國立高雄科技大學

金融資訊系

碩士論文

利用隨機森林與支持向量機預測未來股價趨勢變動 以元大高股息（0056）為例

Predicting Future Stock Price Trends Using Random Forest and Support Vector Machines: A Case Study of Yuanta High Dividend Yield ETF (0056)

研究生：蘇崇煜

指導教授：楊耿杰 博士

中華民國 一一二年

利用隨機森林與支持向量機預測未來股價趨勢變動

以元大高股息（0056）為例

Predicting Future Stock Price Trends Using Random Forest and Support Vector Machines:A Case Study of Yuanta High Dividend Yield ETF (0056)

研究生：蘇崇煜
指導教授：楊耿杰　博士

國立高雄科技大學

金融資訊系

碩士論文

A Thesis
Presented to
Presented to Department of Finance and Information National Kaohsiung University of Science and Technology in Partial Fulfillment of the Requirements
for the Degree of
Master of Science

June 2023

Kaohsiung, Taiwan, Republic of China

中華民國一一二年月

利用隨機森林與支持向量機預測未來股價趨勢變動

以元大高股息（0056）為例

研究生：蘇崇煜　　指導教授：楊耿杰　博士

國立高雄科技大學金融資訊系碩士班

摘要

本研究建構隨機森林（Random Forest）與支持向量機（SVM）模型，以各類技術指標當作輸入特徵，對元大高股息（0056）的 2007/12/26 至 2022/11/30 日資料，預測隔天收盤價的漲跌分類，此外對資料進行變異數分析（ANOVA）當做特徵篩選方法，並且對兩種演算法分別使用 python 中的 optuna 套件尋找最適參數，比較兩種演算法之準確度，並驗證特徵篩選方法及參數優化是否有效。

研究結果顯示經過特徵篩選和參數優化後的模型四表現最佳，在依據事件和年份的分割下平均勝率均在 55%以上。特徵篩選對模型準確度的提升效果不如參數優化，使用 Python 中的 optuna 套件進行參數優化能有效提高模型準確度。比較不同模型發現，在隨機森林模型中使用特徵篩選方法的效果較差，但在支持向量機（SVM）模型中效果較好。最終比較結果顯示，使用特徵篩選和參數優化後的支持向量機（SVM）模型預測元大高股息（0056）的準確度優於隨機森林模型。

關鍵字：機器學習、隨機森林、支持向量機、特徵篩選、參數優化

Predicting Future Stock Price Trends Using Random Forest and Support Vector Machines: A Case Study of Yuanta High Dividend Yield ETF (0056)

Student：Chung-Yu Su Advisor：Keng-Chieh Yang

Department of Finance and Information

National Kaohsiung University of Science and Technology

Abstract

This study constructs random forest and support vector machine (SVM) models using various technical indicators as input features to predict the classification of the next day's closing price for Yuanta High Dividend Yield ETF (0056) from December 26, 2007 to November 30, 2022. Additionally, ANOVA is used as a feature selection method, and the Optuna package in Python is used to find the optimal parameters for each algorithm. The accuracy of the two algorithms is compared, and the effectiveness of the feature selection and parameter optimization is verified.

The results show that the model performs best after feature selection and parameter optimization, with an average win rate of over 55% when divided by events and years. Feature selection has less effect on the accuracy of the model than parameter optimization, and using the Optuna package in Python to perform parameter optimization can effectively improve the model's accuracy. When comparing different models, it is found that the effect of using feature selection in the random forest model is poorer, but better in the SVM model. Finally, the comparison results show that the accuracy of predicting Yuanta High Dividend Yield ETF (0056) using the SVM model with feature selection and parameter optimization is better than that of the random forest model.

