

淡江大學資訊工程學系

上線無悔演算法於投資組合最佳化之應用

專題成果報告書

洪碩廷 410410111 資工 3B

魏品華 410410236 資工 3B

黃裕權 410410863 資工 3B

陳南天 410410749 資工 3B

藍翰翔 410410715 資工 3B

目錄

| | | |
|----|--|----|
| 壹、 | 動機與目的 | 2 |
| 一、 | 研究動機 | 2 |
| 二、 | 研究目的 | 2 |
| 貳、 | 相關研究 | 3 |
| 參、 | 內容描述 | 3 |
| 一、 | 使用技術 | 3 |
| 二、 | 使用套件 | 4 |
| 三、 | 專題架構 | 5 |
| 1. | 架構描述 | 6 |
| 2. | 第一部分 | 6 |
| A. | 模型訓練 | 6 |
| B. | 策略組合分析 | 10 |
| 3. | 第二部分 | 12 |
| A. | LSTM (Long Short-Term Memory) | 12 |
| B. | MWU 無悔演算法 (Multiplicative Weight Update algorithm) | 12 |
| C. | 對照組設計 | 16 |
| 4. | GUI 介面 | 18 |
| A. | 訓練介面 | 18 |
| B. | 分析介面 | 23 |
| 5. | 結果分析 | 25 |
| 肆、 | 總結 | 29 |
| 一、 | 結論 | 29 |
| 1. | 模型訓練成果 | 29 |
| 2. | 策略設計成效 | 30 |
| 3. | MWU 總體成效 | 30 |
| 二、 | 開發困難 | 32 |
| 三、 | 未來展望 | 33 |
| 四、 | SWOT 分析圖 | 34 |
| 五、 | 組員心得感想 | 36 |
| 伍、 | 參考文獻 | 37 |

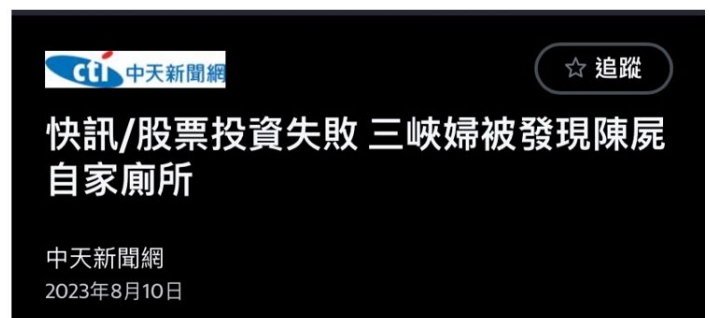
壹、動機與目的

一、研究動機

在當今的數位時代，股票投資已經成為許多人追逐財務自由的途徑之一。然而，許多人卻陷入了投資的高風險陷阱中，因為他們可能沒有足夠的知識或策略來應對市場的波動。這種情況在網路上尤其常見，因為社交媒體和網路論壇提供了許多投資建議和信息，但往往缺乏客觀性和專業性。

另一個常見的問題是缺乏明確的投資策略。許多人可能進入股市，但卻沒有確定的目標或計劃。他們可能隨時變換投資方向，沒有長期的規劃，這樣的操作往往是不穩定和高風險的。

因此我們製作這個專題希望能幫助人們更理性地進行股票投資，避免因為盲目跟風或沒有明確的投資策略而導致損失。



二、研究目的

在這次專題中，我們想要達到幾個目標，分別是：

1. 建構與優化 LSTM 的模型
2. 如何結合 LSTM 模型設計出一個有效策略
3. MWU 演算法實作
4. 實證分析無悔演算法的實作表現
5. 完善的使用者介面便於使用者操作

我們結合深度學習模型、演算法的應用、策略的設計，提供使用者科學、系統化的方法來分析股票市場，並給出合理的投資建議。搭配上無悔演算法，不僅能夠協助投資者穩定地長期投資，更能夠在當下找到最佳的投資組合。這種方法能夠大大降低風險和損失。

貳、 相關研究

AI 股票交易機器人

- TrendSpider

TrendSpider 是一款先進的股票分析軟體，利用其獨特的機器學習演算法，自動掃描歷史市場數據，識別趨勢並將其提供給交易者，從日內交易者到一般投資者皆可受惠。其交易機器人功能可將策略完全自動化，並通過滿足特定條件來觸發事件

- Tickeron

Tickeron 是一個先進的人工智能交易平台，提供廣泛的人工智能交易選項。其 AI 機器人實時掃描股票和 ETF，提供買賣交易的潛在利潤和止損建議。用戶可自定義交易列表，機器人根據實時模式尋找交易機會。

- Kavout

Kavout 是一個創新的人工智能投資平台，其核心機器"Kai"能夠分析海量數據、文件和股票報價，並整合新聞、和社交媒體信息，提供準確的投資視圖。附加功能包括模擬交易投資組合和市場分析工具，可篩選最佳股票並提供日曆功能跟蹤股票表現。

參、 內容描述

一、 使用技術

整個專題主要以 Python 實現，而會選擇 Python 的原因有以下幾個：

- 1) Python 就目前來說是一個非常流行的語言，其龐大的社區和資源為開發者提供了無限可能。在使用 Python 時，我們可以輕鬆獲得大量的支援和解決方案，無論是論壇、文章還是開源項目，都可以找到對各種問題的解答和建議。

| Aug 2022 | Aug 2021 | Change | Programming Language | Ratings | Change |
|----------|----------|--------|---|---------|--------|
| 1 | 2 | ▲ |  Python | 15.42% | +3.56% |
| 2 | 1 | ▼ |  C | 14.59% | +2.03% |
| 3 | 3 | |  Java | 12.40% | +1.96% |
| 4 | 4 | |  C++ | 10.17% | +2.81% |
| 5 | 5 | |  C# | 5.59% | +0.45% |

- 2) Python 擁有豐富的第三方函式庫和工具，這些函式庫和工具提供了豐富的功能，能夠極大地加速開發過程。其中 Pandas 函式庫提供了豐富的數據結構和數據操作功能，能夠讓我們很輕鬆的處理股票資料，如 CSV、Excel、SQL 數據庫等。而在可視化方面，Matplotlib 和 Seaborn 函式庫提供了強大的繪圖功能，能夠創建各種類型的靜態和動態圖表，如折線圖、散點圖、直方圖等，這些圖表可以幫助我們很好的分析股票趨勢及收益成效。
- 3) Python 在機器學習和人工智慧領域也廣受歡迎，擁有眾多強大的機器學習庫，如 TensorFlow、PyTorch、Scikit-learn 等，這些函式庫可以幫助我們實現複雜的深度學習模型，從而進行數據分析、預測和優化等任務。

二、 使用套件

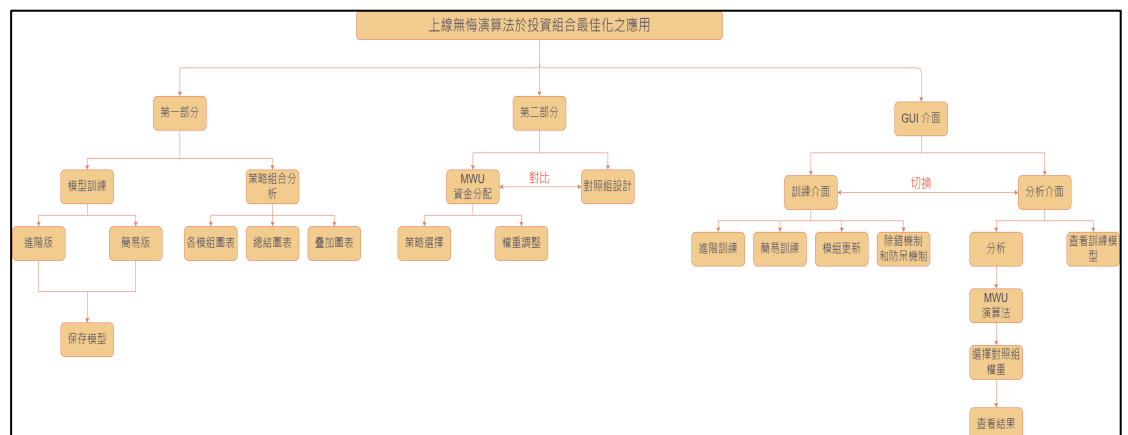
- ◆ Yfinance: 此套件用於從 Yahoo Finance 獲取金融數據。提供了一個簡單易用的方式，輕鬆獲取股票、指數、期貨、外匯等金融資產的歷史數據、即時價格和財務數據。而在此專題中，我們使用此套件幫助我們獲取歷年至今的股票資料。
- ◆ PyTorch: 它是一個開源的機器學習函式庫，主要用於深度學習研究和應用。能建構各種深度學習模型，而我們使用了此套件建立了 LSTM 模型。
- ◆ joblib: 用於提供輕量級的並行計算功能和對 Python 物件的序列化支持。而我們這邊是將它用來進行模型的保存及讀取。使用 `joblib.dump` 將 Python 物件序列化並保存到磁碟

