時間。

4.3 結果截圖



Cookie		過濾項目			
	http://web.nightcover.com.tw:5555	名稱	值	Domain	Path
快取儲存空間		shared_csrf_token	Ap1naSFJ2NSDzXuRFeIQqUwbH10zUAH	.web.nightcover.com.tw	/
Indexed DB		shared_sessionid	kvok7f3jwqdmnjabtafi2o0xs5hs93w	.web.nightcover.com.tw	/
本機儲存空間					



5.1 作業目標

實作追蹤功能 (Tracker)

- 讓使用者能夠將感興趣的股票或交易對策略加入追蹤清單。
- 透過網站介面新增、移除、查看追蹤項目。
- 將追蹤結果（如每日損益、風險指標）視覺化展示。

自動化追蹤與通知功能

- 利用 Crontab 每天自動執行追蹤任務（平日一至五晚上 11 點執行 report.py）。
- Email 通知：每日自動寄送最新的追蹤結果（HTML 報告與 Excel 檔案）給使用者。

網站前後端整合

- 在網站新增 Monitor 分頁，讓使用者管理追蹤策略：
 - 新增追蹤：add_track() 將參數寫入資料庫並儲存結果。
 - 查看追蹤結果：get_track_list() 顯示已追蹤項目。
 - 刪除追蹤：remove_track() 刪除追蹤資料與結果。
- 透過 Highchart 繪製每日資產、損益與股價圖。

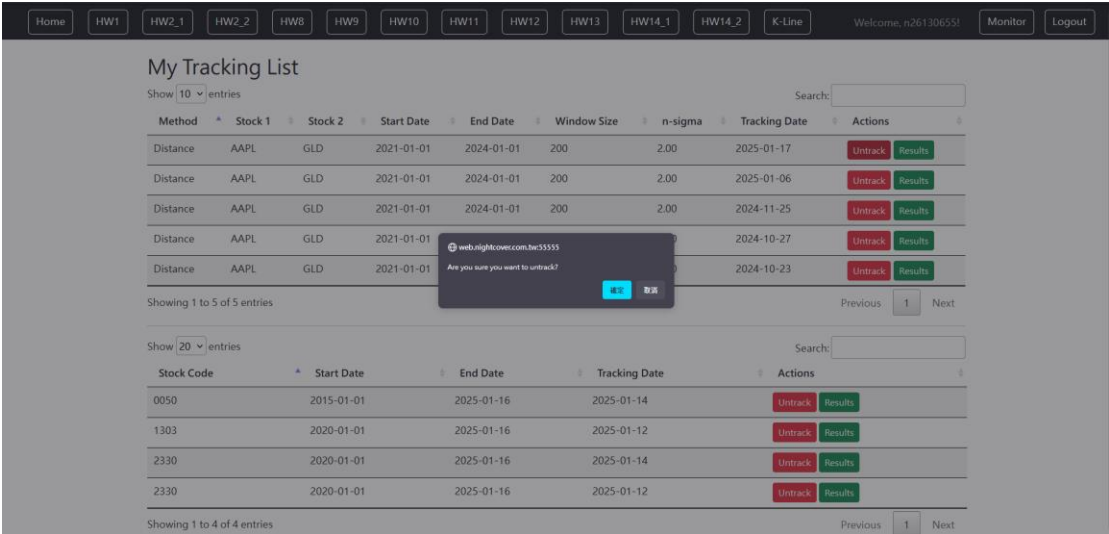
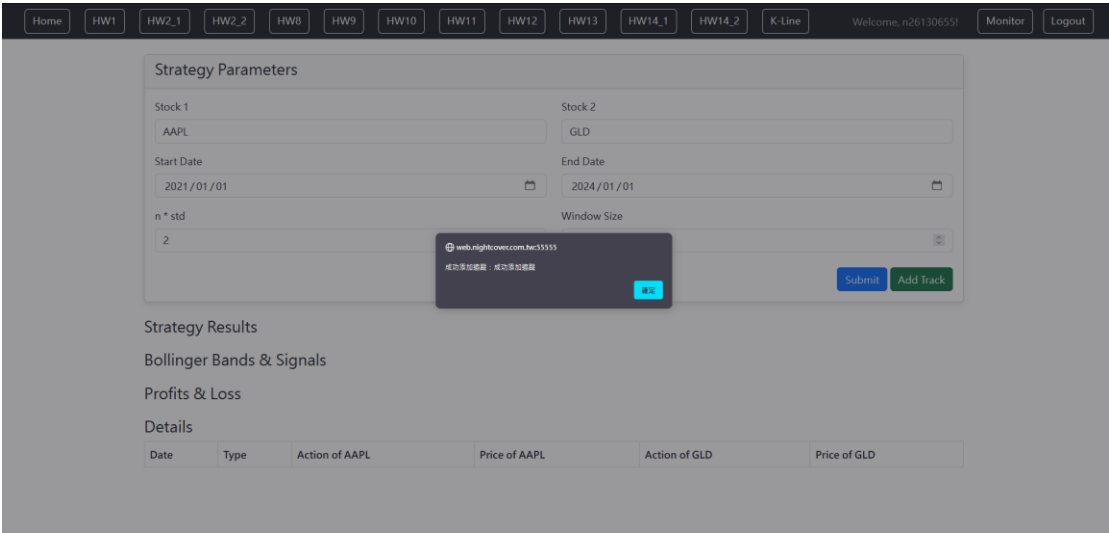
資料庫與 API 整合

- 建立 user_track 資料表儲存追蹤資訊。
- 使用 Function API 執行回測並儲存結果（JSON 格式）。
- 設計 API 與資料庫互動（新增、查詢、刪除追蹤紀錄）。

5.2 實作中學習到的知識與遭遇的困難

Tracker 這個作業我覺得應該是本學期最困難且麻煩的一次作業了，除了前後端和資料庫的問題要去處理，還有每天要定時透過 crontab 來計算每個用戶的追蹤結果並發送 E-mail，在實作過程中蠻多細節需要注意的，透過這次作業我學習到不少網頁 web 比較進階的技術。

5.3 結果截圖



HomeHW1HW2_1HW2_2HW8HW9HW10HW11HW12HW13HW14_1HW14_2K-Line

Welcome, n26130655!MonitorLogout

My Tracking List

Show 10 entries

Search:

Method	Stock 1	Stock 2	Start Date	End Date	Window Size	n-sigma	Tracking Date	Actions
Distance	AAPL	GLD	2021-01-01	2024-01-01	200	2.00	2025-01-17	Untrack Results
Distance	AAPL	GLD	2021-01-01	2024-01-01	200	2.00	2025-01-06	Untrack Results
Distance	AAPL	GLD	2021-01-01	2024-01-01	200	2.00	2024-11-25	Untrack Results
Distance	AAPL	GLD	2021-01-01				2024-10-27	Untrack Results
Distance	AAPL	GLD	2021-01-01				2024-10-23	Untrack Results

Showing 1 to 5 of 5 entries

Previous1Next

Show 20 entries

Search:

Stock Code	Start Date	End Date	Tracking Date	Actions
0050	2015-01-01	2025-01-16	2025-01-14	Untrack Results
1303	2020-01-01	2025-01-16	2025-01-12	Untrack Results
2330	2020-01-01	2025-01-16	2025-01-14	Untrack Results
2330	2020-01-01	2025-01-16	2025-01-12	Untrack Results

Showing 1 to 4 of 4 entries

Previous1Next

web.nightcover.com/bc5555

或點擊此處

☐ 不允許 web.nightcover.com/bc5555 再次向您顯示

✕

HomeHW1HW2_1HW2_2HW8HW9HW10HW11HW12HW13HW14_1HW14_2K-Line

Welcome, n26130655!MonitorLogout

My Tracking List

Showing 1 to 5 of 5 entries

Previous1Next

Strategy Results

Price & Spread

Zoom 1m 3m 6m YTD 1y All

Jan 4, 2021 → Jan 16, 2025

Bollinger Bands & Signals

Zoom 1m 3m 6m YTD 1y All

Jan 4, 2021 → Jan 16, 2025

Monitor Report 2025-01-16 收件匣 x



kymindouvic@gmail.com

寄給 f74091247 ▼

This is signals report.

1 個附件 • Gmail 已掃描檢查 ⓘ



Data Output Messages Notifications												
id [PK] bigint	user_id integer	method character varying (100)	stock1 character varying (10)	stock2 character varying (10)	start_date date	end_date date	n_std numeric (5,2)	window_size integer	track_date date	created_at timestamp with time zone		
1	2	Distance	AAPL	GLD	2021-01-01	2024-01-01	2.00	200	2024-10-22	2024-10-22 21:10:25.011663+08		
2	8	Distance	NVDA	GLD	2023-01-01	2024-01-01	2.00	200	2024-10-23	2024-10-23 23:44:03.482547+08		
3	9	Distance	AAPL	GLD	2021-01-01	2024-01-01	2.00	200	2024-10-23	2024-10-23 23:46:18.440745+08		
4	11	Distance	AAPL	GLD	2021-01-01	2024-01-01	1.00	300	2024-10-23	2024-10-23 23:50:01.029311+08		
5	12	Distance	AAPL	GLD	2021-01-01	2024-01-01	2.00	200	2024-10-23	2024-10-24 01:36:53.436982+08		
6	14	Distance	AAPL	GLD	2021-01-01	2024-01-01	2.00	200	2024-10-27	2024-10-27 18:58:00.553328+08		
7	16	Distance	AAPL	GLD	2021-01-01	2024-01-01	2.00	200	2024-10-29	2024-10-29 22:51:42.004521+08		
8	20	Distance	AAPL	GLD	2021-01-01	2024-01-01	2.00	200	2024-11-25	2024-11-25 20:32:03.583632+08		
9	23	Distance	AAPL	GLD	2021-01-01	2024-01-01	2.00	200	2025-01-06	2025-01-07 01:53:42.523672+08		
10	25	Distance	AAPL	GLD	2021-01-01	2024-01-01	2.00	200	2025-01-07	2025-01-07 19:20:58.844606+08		
11	26	Distance	AAPL	GLD	2021-01-01	2024-01-01	2.00	200	2025-01-07	2025-01-07 19:21:00.000275+08		

5.4 程式使用方式

- 按下 Add Track 將其加入追蹤清單
- 進入到 Monitor 頁面可查看追蹤的標的
- 按下 Untrack 刪除該筆追蹤
- 按下 Results 查看每日的計算結果
- 每日晚上 11 點更新並計算結果傳送到使用者信箱

HW-6 Docker

6.1 作業目標

使用 Docker 部署整體系統

- 將 Web 前端、後端 API、授權系統 (Auth)全部容器化。
- 實現模組化部署，確保各服務（如 /correlation、/Auth、/usFunc）獨立運作並互相連接。

設計與建構 Dockerfile

- 撰寫各模組的 Dockerfile，指定 Python 環境與依賴安裝流程。
- 確保 Django 應用程式能在容器內部啟動並運作。

整合 Docker Compose

- 撰寫 docker-compose.yml 管理多個容器（Web、Auth、API、DB）。
- 設定容器間的連線、端口對應（Port Binding），確保外部可透過對應端口存取各服務。

設定環境變數與資料卷（Volume）

- 配置容器間共享的環境變數與資料卷（如資料庫資料夾與 API 配置）。
- 將敏感資訊（如資料庫帳密、API URL）設置為環境變數。

啟動與測試容器

- 使用 Docker 指令或 docker-compose 啟動、關閉、重啟整個系統。
- 測試各模組功能是否正常運作（網站顯示、API 回應、資料庫連線、追蹤功能）。

6.2 實作中學習到的知識與遭遇的困難

這次的作業我基本上沒遇到什麼太大的困難，因為之前就有使用過 Docker 和 K8S 的經驗，遇到比較大的問題是不同的 server(Auth、Web、API)要安裝的套件都不同，所以在寫 Dockerfile 製作 images 時花了一些功夫，尤其是因為 tablib 沒辦法直接透過 pip 安裝，還需要用 makefile 等等的來完成，透過這次作業也讓我重新學習了容器技術。

6.3 結果截圖

```
root@main:/home/ouvic/Eco_Web# docker-compose down
Stopping ecoweb-container ... done
Stopping auth-container ... done
Stopping funcapi-container ... done
Removing ecoweb-container ... done
Removing auth-container ... done
Removing funcapi-container ... done
root@main:/home/ouvic/Eco_Web# docker images
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
funcapi-app   v1        67377d175721   39 hours ago   1.47GB
ecoweb-app    v1        4813064c3ce2   39 hours ago   1.47GB
<none>        <none>    1f834aec25b7   39 hours ago   1.46GB
<none>        <none>    e8821542ec67   40 hours ago   1.5GB
<none>        <none>    89bafae63586   2 days ago     1.55GB
<none>        <none>    eca9107883d1   2 days ago     1.5GB
<none>        <none>    be459633d8b5   2 days ago     1.5GB
auth-app      v1        f3674f9f0eee   3 days ago     1.47GB
<none>        <none>    15d657c43b51   3 days ago     1.47GB
<none>        <none>    153a849e9e89   3 days ago     1.47GB
<none>        <none>    d03b825eecdb   3 days ago     1.47GB
<none>        <none>    7966cb573abe   3 days ago     1.28GB
<none>        <none>    57f325093684   3 days ago     1.28GB
root@main:/home/ouvic/Eco_Web# docker-compose up -d
Creating funcapi-container ... done
Creating ecoweb-container ... done
Creating auth-container ... done
root@main:/home/ouvic/Eco_Web# docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS          NAMES
b4190d12c5ec   ecoweb-app:v1 "python manage.py ru..." 2 seconds ago Up 2 seconds             ecoweb-container
d8a9ef3fb066   funcapi-app:v1 "python manage.py ru..." 2 seconds ago Up 2 seconds             funcapi-container
27738afb3c8    auth-app:v1    "python manage.py ru..." 2 seconds ago Up 2 seconds             auth-container
root@main:/home/ouvic/Eco_Web#
```

```
🔥 docker-compose.yml
1  version: '3'
2
3  services:
4    auth:
5      image: auth-app:v1
6      container_name: auth-container
7      network_mode: host
8      restart: always
9
10   funcapi:
11     image: funcapi-app:v1
12     container_name: funcapi-container
13     network_mode: host
14     restart: always
15
16   ecoweb:
17     image: ecoweb-app:v1
18     container_name: ecoweb-container
19     network_mode: host
20     restart: always
21
```