Keyword：Machine Learning , Random Forest , Support Vector Machine ,

Feature Selection , Parameter Optimization

致謝

目錄

表目錄

圖目錄

第一章 緒論

1.1 研究背景與動機

在現在的社會中，人們常常提到財富自由，財富自由其實就是指當一個人的被動收入大於生活的必要開銷，也就是說，當一個人不需要工作就可以維持日常所需。想要創造財富自由，對於大多數人來說，如果想要只靠著工作薪資來達成其實並不容易，這就意味著必須要創造被動收入，而許多人選擇的管道就是投資。

選擇投資後，首先考慮的就是投資類型與標的，無論是股票、期貨抑或是外匯都有人選擇，其中 ETF（指數股票型基金）也佔了很重要的一環，它結合了基金的多樣化投資特性以及股票的交易便利性，具有很高的靈活性。當投資人沒有時間盯盤或是想要分散投資風險，不想將風險都固定在單一股票時，就可以考慮投資 ETF。

投資 ETF，投資人要考慮自身的風險承擔能力、財務狀況以及投資目標，元大高股息（0056）是目前市場上討論度極高的一項標的，相較另一項也是討論度極高的標的台灣 50（0050），0056 的價格較便宜，產業類別較多，因此風險分散能力較好，也因為是高股息 ETF，獲得的配息較多。但也因為產業結構的關係，價格漲幅較小。

投資人在開始投資後，會開始煩惱，該在哪個價格買進或是賣出，要依據基本面、技術面、還是籌碼面對標的進行分析，以得知接下來的股價漲跌趨勢，並根據結果進行投資以獲得最大利潤。其中技術分析是透過歷史價格與走勢來研究預測未來的價格，相信技術分析的人相信歷史會持續重演，只要找到

規律，就可以在市場上獲得超額報酬，但是技術分析有許多種，要使用哪一種技術指標非常主觀，因此本研究嘗試使用機器學習來解決問題。

機器學習是透過電腦使用演算法及模型，從大量的數據中，找出隱含的規律並學習，藉由這些規律來預測結果。選擇適合的演算法、輸入特徵以及模型的參數，都可以使模型的準確度提高，更有效預測標的的漲跌趨勢。

演算法的部分，隨機森林（Random Forest）具有高準確率、降低過擬合（Overfitting）的機會等特性。支持向量機（Support Vector Machine , SVM）也具有高準確率、可以處理高維度的特徵資料的優點。因此本研究選擇這兩種演算法對元大高股息（0056）進行股價趨勢的預測。

輸入特徵部分，使用變異數分析（ANOVA）對特徵進行篩選，選取其中影響力最大的 5 項特徵輸入模型。模型參數則使用 optuna 套件，對隨機森林的決策樹數量及葉節點最少數量、支持向量機（SVM）的 C、gamma、核函數進行最適參數調整。

綜上所述，本研究對輸入特徵，也就是技術指標進行特徵篩選，選出最有影響力的五項特徵，接著使用隨機森林以及支持向量機（SVM）建構模型，將特徵輸入到模型中，並尋找最適參數，以預測元大高股息（0056）隔日的漲跌趨勢。最後個別將特徵篩選前後、進行參數優化前後的準確度計算出來，比較模型的準確度，並驗證特徵篩選及進行參數優化是否有效提升模型準確度。

1.2 研究目的

本研究基於隨機森林與支持向量機（SVM）模型，由於機器學習模型在特徵過多的情況下，可能造成過擬合（Overfitting），因此使用變異數分析（ANOVA）當作特徵篩選方法，本研究主要探討下列幾點：

（一）使用隨機森林（RandomForest）作為研究方法，並使用變異數分析（ANOVA）當作特徵篩選方法，建立隨機森林模型，並利用 python 中的 optuna 套件來挑選最佳參數，預測分類結果。

（二）使用支持向量機（SVM）作為研究方法，並使用變異數分析（ANOVA）當作特徵篩選方法，建立支持向量機分類模型，並利用 python 中的 optuna 套件來挑選最佳參數，預測分類結果。