上。並使用 `joblib.load` 從磁碟上加載保存的序列化物件。

- ◆ Subprocess: 用於在 Python 程序中啟動新的 process，並與這些進程進行通信，常見於執行外部命令、程序或腳本等。而此專題將它用來進行兩個使用者介面的切換。
- ◆ os: 用於執行各種操作系統相關的任務，比如獲取、更改當前工作目錄、創建、刪除、重命名文件或目錄、以及執行外部命令等。在此處，我們使用了 `os.path`、`os.remove`、`os.listdir` 來進行模型的控管。
- ◆ Tkinter: 為一個 GUI 工具包，用於創建桌面應用程序的用戶界面。它基於 Tcl/Tk，是一個跨平台的工具包。提供了一個簡單而直觀的界面，用於創建各種 GUI 元素，如窗口、按鈕、標籤等。我們的整個介面包含美化的部分皆是以他所構建出來的。



三、 專題架構



(專題架構圖)

1. 架構描述

◆ 第一部分

以股票資料訓練出 LSTM 深度學習模型，用於預測股票收盤價，最終形成一個策略的組合。並透過不斷優化模型，例如加入技術指標、實施 walk forward 策略等，以提高預測準確度和穩定性。

◆ 第二部分

我們將先利用第一部分的結果，設計出一個有效的策略，用以決定 MWU 公式中的損失函數。接著，將以數學理論為基礎的 MWU 演算法用程式碼的方式實現。最後，MWU 演算法將為使用者提供最佳的資金分配方案，使其能夠在不同時間點選擇最佳的投資策略組合，以最大化收益並降低風險。

◆ GUI 介面

提供一個完善的 GUI 介面，讓使用者能夠輕鬆操作和分析投資策略組合。這個介面將設計許多偵錯及防呆的機制，以避免在分析過程中出現不必要的錯誤，同時也能讓使用者迅速上手。透過這個 GUI 介面，投資者也能夠藉由各式圖表的比較，來統整並分析出結果。

2. 第一部分

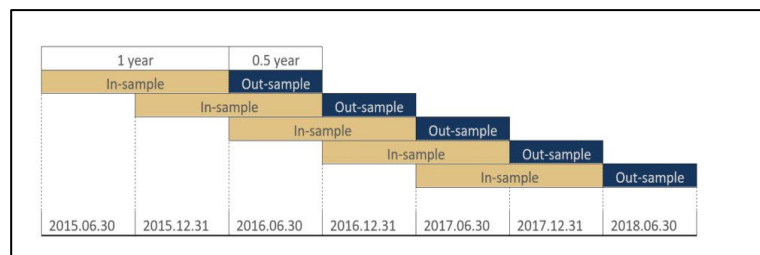
A. 模型訓練

(1) 資料預處理及優化

由使用者輸入多支美股代號，並開始逐一訓練。首先到 Yahoo Finance 下載從 2009 到至今的資料。加入技術指標的特徵欄位，例如：MACD、K、WILLR 可以為模型提供更多有價值的信息，從而提高模型的預測性能。接著進行資料標準化，目的是將不同特徵縮放到相同範圍的過程，消除尺度差異，提高模型收斂速度，避免特徵偏向，並提高模

型穩定性。

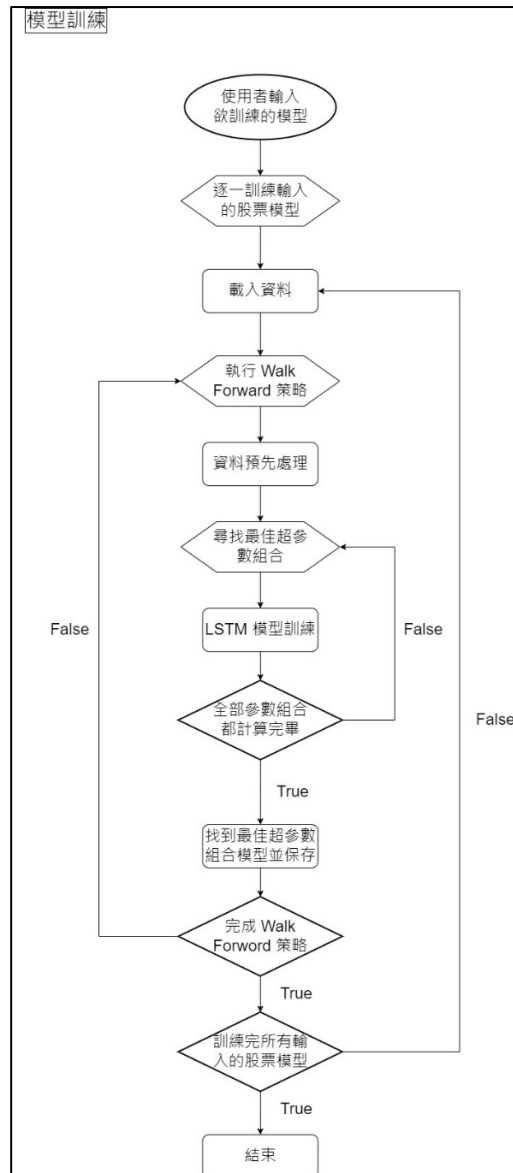
而為了確保模型的穩健性和泛化能力，並在發生突發事件時讓後續分析不受影響，我們並非直接將整個資料及丟入訓練出唯一的模型，我們採取的是 Walk Forward 的策略。而為了實施 Walk Forward 策略，我們會將資料集切分成多個子集，並逐步訓練多個模型。簡單來說，Walk Forward 是一種交叉驗證方法，每次僅使用部分資料來訓練模型，然後將資料逐步向前移動，進而訓練下一個模型。採用這樣的訓練方式能夠評估模型在不同時間段的表現，以確保用此方法訓練模型的可靠性。而就此專題而言，我們將資料分成連續的 5 年資料來訓練模型(test_train)，然後每次使用半年的資料來測試模型(test_data)，並逐步向前直到至今。



(Walk Forward 示意圖)

最後，每個模型內部的訓練則是使用 10 天的資料作為輸入去預測一天的結果。這樣的訓練方式不只能避免單一事件所帶來的影響，同時還能幫助模型更好地考慮到資料間的相關性和時間序列的特性，這對於股票這類型和時間序列相的問題來說是非常重要的，能有效提高模型的預測準確度。因此我們會將資料再進行二次的切分。

(2) 模型建立及訓練



(模型訓練架構圖)

使用 PyTorch 框架來定義和建立 LSTM 模型。

在開始訓練前我們會需要進行超參數例如 hidden_dim、num_layers 和 num_epochs 等的挑選。這部分我們設計了三種不同的訓練模式：

甲、進階版訓練：

該訓練方式提供多組超參數進行訓練，對於每組超參數組合，系統將逐一訓練，最終選

擇 RMSE 最低的模型作為最優模型。使用者可以期待更高的準確性和更好的性能，但可能需要花費更多時間。

乙、簡易版訓練：

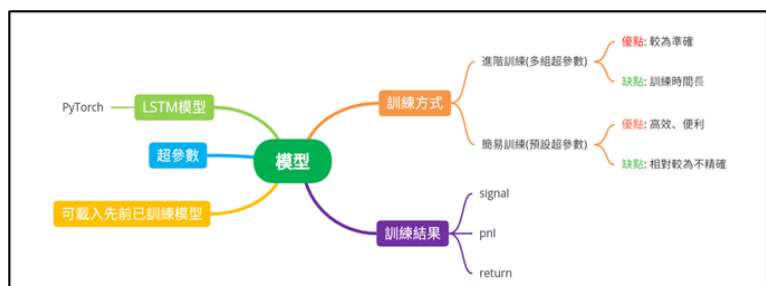
用戶可以快速獲得一個模型，無需進行複雜的超參數選擇，同時能夠滿足基本的需求。特別適用於那些對於模型性能要求不高，或是需要快速訓練的用戶。

丙、模型的更新：

根據資料的時間框架，我們每半年會需要加入一個新的模型。而直接更新的用意是能夠只針對新的模型進行訓練，無需重新訓練全部的模型。

會這樣設計是希望讓使用者能夠根據自身需求靈活選擇訓練方式，同時在不需要進行超參數調整的情況下，仍能獲得一個具有一定效能的模型。

訓練完的模型將會以 .gz 檔的格式被保存到資料夾中，以便日後直接使用無需重新訓練。透過這種方式，用戶可以隨時載入之前訓練好的模型來進行後續的分析，從而節省時間和資源。進階訓練和簡易訓練將以不同的檔名進行區別，以便後續進行一些處理像是模型升級、刪除、分析等動作。最後，根據訓練的結果，能計算出 signal、pnl、return 等相關交易資訊。



B. 策略組合分析

一樣由使用者輸入多支欲分析的股票代號，形成一個投資策略組合。這部分輸入的股票一定要是訓練過的。接著，生成各個股票訓練後的圖表，以便使用者進行觀察，並在後續實行完畢 MWU 演算法後將其結果與各圖表進行比較，進而顯示出 MWU 所帶來的成效。