6.4 程式使用方式

Step1. 將 Auth、網頁後端、API 個別寫一個 Dockerfile

Step2. 撰寫 docker-compose

Step3. 同時啟動三個容器服務

HW-7 選股 fin 模組

7.1 作業目標

熟悉 Finlab 選股模組的基本操作

- 安裝與導入 finlab 模組與所需依賴（如 pandas、numpy、ta-lib、pandas-ta）。
- 登入 Finlab 平台並抓取股票數據進行分析。

實作選股策略

- 撰寫選股策略條件，如：
 - 近 2 月平均營收創 12 個月新高
 - 五日成交均量大於 500 張
 - 近 60 日股價波動區間小於 30%
 - 近 6 月營收月增率小於 0%

組合選股條件並篩選標的

- 使用 `combine()` 函式設計複合選股條件（部分條件符合、完全符合、排除條件）。
- 使用 `largest()` 函式取出符合條件的前 N 名股票。

擴充與自訂選股條件

- 新增選股條件（如：股價創 5 日新低）：
 - 新增計算邏輯至 `calculator_interface.py`、`process_data_manager.py`。
 - 將條件整合至 `filter_interface.py` 提供篩選。

選股結果視覺化與分析

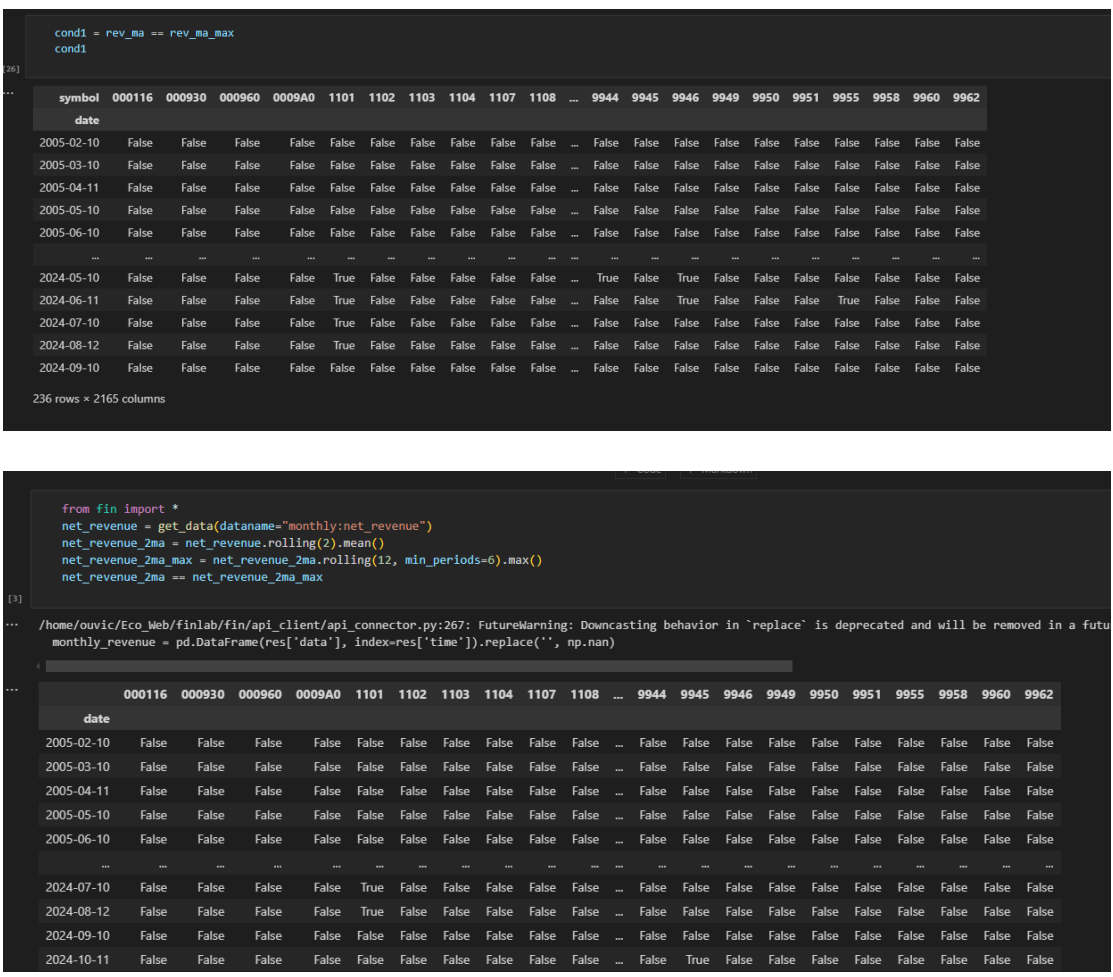
- 顯示選股結果並結合股價走勢圖、技術指標圖。
- 評估策略有效性並可視化展示。

7.2 實作中學習到的知識與遭遇的困難

這次的作業主要就是跑助教附的程式，然後透過自己設定的選股條件來檢驗

Fin 模組是否有篩選成功，這次的作業讓我認識到有 FinLab 這個平台，上面有許多和股市相關的程式套件可以使用，我在上面註冊了一個帳號測試。

7.3 結果截圖



7.4 程式使用方式

使用助教提供的 jupyter notebook 程式檔案測試 Finlab 套件

HW-8 選股 fin 網站

8.1 作業目標

建構選股網站

- 使用 Django 搭建一個選股平台，整合 Fin 模組作為選股核心。
- 實現使用者在網頁上輸入選股條件並查詢結果。

整合 Function API

- 設計 API 提供選股功能，將使用者輸入的選股條件轉換為 JSON，傳遞給後端進行計算。
- API 功能：
 - /finlab_start/：初始化選股條件
 - /finlab_select/：執行選股運算並回傳結果

設計前後端互動流程

- 前端：使用者輸入選股條件 → 點擊搜尋 → 傳送條件到後端
- 後端：接收條件 → 組合選股策略 → 返回結果

8.2 實作中學習到的知識與遭遇的困難

這次的作業主要是把上次 HW7 的 Fin 模組和 FinLab 套件應用到網站上，所以將前端刻完，後端跟 API 來接收條件回傳結果即可，過程中沒有遇到什麼太大的困難，主要的問題是 FinLab 免費版有額度限制，所以每天測試程式會有限制，不能一直無限制地拿取選股資料。