（三）比較上述兩種方法預測模型的準確度，並且分析特徵篩選與尋找最適參數是否有效提升模型準確度。

1.3 研究架構

本研究共分為五個章節，第一章為緒論。第二章為文獻回顧，介紹機器學習、隨機森林模型、支持向量機模型（SVM）對於預測股價之應用。第三章為研究方法，介紹輸入特徵、模型架構以及特徵篩選方法。第四章為實證結果，呈現預測的準確度，並對兩種演算法進行比較。第五章為結論，總結研究的結果，並提供後續研究建議。研究流程圖如圖 1-1：

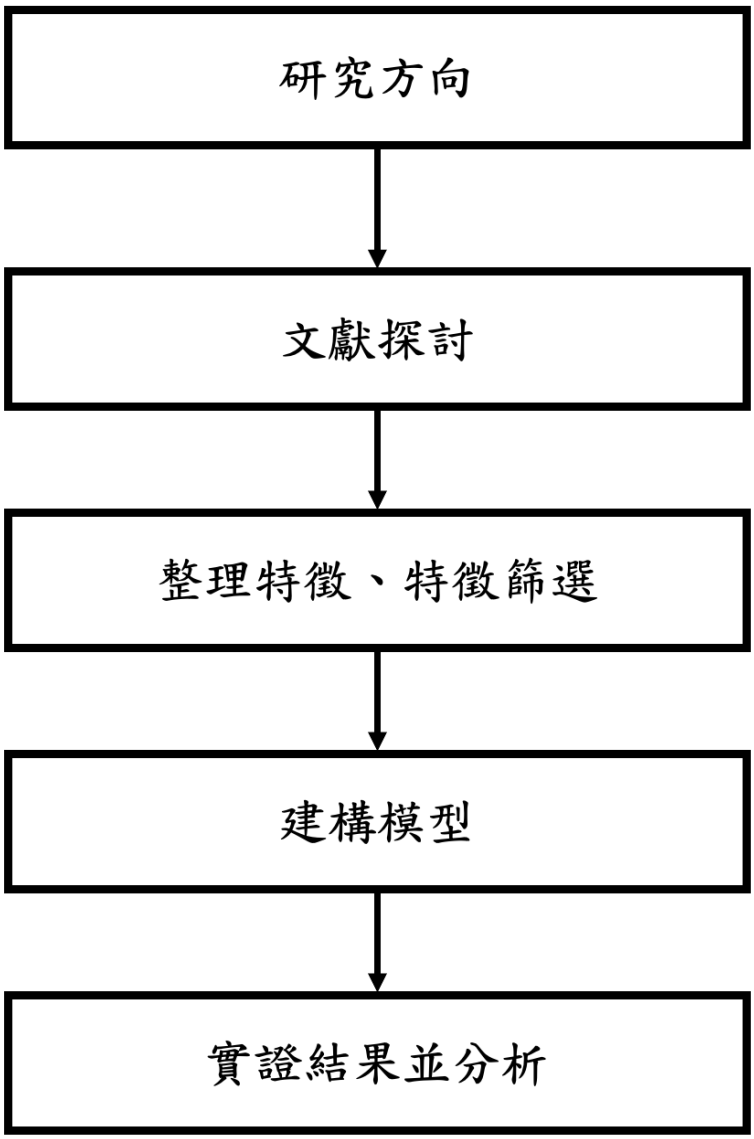

圖 1-1 研究流程

第二章 文獻回顧

2.1 機器學習對股票市場預測的應用

陳怡諭（2021）使用倒傳遞神經網路（BPNN）以及支持向量回歸（SVR）建立股價預測模型，倒傳遞神經網路（BPNN）使用尺度化共軛梯度算法（SCGA）以及梯度遞降算法（GDA），證實出 BPNN-SCGA 模型表現優秀，此外，BPNN-SCGA 模型與 SVR 模型表現優於 BPNN-GDA，因此 BPNN-SCGA 模型與 SVR 模型可以當作有效的股價預測工具。