1. 各模組圖表：



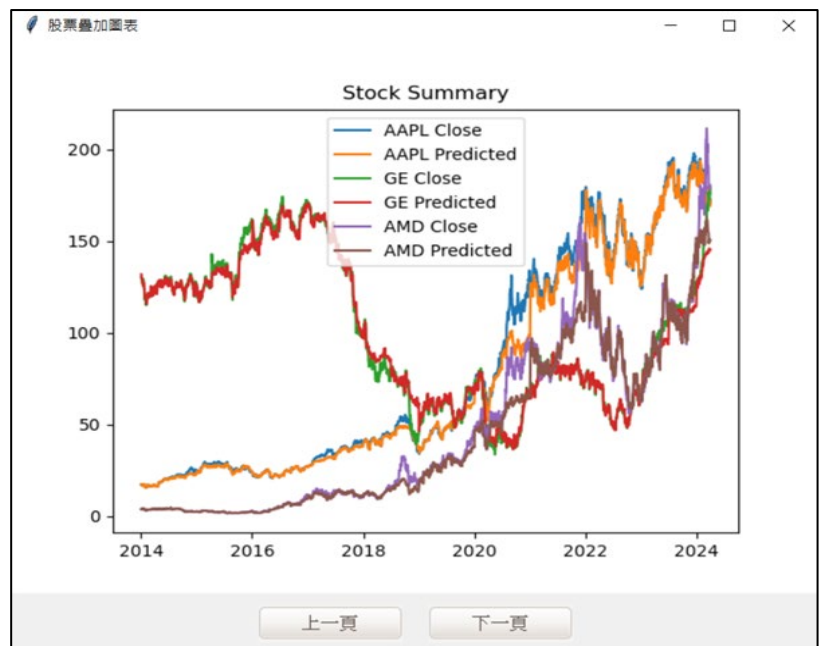
此類型圖表展示出使用 Walk Forward 策略後每個模型分別訓練的結果，能看出細部的預測結果並和實際的收盤價進行對比。由於模組數量不少，因此提供下拉選單方便能快速切換至各支股票。

2. 總結圖表：



此類型圖表將單一股票所有使用 Walk Forward 訓練完的模型結果合併呈現總體結果。將每個模型結合能幫助使用者快速的分析出單支股票的整體走勢，並凸顯出使用 Walk Forward 策略後的訓練成效。

3. 顯示疊加圖表：



此圖表再將所有選擇策略的訓練結果呈現到一張圖表上。用意是方便使用者比較各股票的走勢，並在後續 MWU 演算法實行完畢後能很好的解釋 MWU 所被證實的特性。

3. 第二部分

A. LSTM (Long Short-Term Memory)

LSTM 為深度學習中的一種模型，可以將它視為循環神經網路 (RNN) 的進階版。

而為何會選擇 LSTM 主要的原因有幾個：

1. 股票數據通常和時間序列相關，也就是說過去的價格和交易量等信息對未來的價格走勢具有影響。因此相對於 DNN 這類型的模型，LSTM 能夠捕捉到時間上的依賴關係(輸出會做為下一個時間的輸入)，並且在預測未來時考慮到過去的數據。
2. LSTM 相對於 RNN 雖然都能解決時間序列的問題，但是 RNN 的記憶能力較為有限，而 LSTM 能透過遺忘門的機制更有效捕捉序列中的長期依賴關係(藉由選擇忘記不重要的訊息，不會全部記住)。

B. MWU 無悔演算法 (Multiplicative Weight Update algorithm)

透過前述的訓練和分析，我們形成了一組策略，每個策略對應著一支股票的買賣行為。然而，如何分配每個策略的資金會是我們第二部分使用的 MWU 演算法所涉及的。

1. 什麼是 MWU ?

使用 MWU 的主要目的是幫助我們分配每個策略的資金

比例。它已被證明當後見之明條件下為一種固定投資組合(對照組)，短期來看或許不會比較好，但時間趨近無窮時，其表現與後見之明一樣出色，因此被稱為“無悔演算法”。那這也是後續為何我們會將對照組設計為一個固定投資組合的原因。

透過策略計算每日損益並調整權重。每輪中，表現不佳的決策的權重會降低，以達到避險的效果。

2. MWU 的成效?

- 風險控制：

MWU 演算法能夠有效控制投資組合的風險，通過調整各股票的權重來實現風險的分散，從而降低整個投資組合的波動性。

- 收益最佳化：

MWU 演算法也能夠尋找到當下最優的投資組合比例，以最大化預期收益。通過長時間的調整策略權重，能夠在多個選擇的策略中，有效地追蹤和捕捉收益機會。

- 動態適應：

MWU 演算法具有動態適應的特性，能夠快速反應市場的變化。當市場情況發生變化時，MWU 算法能夠及時調整投資組合，以應對新的市場趨勢和信息。

3. 如何實現 MWU?

Multiplicative Weight Update (MWU)

- Maintain a vector of weights $\mathbf{w}_t = (\mathbf{w}_t(1), \dots, \mathbf{w}_t(n))$ where $\mathbf{w}_1 := (1, 1, \dots, 1)$.
- Update the weights at time t by
 - $\mathbf{w}_t(i) := \mathbf{w}_{t-1}(i) \cdot e^{-\beta \ell_{t-1}(i)}$.
 - $\mathbf{x}_t := \frac{\mathbf{w}_t(i)}{\sum_{j=1}^n \mathbf{w}_t(j)}$.

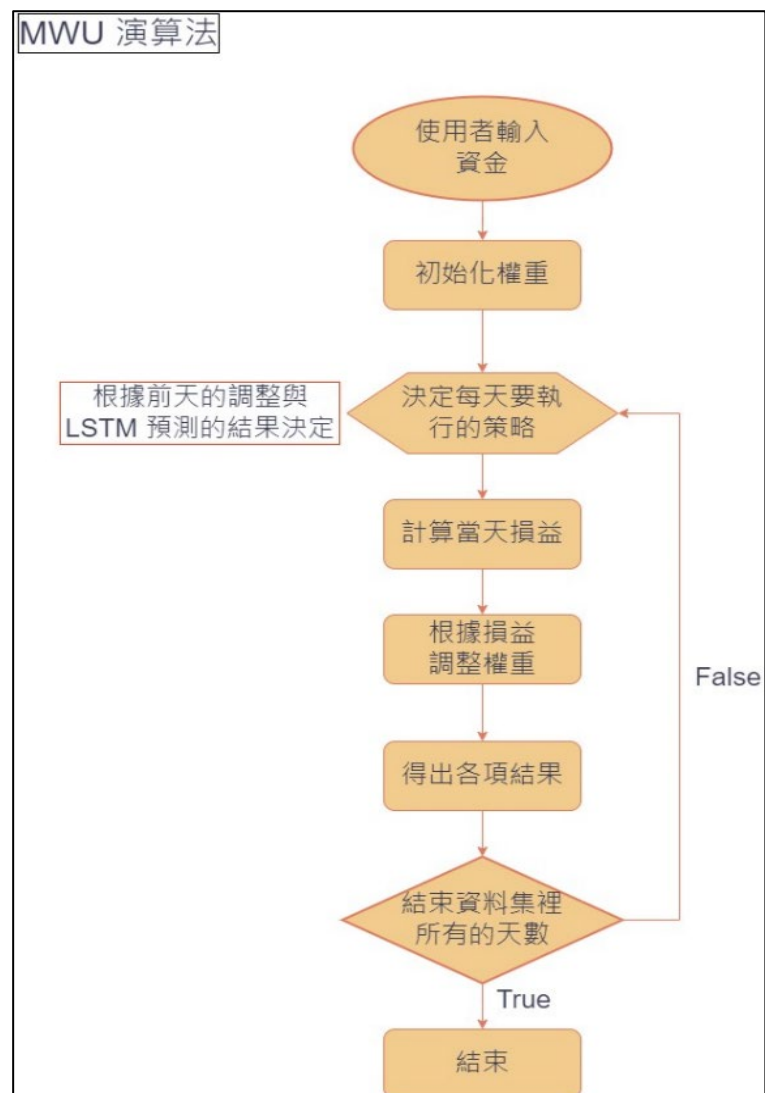
(MWU 數學理論)

STEP 1:

將資金根據策略的數量進行平均分配，以初始化權重。

STEP 2:

根據我們自行設計出的策略計算出每日各策略 MWU 公式中的損益函數，並根據損益函數及理論公式來進行權重的調整得出每日的最佳投資權重及各項結果(EX:持有張數、當下權重、分配資金、損益……)。



(MWU 演算法流程圖)

4. MWU 中的策略(損失函數)是如何設計的?

策略的部分，經過不斷嘗試不同策略後，最終我們的設計方式是這樣的：

- 當某策略今天收盤價上升(↑)並且 LSTM 模型顯示明天的預測收盤價也上升(↑)，此時 MWU 的損失函數會是正的，也就是提升此策略的分配比例。
- 當某策略今天的收盤價下降(↓)並且 LSTM 模型顯示明天的預測收盤價也下降(↓)，此時 MWU 的損失函數會是負的，也就是降低此策略的分配比例。
- 而若以上條件皆不成立，當天將不給予此策略損失函數來進行調整。

此策略的設計不只是結合了前面模型的結果，同時從結果層面來說，將它做為 MWU 損失函數的決策方式是非常有效的。

5. MWU 的結果：

經由每天的調整，我們能透過上方滑動按鈕或直接輸入天數找到各時間段最佳投資策略組合及各項資訊。



(MWU 成果圖)

結果圖能給使用者的資訊包括：

(1). Portfolio Value:

這代表投資者持有的全部投資資產的總價值。
換句話說，它是投資者所有資產的總值，也就是我們當下的總資產。

(2). 每個策略的權重、相對昨天資金和權重的變化：

權重代表者當天所建議分配的資金比例。資金變化則代表當天資金需交易的量，並以顏色標註相對昨天的變化：紅色表示上升(↑)，綠色表示下降(↓)，灰色表示變化不大(-)。

(3). Max Drawdown:

指的是投資組合或資產在一段特定時間內從峰值到谷底的最大損失幅度，它能夠提供投資者對投資組合風險和可能損失的重要洞察，有助於我們用來解釋 MWU 中避險的效果。

(4.) Sharpe Ratio:

用來衡量投資組合的風險調整後報酬的指標，夏普比率越高，表明投資組合在單位風險下獲得的超額報酬越高(比率超過 1 被認為是不錯的，超過 2 則被視為非常好)。我們可以透過他來快速得知投資成果的好壞，有助於我們用來解釋 MWU 中無悔的特性。

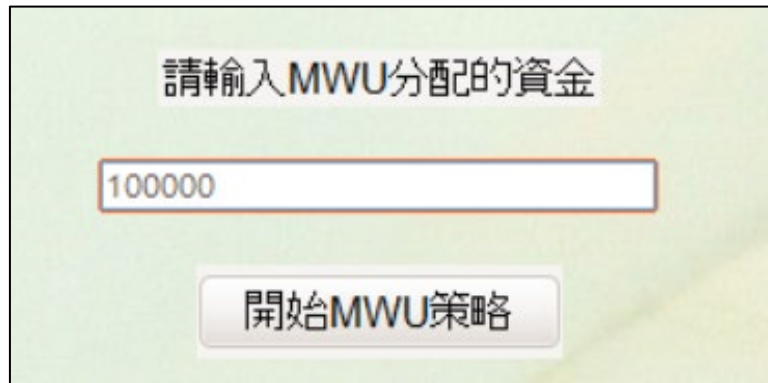
C. 對照組設計

就如前面 MWU 所說的「當後見之明條件下為一種固定投資組合，短期來看或許不會比較好，但時間趨近無窮時，其表現與後見之明一樣出色」。因此對照組的設計旨在提供一種固定的投資方式，作為比較基準，讓使用者能夠清楚地評估 MWU 演算法的效果。透過對照組的設計，我們可以比較 MWU 與固定投資方式之間的表現差異。而使用

者選擇權重的方式如下：

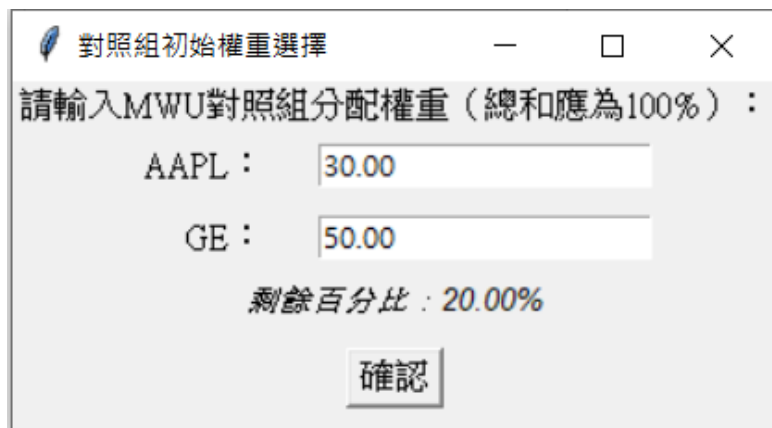
STEP 1:

進行完投資策略組合的分析後，輸入欲投資的總資金。



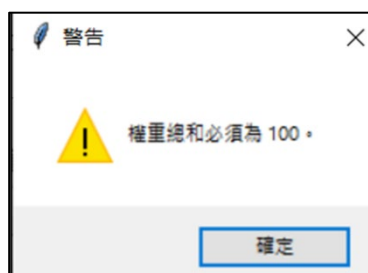
STEP 2:

啟動 MWU 策略後，會彈出一個視窗，讓使用者選擇對照組的固定權重分配。



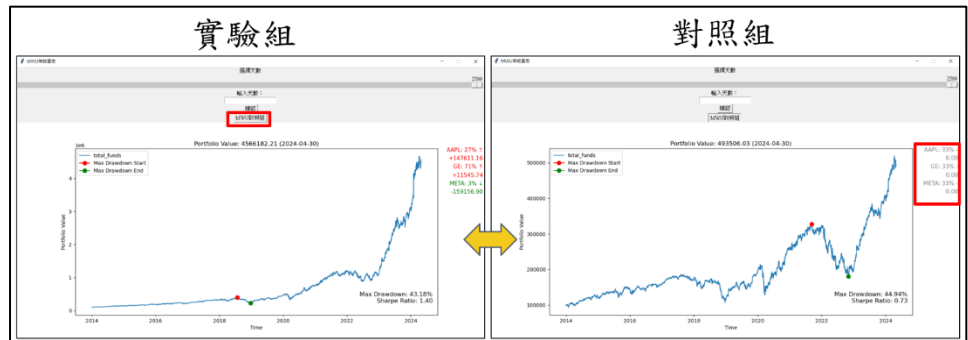
STEP 3:

在選擇對照組權重時，系統會提供剩餘百分比，確保權重總和為 100%。



STEP 4:

對照圖範例顯示。



4. GUI 介面

A. 訓練介面



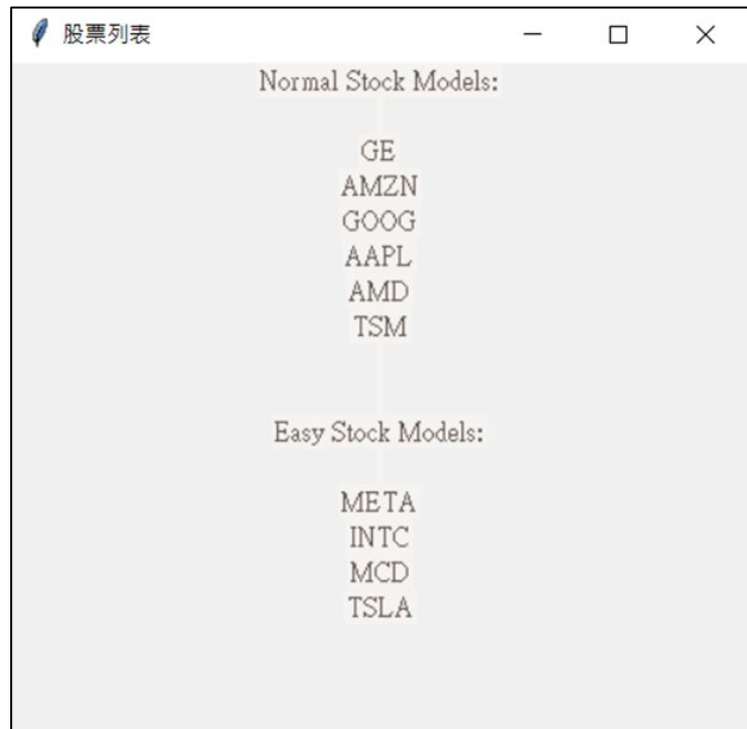
(股票訓練應用程式)

1) 功能介紹:

- 訓練股票(進階版)：用以輸入完畢欲訓練股票後進行進階訓練。
- 訓練股票(簡易版)：用以輸入完畢欲訓練股票後進行簡易訓練。
- 模型更新：用以輸入完畢欲更新股票後進行更

新。

- 顯示以訓練的股票：用來顯示使用者已訓練過的股票模型。上半部分顯示擁有進階版模型的股票，而下半部分則顯示擁有簡易版模型的股票。



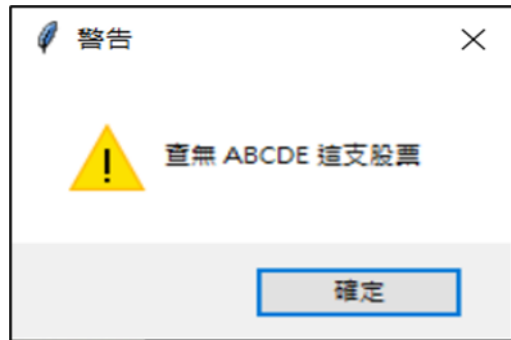
(顯示以訓練的模型)

- 切換到分析 APP：將兩個介面都轉成.exe 的執行檔，讓使用者透過此按鈕快速切換至分析 APP。
- 退出：立即退出訓練介面
- Switch to English version: 提供了英文版的介面，以便不同語言使用者的使用需求。

2) 全方面的錯誤偵測及防呆：

而訓練模型的過程，會讓使用者知道訓練的進度，以及是否發生一些錯誤，若有錯誤會顯示發生了多少個錯誤以便使用者檢查問題。

- 當使用者輸入未知股票代號時



- 輸入的股票還未存在任何模型。



- 已存在進階模型，而使用者又點擊了進階訓練。

| 訓練結果 | |
|--------------------|------|
| 開始訓練，過程請勿中斷 | |
| 2009.0~2014.5_AAPL | 已經存在 |
| 2009.5~2015.0_AAPL | 已經存在 |
| 2010.0~2015.5_AAPL | 已經存在 |
| 2010.5~2016.0_AAPL | 已經存在 |
| 2011.0~2016.5_AAPL | 已經存在 |
| 2011.5~2017.0_AAPL | 已經存在 |
| 2012.0~2017.5_AAPL | 已經存在 |
| 2012.5~2018.0_AAPL | 已經存在 |
| 2013.0~2018.5_AAPL | 已經存在 |
| 2013.5~2019.0_AAPL | 已經存在 |
| 2014.0~2019.5_AAPL | 已經存在 |
| 2014.5~2020.0_AAPL | 已經存在 |
| 2015.0~2020.5_AAPL | 已經存在 |
| 2015.5~2021.0_AAPL | 已經存在 |
| 2016.0~2021.5_AAPL | 已經存在 |
| 2016.5~2022.0_AAPL | 已經存在 |
| 2017.0~2022.5_AAPL | 已經存在 |
| 2017.5~2023.0_AAPL | 已經存在 |
| 2018.0~2023.5_AAPL | 已經存在 |
| 2018.5~2024.0_AAPL | 已經存在 |
| 2019.0~2024.5_AAPL | 已經存在 |
| AAPL 訓練成功 | |