8.3 結果截圖

[Home](#) [HW1](#) [HW2_1](#) [HW2_2](#) [HW8](#) [HW9](#) [HW10](#) [HW11](#) [HW12](#) [HW13](#) [HW14_1](#) [HW14_2](#) [K-Line](#) Welcome, n26130655! [Monitor](#) [Logout](#)

FinLab 股票篩選

篩選條件

當月營收 (大於)

收盤價 (大於)

ROA (大於)

[開始篩選](#)

篩選結果

Show entries

Search:

股票代碼	最新收盤價
1476	525
1519	540
1590	763
2059	1125
2207	618
2327	577
2330	904
2357	527
2454	1130
3008	2620

Showing 1 to 10 of 37 entries

[Previous](#) [1](#) [2](#) [3](#) [4](#) [Next](#)

8.4 程式使用方式

Step1. 前端網頁輸入當月營收、收盤價、ROA 等篩選條件

Step2. 後端計算前端發送 Post 請求並回傳結果

Step3. 前端頁面顯示選股結果

HW-9 股票定價策略

9.1 作業目標

實作四種股票定價策略

- 股利法 (Dividend Discount Model, DDM)
- 高低價法 (Historical High-Low Price Method)
- 本淨比法 (Price-to-Book Ratio, PBR)
- 本益比法 (Price-to-Earnings Ratio, PER)

動態計算股票價格區間

- 根據歷史數據 (10 年內) 計算每支股票的：

- 便宜價 (Undervalued Price)
 - 合理價 (Fair Price)
 - 昂貴價 (Overvalued Price)
- 與即時股價比較，判斷是否進場。

資料收集與處理

- 利用 爬蟲技術 (如 twstock 套件) 或 API 獲取歷史股價、EPS、BPS、股利等數據。
- 整理資料並應用四種定價策略計算價格區間。

設計 API 並整合前後端

- 後端 API：接收股票代號與年份，計算並回傳各策略的定價結果。
- 前端展示：以表格或圖表 (如 Highcharts) 展示四種策略的計算結果與即時股價比較。

9.2 實作中學習到的知識與遭遇的困難

這次的作業是要去實作定價策略，在畫製 highcharts 跟前後端的設計比較沒有問題，遭遇的困難是我之前沒有接觸過太多的金融知識，花了蠻多時間去看網路上的影片跟文章，理解這些定價策略該如何實作，另外有些資料不能透過 yfinance 直接抓，所以還得學一些網路爬蟲的技術。

9.3 結果截圖



9.4 程式使用方式

Step1. 前端網頁輸入股票代碼、歷史幾年資料

Step2. 後端計算前端發送 Post 請求並回傳結果

Step3. 前端頁面渲染圖表以及計算結果

HW-10 本益比河流圖

10.1 作業目標

了解與實作本益比河流圖 (PER River Chart)

- 公式：
 - 本益比 (PER) = 股價 ÷ 每股盈餘 (EPS)
 - 股價 = 本益比 (PER) × 每股盈餘 (EPS)
- 利用不同倍數的本益比 (如 15X、20X、25X) 計算對應股價，繪製股價隨 EPS 變化的趨勢圖。

動態計算與視覺化

- 根據歷史數據（如 EPS、PER）動態計算股票的便宜價、合理價、昂貴價。
- 以河流圖方式呈現不同倍數本益比對應的股價走勢。

倍率設定（PER 倍率調整）

- 歷史數據分析：選擇過去 3 年、5 年、10 年的平均 PER 值，設定高估與低估區間（如 10X、15X、20X、25X）。
- 行業特性估值：根據產業類別（穩定型或成長型）調整 PER 倍率範圍。

前端視覺化呈現

- 使用 Highcharts 或其他圖表工具，動態顯示本益比河流圖。
- 橫軸：可切換顯示單位（月、季、年）。
- 縱軸：對應不同倍數的股價區間（便宜價、合理價、昂貴價）。

選擇個股並即時更新數據

- 以台積電（2330）為範例，顯示即時股價與本益比河流圖的比較（如 $EPS = 40.01$ 、股價 = 1016.5）。
- 提供股票代號輸入，計算並更新本益比河流圖。

10.2 實作中學習到的知識與遭遇的困難

這次的作業是要實作河流圖，跟上次作業遇到的問題類似，因為我沒有很堅實的金融知識，所以也是花了一段時間理解什麼是本益比河流圖，在放上網站系統上則是沒有太大的困難。

10.3 結果截圖

本益比河流圖

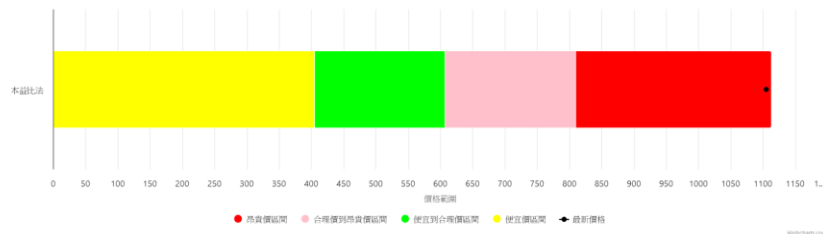
股票代號&名稱:

2330

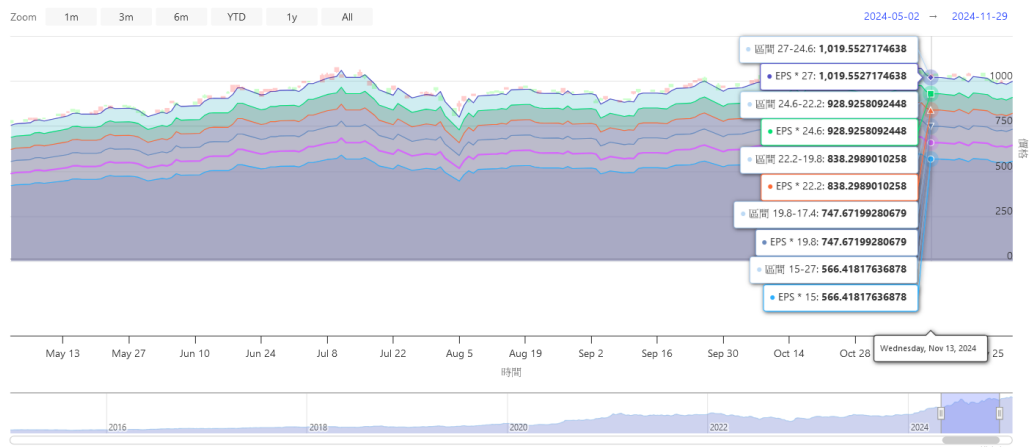
搜尋

最新價格: 1105

本益比河流圖結果



本益比河流圖



10.4 程式使用方式

Step1. 前端網頁輸入股票代碼

Step2. 後端計算前端發送 Post 請求並回傳結果

Step3. 前端頁面渲染圖表以及計算結果

HW-11 天花板地板線

11.1 作業目標

實作天花板線 (Resistance Line) 與地板線 (Support Line) 計算

- 天花板線：預測股價的壓力線，判斷股價可能無法突破的區間。
- 地板線：預測股價的支撐線，判斷股價可能不會跌破的區間。

三種計算方法

- 方法一：基於乖離率（Bias）的天花板與地板線
 - 天花板線 = 正乖離率第 95% 分位數 × 20 日均線 + 20 日均線
 - 地板線 = 負乖離率第 5% 分位數 × 20 日均線 + 20 日均線
- 方法二：乖離率平均與標準差調整
 - 天花板線 = 20 日均線 + 20 日均線 × (正乖離率平均 + 2 倍標準差)
 - 地板線 = 20 日均線 + 20 日均線 × (負乖離率平均 - 2 倍標準差)
- 方法三：極端值 + 交易量條件
 - 天花板線 = 正乖離率第 99% 分位數 × 20 日均線 + 20 日均線
 - 地板線 = 負乖離率第 1% 分位數 × 20 日均線 + 20 日均線
 - 條件：當日成交量需大於 20 日均量的 2 倍。

數據擷取與處理

- 輸入參數：股票代號、資料起始日、MA 長度、MA 類型（SMA/WMA）、計算方法（1/2/3）。
- 資料來源：使用 yfinance 或其他金融資料 API 取得股價資料（開盤價、最高價、最低價、收盤價、成交量）。

視覺化與結果呈現

- 繪製天花板線與地板線。
- 標記突破點（當股價突破天花板線或跌破地板線）。
- 成交量直方圖：在符合條件的交易量以紅色標示。

網站整合與 API 設計

- 後端 API：接收使用者輸入並計算天花板線與地板線。
- 前端：展示結果圖表（天花板線、地板線、成交量）與突破點。

11.2 實作中學習到的知識與遭遇的困難

這次的作業開始要分析技術面的指標，但我是第一次接觸這些東西，好在有 talib 這個套件，所以我就稍微理解這些指標的作用然後就直接使用 talib 來幫我算，並根據助教的投影片來實作一些不同的方法。

11.3 結果截圖

天花板地板線

Strategy Parameters

股票代號: 2330	起始日期: 2020/01/01	結束日期: 2025/01/01
MA長度: 20	MA類型: SMA	方法: 方法一
<input type="button" value="存單"/>	SMA	
	WMA	



11.4 程式使用方式

Step1. 前端網頁輸入股票代碼、初始日期、結束日期、MA 長度、MA 類型、方法

Step2. 後端計算前端發送 Post 請求並回傳結果

Step3. 前端頁面渲染圖表以及計算結果和成交量

HW-12 KD 指標、MACD 指標、布林通道

12.1 作業目標

實作技術分析指標

- KD 指標 (Stochastic Oscillator)：用於判斷股價是否超買或超賣。
- MACD 指標 (Moving Average Convergence Divergence)：用於判斷多空趨勢與動能轉折。
- 布林通道 (Bollinger Bands)：用於觀察股價波動範圍與趨勢。

技術指標計算與視覺化

- 計算並繪製各指標的趨勢圖，與股價結合分析。
- 在圖表中標示進出場訊號（如黃金交叉、死亡交叉、突破通道）。

前後端整合

- 後端 API：輸入股票代號、日期範圍，回傳指標數據與買賣訊號。
- 前端呈現：圖表動態展示指標與進出場訊號。

12.2 實作中學習到的知識與遭遇的困難

這次的作業跟上次一樣都是分析相關的技術指標，也是稍微理解指標的核心概念並直接使用 talib 計算出結果，過程中沒遇到什麼太大的困難。

12.3 結果截圖

Home HW1 HW2_1 HW2_2 HW8 HW9 HW10 HW11 HW12 HW13 HW14_1 HW14_2 K-Line Welcome, n26130659! Monitor Logout

KD MACD 布林通道

Strategy Parameters

股票代號:	2330	起始日期:	2020/01/01	結束日期:	2025/01/01
KD參數					
Fast K 週期:	9	Slow K 週期:	3	Slow D 週期:	3
MACD參數					
快速週期:	12	慢速週期:	26	訊號週期:	9
布林通道參數					
時間週期:	5	上軌標準差倍數:	2	下軌標準差倍數:	2

View



12.4 程式使用方式

Step1. 前端網頁輸入股票代碼、初始日期、結束日期、KD 參數、MACD 參數、布林通道參數

Step2. 後端計算前端發送 Post 請求並回傳結果

Step3. 前端頁面渲染圖表以及計算結果和成交量

HW-13 RSI 指標、ADX 指標、DMI 指標

13.1 作業目標

實作技術指標計算

- RSI 指標 (Relative Strength Index)：判斷股價是否進入超買或超賣狀態。
- ADX 指標 (Average Directional Movement Index)：衡量市場趨勢的強度。
- DMI 指標 (Directional Movement Index)：判斷市場的多空趨勢方向（透過 +DI 和 -DI 交叉）。

視覺化技術指標與進出場訊號

- RSI：超過 70（超買）或低於 30（超賣）時標示。
- ADX + DMI：標示黃金交叉（+DI 上穿 -DI，買進）與死亡交叉（-DI 上穿 +DI，賣出）。
- 動態圖表展示各指標隨時間變化，結合股價趨勢。