劉栩憬（2021）利用三種方法探討特徵數目的增加是否可以提升預測台灣加權股票指數模型的效能，分別如下：

1. 使用 12 種技術指標，建立 10 個、31 個、33 個、54 個特徵數量，比較特徵數量與模型效能的關聯性。
2. 建立五種分類模型，羅吉斯回歸、隨機森林、樸素貝葉斯、人工神經網路、支持向量機，比較演算法的預測效能。
3. 將資料分成連續型數值及趨勢型數值，比較數值型態對效能的影響。

結果表示，特徵數目以 31 個及 33 個表現最優，演算法以人工神經網路準確率最佳，數值型態則是趨勢型數值表現較佳。

Hao Li et al.（2018）使用多變量輸入的 LSTM 模型，該模型可以從低相關的特徵中提取重要信息，並透過有較高影響度的輸入控制器來消除噪聲，本研究還加入了其他的特徵，例如其他相關股票的價格，以提升準確度。研究結果表明對於中國股票市場的數據進行預測時，有很好的預測結果。

2.2 支持向量機（SVM）對股票市場預測的應用

紀凱耀（2010）結合粒子群分群法及支持向量機之決策模型應用於台灣加權股價指數預測以技術指標建立個股股價趨勢之預測模型，並訂定選股策略藉以找出適合投資之個股。研究以台灣股票電子類個股為研究對象，經由重要指標篩選以及資料分群後，再以 PSO 判斷隔日是否為漲、跌、持平，並計算其準確率。實證結果證實該研究所提出之 PSO—SVM 預測模 式在經過多次分群測試下，看出本研究模式之預測為持平、下跌效果較佳。

蘇彥廷（2016）利用結合遺傳演算法與支持向量機模型，以各類技術指標值做為輸入特徵，對台灣加權指數 2000 年至 2014 年間日資料，預測未來一週的漲和跌兩分類，並且將預測結果用於期貨投資回測實證上。另外嘗試對訓練資料使用分群方法、F 分數特徵挑選方法及逐步迴歸特徵挑選方法，證實機器學習模型可應用於證券投資領域，依預測的訊號搭配合適的交易策略進行投資，可從市場中獲取可觀的報酬。

李欣隆（2021）結合輿情分析、支持向量機（SVM）以及決策樹建立模型，預測股價漲跌，資料使用 2018/08/01~2021/07/05 的 Google 財經資料及 Yahoo 的股市資料當作搜集資料來源，預測的標的為航運股，結果證實股價漲跌趨勢與財經新聞的情緒存在高相關性，加入新聞輿情指數能夠有效提升模型預測準確率。

Gurjeet Singh（2022）利用 8 種監督式機器學習模型來預測 Nifty 50 指數。實證研究使用的技術包括適應性提升（AdaBoost）、k-近鄰（kNN）、線性回歸（LR）、人工神經網絡（ANN）、隨機森林（RF）、隨機梯度下降（SGD）、支持向量機（SVM）和決策樹（DT）。比較了所使用模型的預測性能，評估結

果表明，隨著數據集的大小增加，適應性提升、k-近鄰、隨機森林和決策樹的表現較差。線性回歸和人工神經網絡在所有模型中顯示出幾乎相同的預測結果，但人工神經網絡在訓練和驗證模型時花費的時間較多。此後，支持向量機在其他模型中表現較佳，但隨著數據集的大小增加，隨機梯度下降的表現較支持向量機佳。

Kara et al.（2011）使用人工神經網路與支持向量機（SVM）預測股票漲跌，並應用於伊斯坦布爾證券交易所的實證，樣本從 2002 年到 2009 年，使用不同的技術指標當作輸入特徵，實證結果顯示，人工神經網路與支持向量機（SVM）都能有效的預測股價漲跌。另外也對人工神經網路與支持向量機（SVM）進行比較，結果顯示人工神經網路的性能在某些情況下比支持向量機（SVM）來的優秀，而該研究之方法可以幫助投資人預測股票漲跌。