- 已存在進階或簡易版模型，而使用者又點擊了簡易版的訓練。

| 訓練結果 | |
|--------------------|---------------|
| 開始訓練，過程請勿中斷 | |
| 2009.0~2014.5_INTC | 已經存在或擁有更完善的模組 |
| 2009.5~2015.0_INTC | 已經存在或擁有更完善的模組 |
| 2010.0~2015.5_INTC | 已經存在或擁有更完善的模組 |
| 2010.5~2016.0_INTC | 已經存在或擁有更完善的模組 |
| 2011.0~2016.5_INTC | 已經存在或擁有更完善的模組 |
| 2011.5~2017.0_INTC | 已經存在或擁有更完善的模組 |
| 2012.0~2017.5_INTC | 已經存在或擁有更完善的模組 |
| 2012.5~2018.0_INTC | 已經存在或擁有更完善的模組 |
| 2013.0~2018.5_INTC | 已經存在或擁有更完善的模組 |
| 2013.5~2019.0_INTC | 已經存在或擁有更完善的模組 |
| 2014.0~2019.5_INTC | 已經存在或擁有更完善的模組 |
| 2014.5~2020.0_INTC | 已經存在或擁有更完善的模組 |
| 2015.0~2020.5_INTC | 已經存在或擁有更完善的模組 |
| 2015.5~2021.0_INTC | 已經存在或擁有更完善的模組 |
| 2016.0~2021.5_INTC | 已經存在或擁有更完善的模組 |
| 2016.5~2022.0_INTC | 已經存在或擁有更完善的模組 |
| 2017.0~2022.5_INTC | 已經存在或擁有更完善的模組 |
| 2017.5~2023.0_INTC | 已經存在或擁有更完善的模組 |
| 2018.0~2023.5_INTC | 已經存在或擁有更完善的模組 |
| 2018.5~2024.0_INTC | 已經存在或擁有更完善的模組 |
| 2019.0~2024.5_INTC | 已經存在或擁有更完善的模組 |
| INTC 訓練成功 | |

- 訓練時發現資料夾中參雜了進階和簡易版的模型檔案。

```

訓練結果
開始訓練，過程請勿中斷
2009.0~2014.5_TSLA 已經存在或擁有更完善的模組
2009.5~2015.0_TSLA 已經存在或擁有更完善的模組
2010.0~2015.5_TSLA 已經存在或擁有更完善的模組
2010.5~2016.0_TSLA 已經存在或擁有更完善的模組
2011.0~2016.5_TSLA 已經存在或擁有更完善的模組
2011.5~2017.0_TSLA 已經存在或擁有更完善的模組
2012.0~2017.5_TSLA 已經存在或擁有更完善的模組
2012.5~2018.0_TSLA 已經存在或擁有更完善的模組
2013.0~2018.5_TSLA 已經存在或擁有更完善的模組
2013.5~2019.0_TSLA 已經存在或擁有更完善的模組
2014.0~2019.5_TSLA 已經存在或擁有更完善的模組
2014.5~2020.0_TSLA 已經存在或擁有更完善的模組
2015.0~2020.5_TSLA 已經存在或擁有更完善的模組
2015.5~2021.0_TSLA 雖然不存在，但須進行一般訓練
2016.0~2021.5_TSLA 雖然不存在，但須進行一般訓練
2016.5~2022.0_TSLA 雖然不存在，但須進行一般訓練
2017.0~2022.5_TSLA 雖然不存在，但須進行一般訓練
2017.5~2023.0_TSLA 雖然不存在，但須進行一般訓練
2018.0~2023.5_TSLA 雖然不存在，但須進行一般訓練
2018.5~2024.0_TSLA 雖然不存在，但須進行一般訓練
2019.0~2024.5_TSLA 雖然不存在，但須進行一般訓練
發生了8個錯誤，請檢查

```

- 當檢測到已擁有最新的模型時，而使用者又點擊了模型更新

```

訓練結果
開始訓練，過程請勿中斷
2019.0~2024.5_AAPL 模組無需更新
AAPL 訓練成功

```

- 在尚未訓練模型的情況下，使用者點擊了模型更新

```

訓練結果
開始訓練，過程請勿中斷
2019.0~2024.5_GH 模組尚未訓練或損毀，請重新訓練
發生了1個錯誤，請檢查

```


設立以上機制的目的為：

- ◆ 確保剛開始使用的使用者進行正確的操作。
- ◆ 防止發生模型的重複保存。
- ◆ 避免進行不必要的訓練。
- ◆ 確保後續不會因為模型錯亂導致分析出現問題。

B. 分析介面



(股票訓練應用程式介面1)

(股票訓練應用程式介面2)

1) 功能介紹：

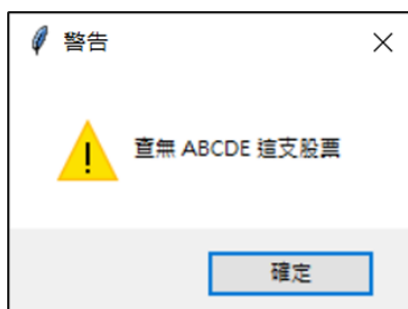
- ◆ 分析股票：用以輸入完畢欲分析之策略組合後讀取資料夾模組並產生各項結果及後續 MWU 的操作介面(股票訓練應用程式介面 2)。
- ◆ 顯示以訓練的股票：如同訓練介面，此處能幫助使用者知道當下有哪些股票模型是能夠分析的。
- ◆ 切換到訓練 APP：將兩個介面都轉成.exe 的執行檔，讓使用者透過此按鈕快速切換至訓練 APP。
- ◆ 退出：立即退出分析介面。
- ◆ 開始 MWU 策略：使用者輸入欲投入資金後點擊即可開始 MWU 演算法。
- ◆ 顯示各模組圖表、總結圖表、疊加圖表：呈現出輸入投資組合的各個輸出圖表
- ◆ 重置：每結束一次分析之後必須按下重置才能進行下一組的分析

2) 全方面的錯誤偵測及防呆：

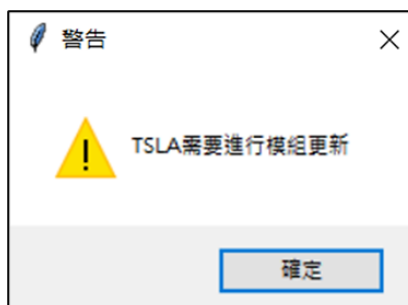
- 當用戶輸入尚未訓練過的股票代號並按下分析時，系統將顯示提示訊息，指出尚未進行過分析或該股票的模型遺失，並建議用戶重新訓練模型。



- 當用戶輸入未知的股票代號並按下分析時，系統將顯示提示訊息，指出查無此支股票。

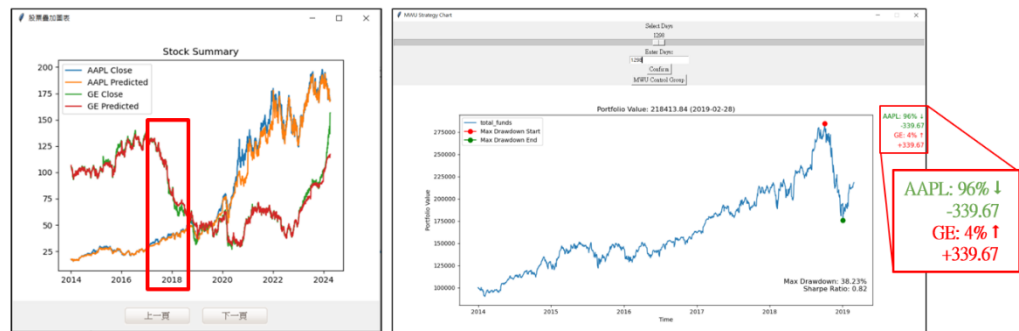


- 當用戶輸入的股票缺少最新的模型時，系統將顯示提示訊息，指出需要進行模型更新。



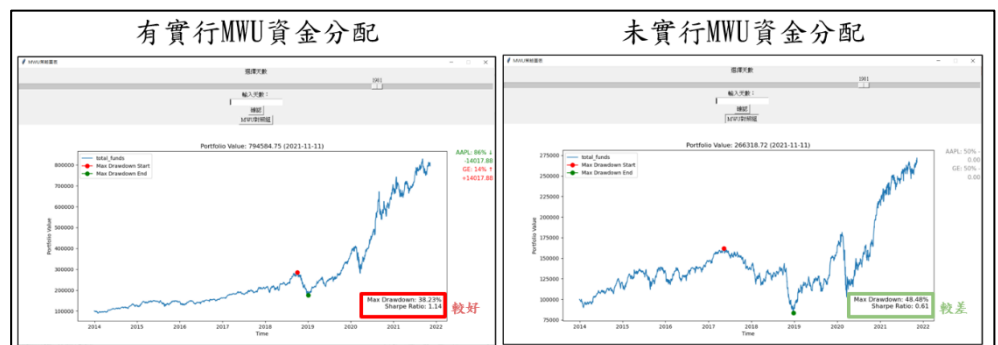
5. 結果分析

1) 從資金分配的角度分析(AAPL + GE)