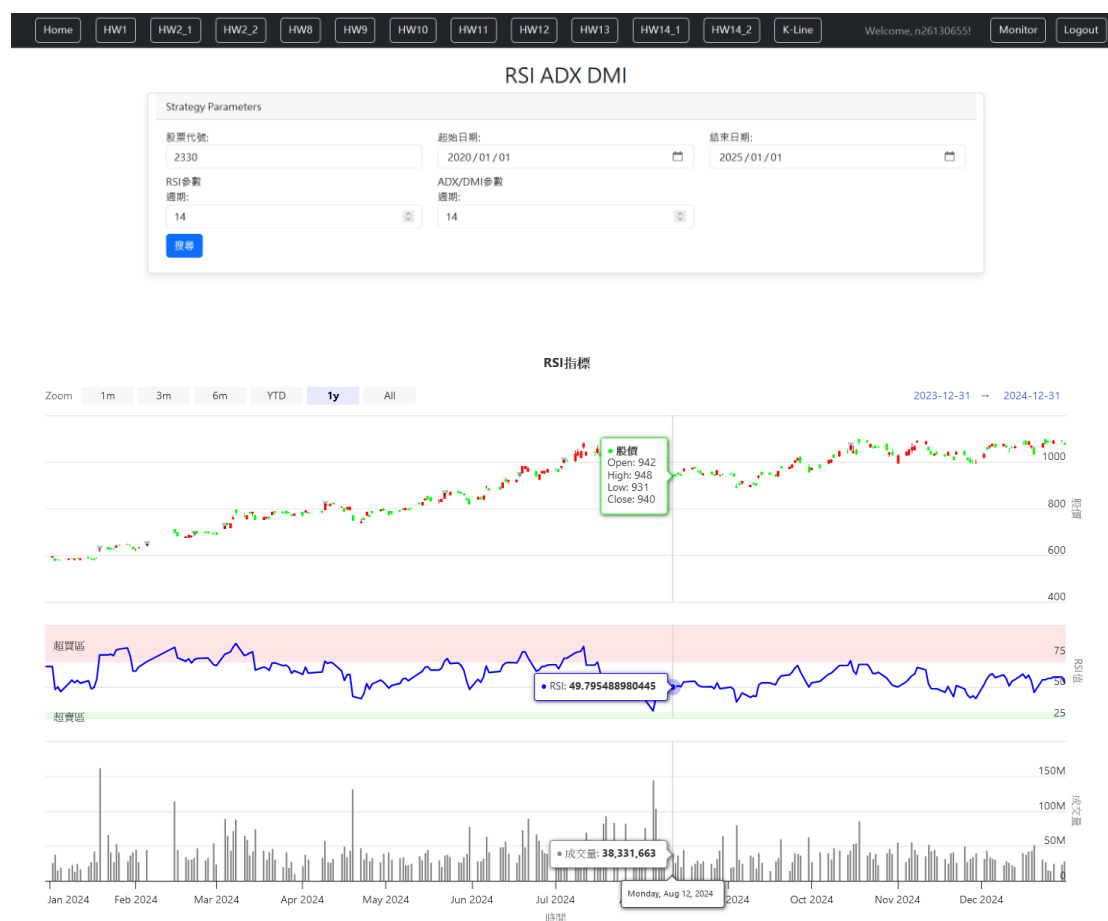
API 與網站整合

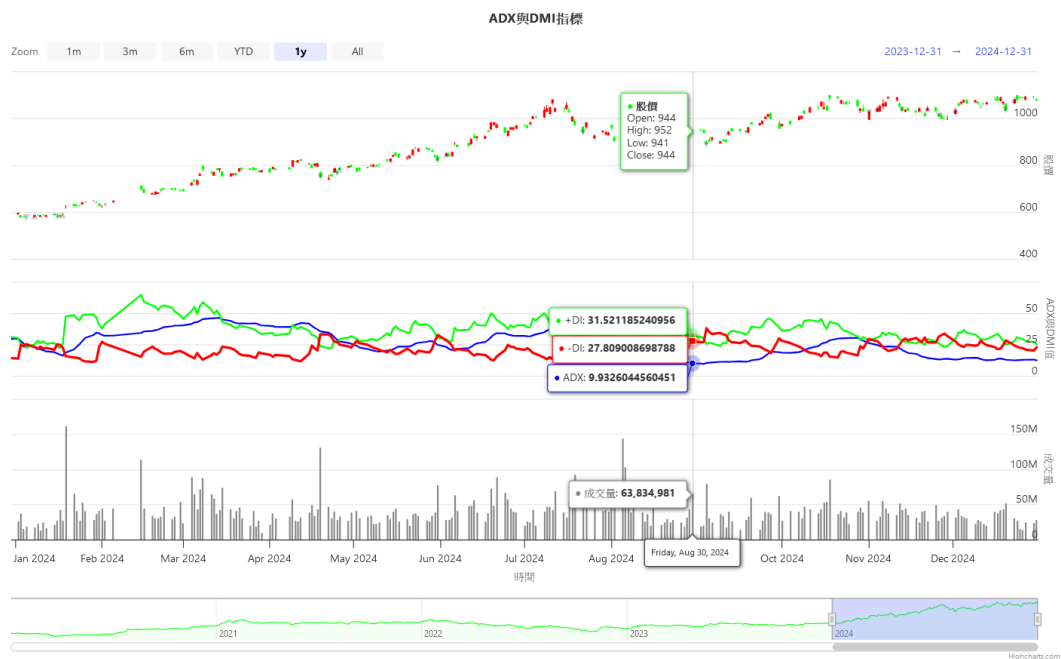
- 後端 API：輸入股票代號、日期區間，回傳 RSI、ADX、DMI 計算結果。
- 前端展示：透過圖表（Highcharts 或 Matplotlib）呈現技術指標與進出場訊號。

13.2 實作中學習到的知識與遭遇的困難

這次的作業跟上次一樣都是分析相關的技術指標，也是稍微理解指標的核心概念並直接使用 talib 計算出結果，過程中沒遇到什麼太大的困難。

13.3 結果截圖





13.4 程式使用方式

Step1. 前端網頁輸入股票代碼、初始日期、結束日期、RSI 參數、ADX/DMI 參數

Step2. 後端計算前端發送 Post 請求並回傳結果

Step3. 前端頁面渲染圖表以及計算結果和成交量

HW-14 FinalProject、K 線型態

14.1 作業目標

整合股票進出場訊號偵測工具

- 綜合應用已學習的各種策略與技術指標，設計一套股票進出場訊號偵測系統。
- 包含內容：
 - 股票定價策略：股利法、高低價法、本淨比法、本益比法、本益比河流圖。

- 技術指標：天花板地板線、KD、MACD、布林通道、RSI、ADX、DMI、K 線型態。

K 線型態辨識與應用

- K 線型態偵測：利用 Talib 庫偵測經典 K 線型態。
- 進出場策略設計：根據 K 線型態產生買進與賣出訊號。

視覺化與結果呈現

- 將偵測到的進出場訊號以圖表呈現（如 Highcharts 或 Matplotlib）。
- 繪製 Heat Map 展示五種定價策略的結果（便宜價、合理價、昂貴價）。

自動追蹤與通知功能

- 建立 Tracker 功能，讓用戶訂閱股票清單，每日自動計算進出場訊號。
- 計算結果存檔並透過 Email 通知用戶。

14.2 實作中學習到的知識與遭遇的困難

這次的作業是期末的 project，老師說有做出來就會通過，主要有兩部分，一個是定價策略，另一個是個股分析功能和 Tracker，那整個作業其實就是將之前學習的內容都整合在 project 裡面，我在前面進度都有盡量跟上，所以 final project 做的過程還算是順利，另外就熱力圖的呈現因為是有不同的時間區間，我還費了蠻多時間去想該怎麼實作，个股分析功能倒是沒有問題，就是將之前的那些技術指標都拿來用，另外還要多增加一個 K 線型態，最後則是 Tracker，Tracker 的實作我是上課聽老師的說明才知道要如何呈現，不然做出來感覺跟大家的都不一樣，最後也有順利弄出來，可能還是會有點問題，但至少有個樣子。

14.3 結果截圖

HW14_1

股票定價策略: 便宜價 合理價 昂貴價

Strategy Parameters

股票代號&名稱:

2330

起始日期:

2024/12/30

結束日期:

2025/01/18

搜尋

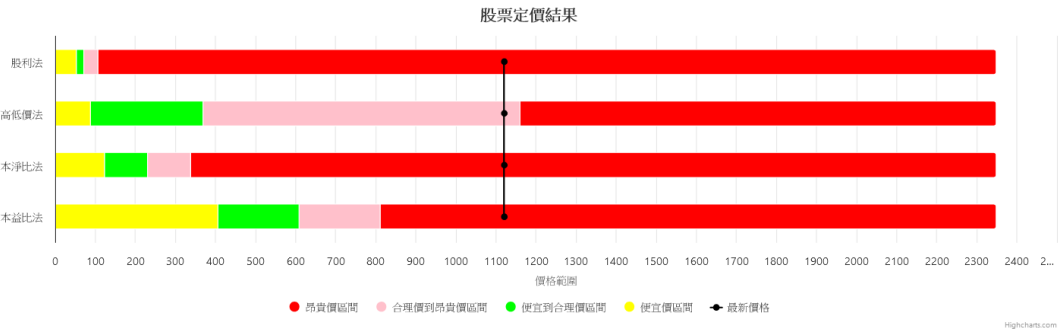
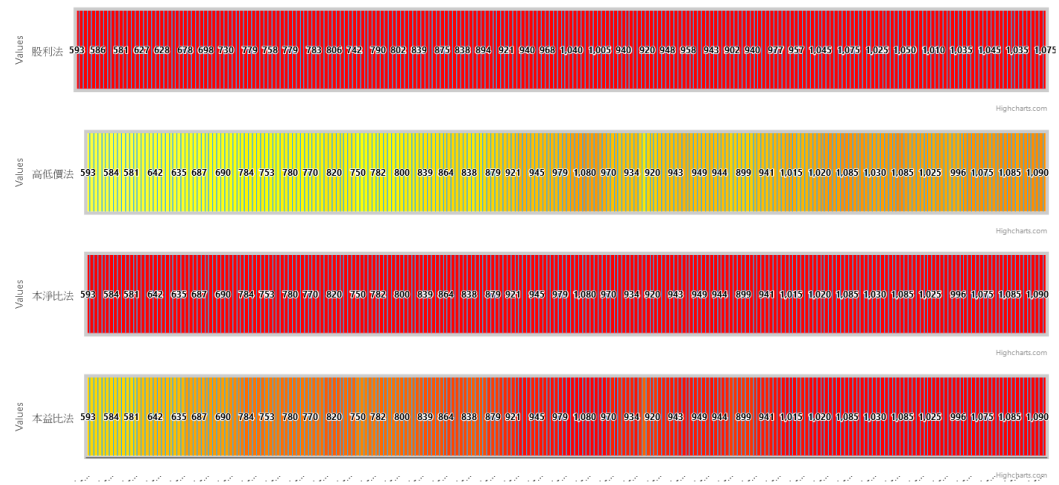
短週期(2024. 12. 30~2025. 01. 17)

最新價格: 1120



長週期(2024. 01. 01~2025. 01. 01)

最新價格: 1120



HW14_2

股票進出場策略

Strategy Parameters

股票代號: 2330 起始日期: 2019/01/01 結束日期: 2025/01/16

天花板地板線參數

MA長度: 20 MA類型: SMA 方法: 方法一

KD參數

Fast K 週期: 9 Slow K 週期: 3 Slow D 週期: 3

MACD參數

快速週期: 12 慢速週期: 26 訊號週期: 9

布林通道參數

時間週期: 5 上軌標準差倍數: 2 下軌標準差倍數: 2

RSI參數

週期: 14

ADX/DMI參數

週期: 14

搜尋 Add Track







Home HW1 HW2_1 HW2_2 HW8 HW9 HW10 HW11 HW12 HW13 HW14_1 HW14_2 K-Line Welcome, n26130655! Monitor Logout

股票進出場策略

Strategy Parameters

股票代號: 2330 開始日期: 2019/01/01 結束日期: 2025/01/16

天花板地板線參數
MA長度: 20 MA類型: SMA 方法: 方法一

KD參數
Fast K 週期: 9 Slow D 週期: 3

MACD參數
快速週期: 12 訊號週期: 9

布林通道參數
時間週期: 5 上軌標準差倍數: 2 下軌標準差倍數: 2

RSI參數
週期: 14 ADX/DMI參數
週期: 14

新增 Add Track

HomeHW1HW2_1HW2_2HW8HW9HW10HW11HW12HW13HW14_1HW14_2K-LineWelcome, n26130655!MonitorLogout

My Tracking List

Showing 1 to 4 of 4 entries

Show 10 entries

Search:

Method	Stock 1	Stock 2	Start Date	End Date	Window Size	n-sigma	Tracking Date	Actions
Distance	AAPL	GLD	2021-01-01	2024-01-01	200	2.00	2025-01-06	<div>UntrackResults</div>
Distance	AAPL	GLD	2021-01-01	2024-01-01	200	2.00	2024-11-25	<div>UntrackResults</div>
Distance	AAPL	GLD	2021-01-01	2024-01-01	200	2.00	2024-10-27	<div>UntrackResults</div>
Distance	AAPL	GLD	2021-01-01	2024-01-01	200	2.00	2024-10-23	<div>UntrackResults</div>

Showing 1 to 4 of 4 entries

Previous1Next

Show 20 entries

Search:

Stock Code	Start Date	End Date	Tracking Date	Actions
0050	2015-01-01	2025-01-15	2025-01-14	<div>UntrackResults</div>
1303	2020-01-01	2025-01-15	2025-01-12	<div>UntrackResults</div>
2330	2019-01-01	2025-01-16	2025-01-16	<div>UntrackResults</div>
2330	2020-01-01	2025-01-15	2025-01-14	<div>UntrackResults</div>
2330	2020-01-01	2025-01-15	2025-01-12	<div>UntrackResults</div>

Showing 1 to 5 of 5 entries

Previous1Next

HomeHW1HW2_1HW2_2HW8HW9HW10HW11HW12HW13HW14_1HW14_2K-LineWelcome, n26130655!MonitorLogout

My Tracking List

Showing 1 to 4 of 4 entries

Show 10 entries

Search:

Method	Stock 1	Stock 2	Start Date	End Date	Window Size	n-sigma	Tracking Date	Actions
Distance	AAPL	GLD	2021-01-01	2024-01-01	200	2.00	2025-01-06	<div>UntrackResults</div>
Distance	AAPL	GLD	2021-01-01	2024-01-01	200	2.00	2024-11-25	<div>UntrackResults</div>
Distance	AAPL	GLD	2021-01-01	2024-01-01	200	2.00	2024-10-27	<div>UntrackResults</div>
Distance	AAPL	GLD	2021-01-01	2024-01-01	200	2.00	2024-10-23	<div>UntrackResults</div>

Showing 1 to 4 of 4 entries

Previous1Next

Show 20 entries

Search:

Stock Code	Start Date	End Date	Tracking Date	Actions
0050	2015-01-01	2025-01-15	2025-01-14	<div>UntrackResults</div>
1303	2020-01-01	2025-01-15	2025-01-12	<div>UntrackResults</div>
2330	2019-01-01	2025-01-16	2025-01-16	<div>UntrackResults</div>
2330	2020-01-01	2025-01-15	2025-01-14	<div>UntrackResults</div>
2330	2020-01-01	2025-01-15	2025-01-12	<div>UntrackResults</div>

Showing 1 to 5 of 5 entries

Previous1Next

HomeHW1HW2_1HW2_2HW8HW9HW10HW11HW12HW13HW14_1HW14_2K-LineWelcome, n26130655!MonitorLogout

My Tracking List

Showing 1 to 4 of 4 entries

Show 10 entries

Search:

Method	Stock 1	Stock 2	Start Date	End Date	Window Size	n-sigma	Tracking Date	Actions
Distance	AAPL	GLD	2021-01-01	2024-01-01	200	2.00	2025-01-06	<div>UntrackResults</div>
Distance	AAPL	GLD	2021-01-01	2024-01-01	200	2.00	2024-11-25	<div>UntrackResults</div>
Distance	AAPL	GLD	2021-01-01	2024-01-01	200	2.00	2024-10-27	<div>UntrackResults</div>
Distance	AAPL	GLD	2021-01-01	2024-01-01	200	2.00	2024-10-23	<div>UntrackResults</div>

Showing 1 to 4 of 4 entries

Previous1Next

Show 20 entries

Search:

Stock Code	Start Date	End Date	Tracking Date	Actions
0050	2015-01-01	2025-01-15	2025-01-14	<div>UntrackResults</div>
1303	2020-01-01	2025-01-15	2025-01-12	<div>UntrackResults</div>
2330	2019-01-01	2025-01-16	2025-01-16	<div>UntrackResults</div>
2330	2020-01-01	2025-01-15	2025-01-14	<div>UntrackResults</div>
2330	2020-01-01	2025-01-15	2025-01-12	<div>UntrackResults</div>

Showing 1 to 5 of 5 entries

Previous1Next

Strategy Results

參數設定

參數	值
stock1	0050
MA length	20
MA type	sma
method	method1
fastk_period	9
slowk_period	3
slowd_period	3
fastperiod	12
slowperiod	26
signalperiod	9
timeperiod	5
nbdevup	2
nbdevdn	2
rsi_period	14
adx_period	14

kymindouvic@gmail.com

寄給 我

1月14日 週二 下午11:00 (2 天前)

☆

↶

⋮

🗨️ 翻譯成中文 (繁體)

✕

This is signals report.

1 個附件 • Gmail 已掃描檢查

✕

2025-01-14_signals_report.xlsx

5.3 KB

📄

📎

✎

	A	B	C	D	E	F	G	H	I
1	stock_code	track_date	ilingfloor_sig	kd_signal	macd_signal	bool_signal	rsi_signal	adx_signal	dline_pattern
2	2330	2025-01-13	No signal	No signal	sell	No signal	No signal	sell	bearish
3	1303	2025-01-13	No signal	No signal	No signal	No signal	No signal	No signal	bullish
4	2454	2025-01-13	No signal	No signal	sell	No signal	No signal	No signal	bearish
5	2454	2025-01-13	No signal	No signal	sell	buy	No signal	No signal	bearish
6	0050	2025-01-14	No signal	No signal	sell	No signal	No signal	sell	bearish
7	2330	2025-01-14	No signal	No signal	sell	No signal	No signal	sell	bearish
8									
9									
10									
11									
12									
13									
14									
15									
16									
17									
18									
19									
20									

Data Output														
	id [PK] integer	user_id integer	stock_code character varying (10)	start_date date	end_date date	ma_length integer	ma_type character varying (20)	method character varying (20)	fastk_period integer	slowk_period integer	slowd_period integer	fastperiod integer	slowp integer	
1	7	2	2330	2020-01-01	2025-01-17	20	sma	method1	9	3	3	12		
2	8	2	1303	2020-01-01	2025-01-17	20	wma	method2	9	3	3	12		
3	9	1	2330	2020-01-01	2025-01-17	20	sma	method1	9	3	3	12		
4	10	1	1303	2020-01-01	2025-01-17	20	sma	method1	9	3	3	12		
5	13	4	2330	2019-01-01	2025-01-17	20	sma	method1	9	3	3	12		
6	14	4	1303	2019-01-01	2025-01-17	20	sma	method1	9	3	3	12		
7	15	2	2330	2020-01-01	2025-01-17	20	sma	method1	9	3	3	12		
8	17	2	2330	2021-01-05	2025-01-17	30	wma	method2	9	3	3	12		
9	18	2	2330	2021-01-01	2025-01-17	20	sma	method1	9	3	3	12		
10	19	1	0050	2015-01-01	2025-01-17	20	sma	method1	9	3	3	12		
11	20	1	2330	2020-01-01	2025-01-17	20	sma	method1	9	3	3	12		

14.4 程式使用方式

HW14_1:

Step1. 前端網頁輸入股票代碼、初始日期、結束日期

Step2. 後端計算前端發送 Post 請求並回傳結果

Step3. 前端頁面渲染圖表以及計算結果

HW14_2:

分析功能

Step1. 前端網頁輸入股票代碼、初始日期、結束日期、天花板地板線參數、KD 參數、MACD 參數、布林通道參數、RSI 參數、ADX/DMI 參數

Step2. 後端計算前端發送 Post 請求並回傳結果

Step3. 前端頁面渲染圖表以及計算結果和成交量

Tracker 功能

- 按下 Add Track 將其加入追蹤清單
- 進入到 Monitor 頁面可查看追蹤的標的
- 按下 Untrack 刪除該筆追蹤
- 按下 Results 查看每日的計算結果
- 每日晚上 11 點更新並計算結果傳送到使用者信箱

修課心得

這門課我是在選課第三階段才看到的，那時候想說對金融知識有興趣又剛好開在電機系所以第一堂就先聽看看，後來是找老師幫我加簽，開始上課之後慢慢瞭解到這門課的運作方式就是助教講解然後老師補充說明，剩下的作業內

容就是我們自己回去完成，當初以為網站是只要簡單的一個前後端就好，但後來要求用到三個 Django 的專案，那時候在設定資料庫跟跨站的資料互通就花了很多時間，好像也是那時候蠻多同學就沒有再繼續修，可能是覺得作業量太大還有要花時間查資料，但我就是盡量利用空閒跟假日的時間來完成各個作業，雖然助教在上課時間不會講得太仔細，但還好它們都有附上參考資料和關鍵字，所以在自學的時候節省了不少時間，透過這門課我算是完整的建立起前後端還有資料庫，另外也有自己一套的 Linux 系統在跑我的程式，除了學習到資訊系統的知識之外還有金融相關的知識，這些金融知識雖然是最基本的那些，但也起到了拋磚引玉的效果，未來想更深入地去研究就可以從這些知識點切入，總結來說，修完這門課我覺得自己是收穫良多，相信未來在做研究或是工作時遇到類似的問題我也能夠用同樣的態度跟知識來解決問題。