由左邊的策略疊加圖表能夠看出 2017~2019 這個時間段，GE 股票不斷的在下跌，因此右圖的 MWU 結果圖在這個時間段將大部分的權重都集中在 AAPL 這支股票身上，有符合 MWU 避險的效果。

2) 從 Portfolio Value 的角度(AAPL + GE)



■ (2021/11/11 該天資料)有實行 MWU 資金分配

Max Drawdown: 38.20%

Sharpe Ratio: 1.14

■ (2021/11/11 該天資料)未實行 MWU 資金分配

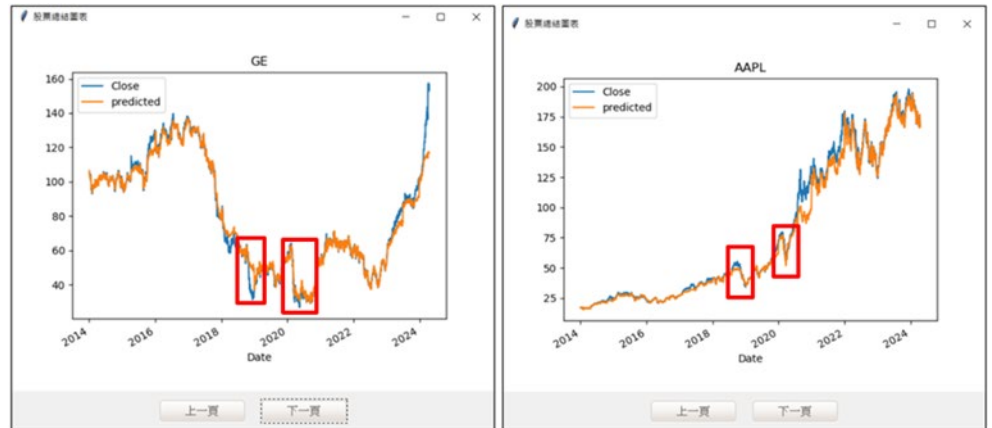
Max Drawdown: 48.48%

Sharpe Ratio: 0.61

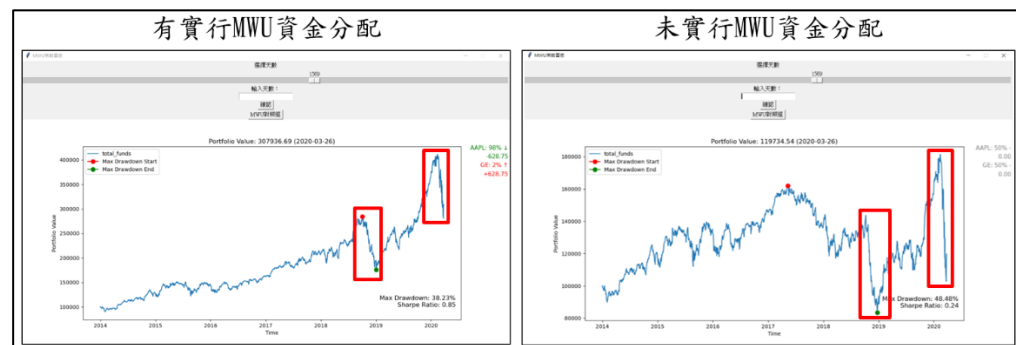
能看出未實行 MWU 資金分配時，也就是對照組，不管是 Sharpe ratio 或是 Max Drawdown 都是較差的。

這邊也能明顯看出 MWU 在長期投資下的成效，也就是無悔的部分。

3) 從 Portfolio Value 下降時進行分析(AAPL + GE):



由這兩張圖可發現兩隻股票在 2018 及 2020 這兩個時間段都呈現下跌的狀態。



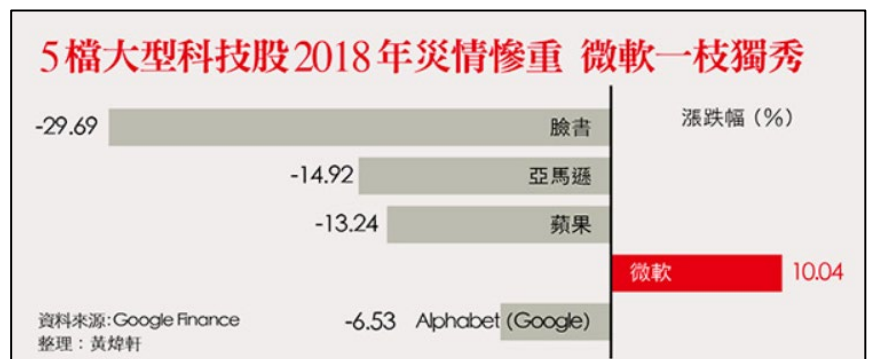
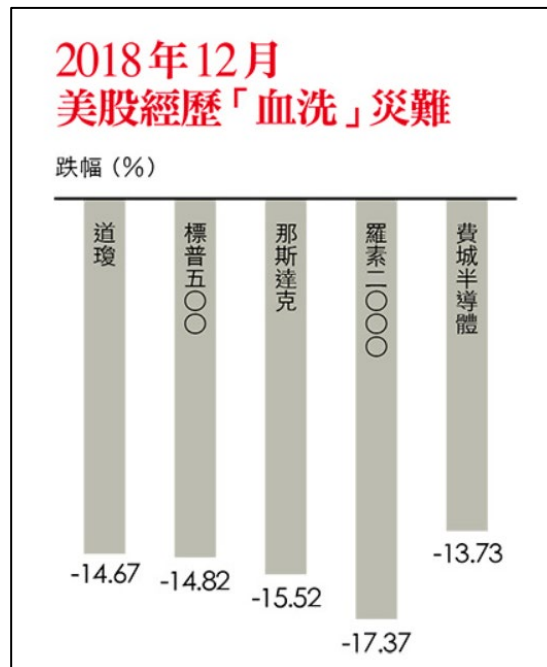
我們擷取出了 2018 及 2020 這兩個時間段總資產嚴重下降的情況，並發現此時不管 MWU 如何分配資金比例，Portfolio Value 都會有所下跌，但從左右兩張圖就可以明顯看出雖然都是損失，使用 MWU 時能有效降低損失的幅度。

4) 為何大多股票在這些時間段皆呈現下跌的趨勢?

我們更進一步的研究為何在這些時間段大多股票皆呈現下跌的趨勢，我們整理出了幾種原因：

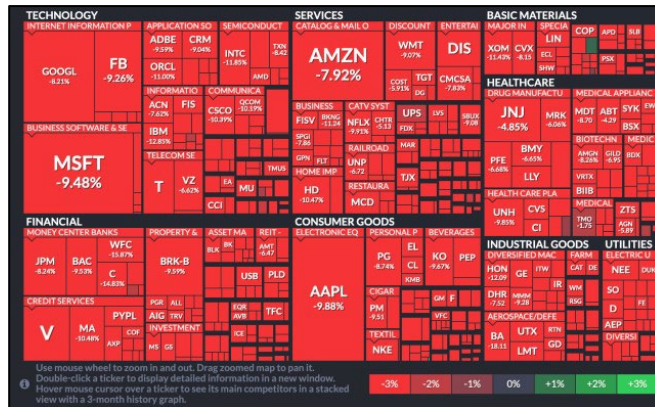
■ 中美貿易戰：

2018 年是中美貿易戰的開始，雙方互相加徵關稅，這導致了投資者對全球貿易環境的不確定性和擔憂，進而影響了大型企業的股票表現。



■ COVID-19 疫情爆發：

COVID-19 疫情在 2020 年初在全球蔓延，導致許多國家實施封鎖和限制措施，以控制病毒的傳播。這對全球經濟造成了重大影響，企業關閉或減產，消費活動大幅下降，導致全球供應鏈中斷，進而影響了許多公司的業績和股票價格。



| 美股三大股災 | | |
|--------|------|--|
| 美股股災 | 下跌幅度 | 原因 |
| 2000 年 | 73% | <ul style="list-style-type: none"> 股價急遽上漲，但營收跟不上 受到聯準會升息影響 |
| 2008 年 | 52% | 面對次貸的連環爆，這讓投資人對市場信心嚴重不足引發拋售 |
| 2020 年 | 35% | 投資人預測疫情將會嚴重影響經濟情況，於是發生不理性的拋售事件 |

由以上結果，我們可以得出分析後的結論：

即使擁有深度學習、演算法和人工智慧等先進技術的幫助，面對突發事件或災難仍然可能難以完全預測和控制。這些事件可能對整個經濟和市場產生深遠的影響，超出了單一技術或模型的能力範圍。而此專題的用意就是在於幫助投資者有效的降低損失，選擇一個最佳的投資組合。

肆、總結

一、結論

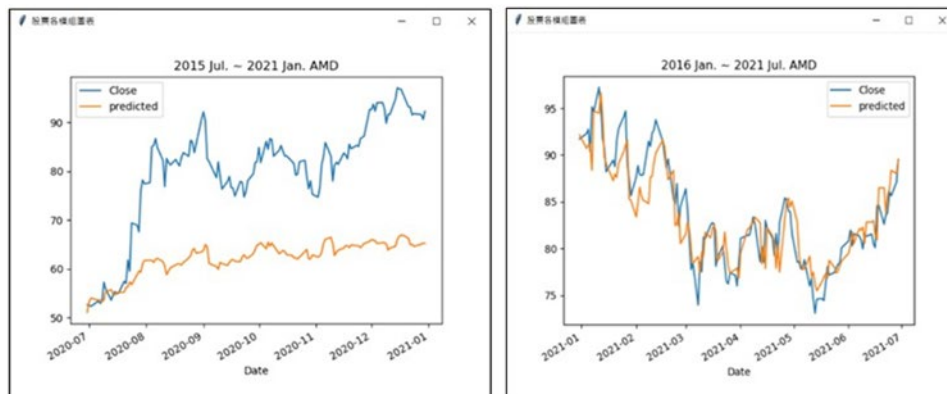
1. 模型訓練成果

整體訓練的結果再進行多項優化後的確上升了不少。



(predicted 與 close 相近)

但在遭遇到突發事件的年份例如新冠疫情時，模型整體的準確率會發生大幅下降的情況(下方左圖)。但由於我們使用了Walk Forward 策略，能避免在突發事件後的模型受到影響，準確率也跟著下降(右圖)。



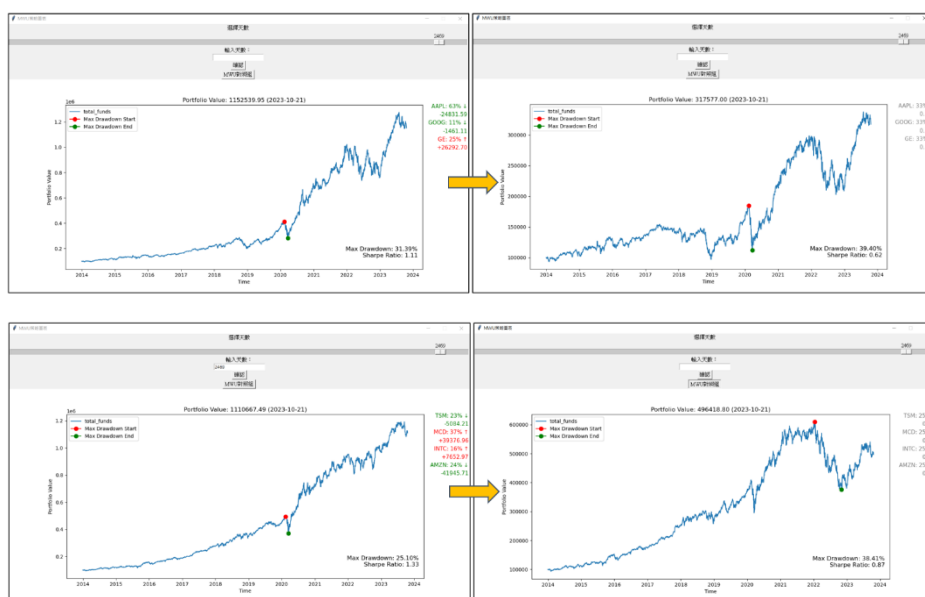
(右圖未受到左圖突發事件的影響)

2. 策略設計成效

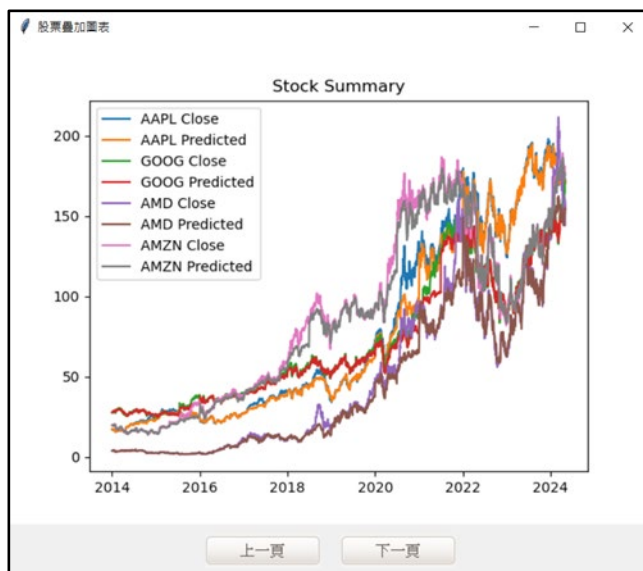
即使 LSTM 模型的預測結果在數值上與真實情況存在差異，但其對走勢的準確性使其成為 MWU 策略設計的重要參考依據。藉由結合 LSTM 模型的預測結果，MWU 策略得以更好地適應市場變化，並提高了整體投資組合的效能。因此，此策略的設計方式確實能有效的抉擇 MWU 公式中的損失函數。

3. MWU 總體成效

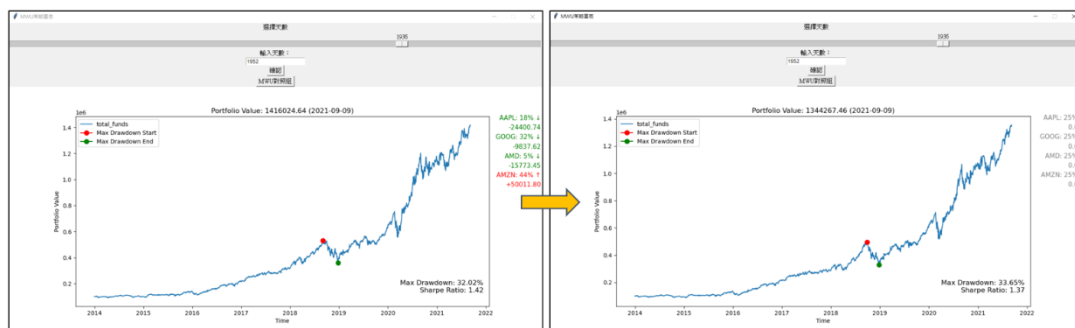
經過我們的測試，絕大多數存在較差策略的投資組合在經過長時間的調整過後使用 MWU 的結果都會比對照組的結果來的優秀，證實了 MWU 所被證明出無悔的特性。



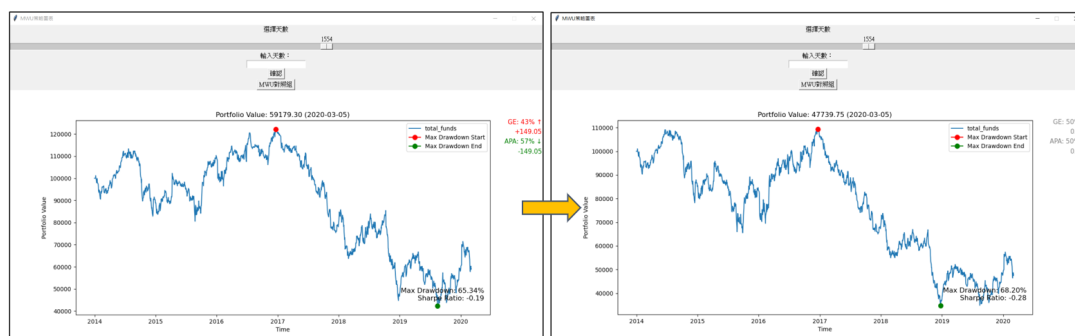
而 MWU 主要是能幫我們進行避險，所以當所有選擇的策略走勢是相近的，此時 MWU 所帶來的成效就會不明顯。舉例說明，當我們選擇 AAPL + GOOG + AMD + AMZN 皆為上漲的策略組合，從下圖可知四個策略的走勢非常相似。



可以由下圖看出，此時無法很好突顯出 MWU 的成效。



當然，若選擇到皆為下跌的投資組合，還是會發生賠錢的情況，不過和前面所說的一樣，MWU 能讓我們賠的相對較少。



二、開發困難

1) 資料的處理

在開發過程中，資料處理成為一個具有挑戰性的任務。特別是在切分 Walk Forward 策略和 `input_size` 時，需要對資料進行準確的切分和調整。此外，我們還需要在訓練完畢後加入一些相關欄位，例如 `position`。然而，我們在處理資料時遇到了一些困難，例如在調整特徵欄位時出現錯誤，導致使用了未來的資訊來進行分析，這對於模型的正確性產生了負面影響。後來經過仔細檢查和修正，我們成功解決了這個問題，確保了資料的準確性和一致性。

2) 策略的設計

此部份我們經過多次的修改並且在開會時詢問指導教授意見，最終選擇一個最適合的方式當作最終的結果。以下舉例出兩個原先的策略設計：

I. 根據股票張數變化(權重) + `signal`(買賣信號)欄位：

-> 是一個可行的方法，但我們發現使用到股票張數變化會導致剛開始表現優秀的策略一直維持著高權重的資金比例，無法實現出 MWU 的效果。

II. 直接根據 LSTM 預測的結果：

-> 會變成每天都進行調整的動作，導致權重變化太快，並且沒有我們現有策略的確保機制，導致若當下模型不夠準確，會發生決策分配失敗的情況。

3) MWU 的實現

MWU 演算法是一個基於數學理論的證明，但相關資訊在網路上並不容易獲得。我們必須透過與指導教授的交流、閱讀英文文獻等方式，逐步理解 MWU 演算法的核心概念，並思考如何將其應用於專題研究中。理解 MWU 演算法後，需要努力將其理論轉化為可執行的程式碼，並進行必要的調整以適應特定的應用場景，這一過程需要耗費大量的時間和精力。

Lecture 16

The Multiplicative Weights Algorithm*

In the next couple lectures, we will devise modular, iterative algorithms for solving LPs and SDPs. "Multiplicative weights" is a retronym for the simple iterative rule underlying these algorithms; it is known by different names in the various fields where it was (re)discovered. Check out the survey by Arora, Hazan and Kale [AHK05]; our discussion will be based on their treatment. Due to its broad appeal, we will consider multiplicative weights in more generality than we need for solving LPs and SDPs. In this lecture, we'll introduce some strategies for playing a prediction game. We'll tweak the game to suit our optimization needs. Finally, we'll play the tweaked game with a strategy called Hedge.

16.1 Warmup: prediction with expert advice

The following sequential game is played between an omniscient Adversary and an Aggregator who is advised by N experts. Special cases of this game include predicting if it will rain tomorrow, or if the stock market will go up or down.

For $t = 1, \dots, T$:

1. Each expert $i \in [N]$ advises either yes or no.

4) GUI 實現

首先是功能的設計，這個部分主要是要考慮要設計哪些功能才能讓使用者流暢地進行操作，例如像是”顯示以訓練股票”，或是多種類型的訓練方式讓使用者能夠根據自身需求靈活選擇。

接著是介面的編排及美化，我們花了很多時間在進行個按鈕的編排及相關美觀的設計，目的是為了提供使用者一個直觀且乾淨的介面。

5) 錯誤偵測的設計

由於剛開始我們在還未實施錯誤偵測時，時常發生重複訓練、模型錯亂等問題。因此，我們進一步考慮了在操作過程中可能發生的所有情況，並決定了在出現這些情況時應向使用者顯示什麼信息，以指導他們進行後續的處理。特別是在訓練過程中，我們將可能出現的情況分為了6種，並對每一種情況進行了相應的處理。

三、 未來展望

- 1) 將策略設計更加靈活化，這意味著我們將考慮到資金不會全部投入的情況，並根據實際情況動態調整策略。這樣的設計將使我們的系統更具適應性和靈活性，能夠應對不同的市場狀況和投資者需求。

- 2) 我們希望能夠實際進行投資，從實際操作中獲取反饋信息，並進一步優化我們的專題。通過投資實踐，我們可以更深入地了解市場，發現專題中需要改進的地方，從而不斷提升系統的效能和準確性。
- 3) 我們打算和專題指導教授及中興大學資管系教授一同將專題投稿至 FI-AWARD 2024，參加一個國際性的專題競賽。



四、SWOT 分析圖



上圖為根據此專題主題所製作的 SWOT 分析圖，將它分為四大部分。

◆ 優勢

- 技術創新：對策略和 MWU 算法的自主設計和開發展示了技術上的創新能力，這有助於專題的獨特性和競爭優勢。
- 風險管理和自動化決策：通過策略 +MWU，可以有效地管理風險並自動進行決策

◆ 劣勢

- 無法短期獲利：受限於 MWU 理論，可能無法立即實現獲利，需要長期持續的投資和努力。
- 突發性問題：雖然可以盡可能降低損失，但無法完全避免突發性問題，這可能對投資結果產生不利影響。
- 策略固定：目前的策略可能缺乏靈活性，無法考慮到不同情況下資金不全投入的情況，這可能限制了對市場變化的應對能力。

◆ 機會

- 市場需求增長：隨著金融市場的不斷發展，對於風險管理和自動化決策的需求可能會不斷增長，為專題提供了更廣闊的市場機遇。
- 技術進步：專題可以不斷優化和升級策略和算法，以適應市場的變化和需求。
- 擴展應用範圍：專題所使用的技術和方法可能具有擴展應用範圍的潛力，可以應用於其他領域或問題的解決中。

◆ 威脅

- 市場競爭：金融市場競爭激烈，存在許多競爭者。若無法體現出實際成效可能會被取代。
- 監管變化：金融市場監管政策和法規經常變化，可能對專題的操作和策略產生影響，需要及時應對。
- 技術失效風險：專題所依賴的技術和算法可能存在失效風險，例如模型不準確或決策錯誤，這可能對專題的運作和效果產生不利影響。

五、 組員心得感想

在這次專題的進行過程中，我們作為一個團隊，面臨了許多的挑戰與困難，經過我們不斷的溝通與合作才成功完成了這項專題。

在製作專題的時候我們深刻體會到了團隊合作的力量。在開發策略和進行數據分析的過程中，每位成員都扮演了關鍵的角色。我們學會了如何在壓力下有效溝通，尊重彼此的意見，並利用每個人的專長解決問題。透過持續的討論和合作，我們不僅成功克服了困難，也將複雜的演算法實際應用到股票投資策略中，這是一個非常難得的機會讓我們將理論知識成功轉化為實際操作的能力。

在技術實施方面，我們這次專題使我們深入了解到數據處理和深度學習的實際應用。負責數據處理和模型訓練部分的過程中，我們熟練運用 Python 的套件，如 Pandas、NumPy 以及 PyTorch。這些技能的學習和應用，不僅提高了我們處理大量數據的能力，也增進了我們在特徵工程和模型優化方面的經驗。

我們還學到了如何設計並實現方便使用者的 GUI 介面，這對我們未來的軟件開發及設計工作極具參考價值。透過不斷的測試和調整，我們理解了如何從使用者的角度出發，提供直觀且功能完善的操作介面。不僅改善了應用程序的可用性，也提高了使用者的體驗。

這次專題經驗是一次全面性的學習旅程，它不僅豐富了我們的技術知識和操作技能，更加深了我們對於團隊合作和時間規劃的認識。透過這次經驗，我們更加確信，將學術理論與實際應用相結合，是提升個人能力和實現職業發展的關鍵。未來，我們期待將這些寶貴的學習應用到更廣泛的領域中，繼續探索和創新。

伍、參考文獻

- ◆ [1] Clay. (2020, June 12). LSTM 的原理與輸入輸出格式紀錄.
◆ <https://clay-atlas.com/blog/2020/05/12/pytorch-lstm-%E7%9A%84%E5%8E%9F%E7%90%86%E8%88%87%E8%BC%B8%E5%85%A5%E8%BC%B8%E5%87%BA%E6%A0%BC%E5%BC%8F%E7%B4%80%E9%8C%84/>
- ◆ [2] Data Application Lab. (2018, August 14). LSTM 長短期記憶網路的學習與實現.
https://youtu.be/EC3SvfW0Z_A?si=Rnx9U1WRmDmP3_ST
- ◆ [3] Lucas Liew. (2020, June 24). What is a Walk-Forward Optimization and How to Run It?.
<https://algotrading101.com/learn/walk-forward-optimization/>
- ◆ [4] Sanjeev Arora. (n.d.). The Multiplicative Weights Update Method.
<https://www.cs.princeton.edu/~arora/pubs/MWsurvey.pdf>
- ◆ [5] celestem. (2023, October 26). Online Learning — The Multiplicative-Weight Update Algorithm.
<https://zhuanlan.zhihu.com/p/661862719>
- ◆ [6] 專題老師. (2023, Spring). Online_Learning__MWU.pdf
- ◆ [7] seaotter. (2021, October 16). Tkinter-利用 Python 建立 GUI(基本操作及佈局篇)._
<https://ithelp.ithome.com.tw/articles/10278264?sc=hot>
- ◆ [8] 欣源科技 2023. (2023, September 23). 【Python】easygui 的介绍与使用.
https://blog.csdn.net/weixin_45953322/article/details/127989595
- ◆ [9] 飛飛. (2018, April 10). 【Python】使用 PyInstaller 將 Python 打包 exe 檔. <https://medium.com/pyladies-taiwan/python-%E5%B0%87python%E6%89%93%E5%8C%85%E6%88%90exe%E6%AA%94-32a4bacbe351>
- ◆ [10] 樂羽嘉. (2021, August 25). 行為科學專家：「害怕錯過行情」跟風買股，會讓人走偏._
<https://www.cw.com.tw/article/5117855>

- ◆ [11] 中天新聞網. (2023, August 10). 股票投資失敗 三峽婦被發現陳屍自家廁所.
<https://tw.news.yahoo.com/%E5%BF%AB%E8%A8%8A-%E8%82%A1%E7%A5%A8%E6%8A%95%E8%B3%87%E5%A4%B1%E6%95%97-%E4%B8%89%E5%B3%BD%E5%A9%A6%E8%A2%AB%E7%99%BC%E7%8F%BE%E9%99%B3%E5%B1%8D%E8%87%AA%E5%AE%B6%E5%BB%81%E6%89%80-052559505.html>
- ◆ [12] 謝金河. (2018, December 26). 美股的空前變局.
<https://www.businesstoday.com.tw/article/category/154685/post/201812260012/>
- ◆ [13] 李柏鋒. (2020, March 24). 【美股投資】2020 年武漢肺炎美國股災過程全記錄.
<https://tw.stock.yahoo.com/news/2020%E5%B9%B4%E6%AD%A6%E6%BC%A2%E8%82%BA%E7%82%8E%E7%BE%8E%E5%9C%8B%E8%82%A1%E7%81%BD%E9%81%8E%E7%A8%8B%E5%85%A8%E8%A8%98%E9%8C%84-171816433.html>
- ◆ [14] 玉山證券. (2023, May 29). 想投資不知從何開始？長、中、短各類策略入門詳解.
<https://www.esunsec.com.tw/article/post.aspx?articleid=34>