Lee, M. C. (2009)使用支持向量搭配複合式特徵篩選方法 F_SSFS，他結合了過濾式特徵篩選與包裝式特徵篩選的優點。為了比較模型準確度，將模型與反向傳導神經網路（BPNN）以及較常見的三種特徵篩選方式進行比較，分別是資訊增益、對稱不確定性和 t 檢定。結果顯示支持向量機（SVM）在預測股價漲跌上勝過 BPNN，而特徵篩選方法則是使用 F_SSFS 會取得最佳結果。

2.3 隨機森林對股票市場預測的應用

楊駿豪（2021）基於人工神經網路、隨機森林、極限梯度提升樹等 3 種分類器，提出一種用於隔日股價方向預測的方法，每個分類器透過手動的超參數調整來訓練預測模型，然後以多數投票的方式產生最終的預測結果。 另外使用套索回歸進行特徵篩選，結果證實執行特徵篩選的模型準確率高於未實行特徵篩選的模型。

洪育民（2022）以台灣 50 指數成分股為預測標的，以隨機森林演算法並基於迴歸樹 對臺灣 50 指數成分股進行一季以後的報酬率預測，藉由特徵擴充、篩選、調整參數後，依照不同的訓練期完成模型建構，並探討影響模型是否成功的主要原因。經實證結果發現，模型解釋能力高需要滿足以下幾點：

（1）訓練期間報酬的波動幅度必須能包含目標期間的報酬波動。

（2）在訓練期間，模型所選出之特徵要能對報酬進行正確分類。

（3）目標期間的特徵和報酬之走勢，要能延續訓練期間所建立的規則。

而失敗的模型，問題大多出在第三點，就是在目標期間，其特徵與報酬之關係改變，導致其走勢與前面訓練期間所建立之規則不同，而其中一個則是因為模型所選出之特徵並不能對報酬做出太好的分類。

洪御仁（2022）分別使用類神經網路、支援向量回歸、隨機森林、決策樹這些演算法對台灣 50 進行股價預測，結果顯示使用隨機森林的準確率高於類神經網路及支援向量回歸。

Abraham R. et al. （2022）使用隨機森林搭配基因演算法（GA）當做特徵篩選方法，揭示股票指數與股票趨勢之間的關係，透過 15 支股票的趨勢來預測模型，實驗結果證明，預測模型的準確率有 80%。其中標普 500 為最有效的標的，CAC40 則為最無效之標的。此研究證實使用國際股票指數預測股票趨勢為有效的。

Basak et al.（2018）使用隨機森林演算法以及 XGBoost 演算法對股票趨勢進行預測，預測標的包含 Nike、Apple、Honda、Amazon 等大廠，特徵則使用技術指標，研究結果顯示兩者的表現均非常出色。計算出準確度、召回率、F-score 等參數之後，顯示模型的有效性。此研究的模型可以用來制定新的交易策

略或是進行投資組合管理，未來可以透過增強樹模型來預測短時間窗口的趨勢，也可以測試不同演算法在股票預測的穩健性。

2.4 特徵挑選方法

Nadir Omer Fadl Elssied et al.（2013）利用支持向量機模型搭配變異數分析（ANOVA）進行垃圾郵件的分類，實證結果顯示基於以上所產生的 FSSVM 模型，比 SVM 本身與當時其他模型，更能準確分辨垃圾郵件。

Arowolo et al.（2016）利用支持向量機結合變異數分析（ANOVA）進行癌症分類預測，以減少檢驗癌症的成本，結果顯示利用變異數分析（ANOVA）可以有效篩選出特徵並提高模型的準確率。

吳晟源（2020）利用模型中特徵與二元響應變量之 F 分布關係當作篩選標準，找出最相關的特徵，並計算出勝算比，探討該特徵對預測結果的影響力，經由篩選後的特徵帶入類神經網路模型，可以減少計算量，提升準確率，研究結果顯示經過特徵篩選後的訓練模型都有很好的鑑別能力。

Chen, Y.W.等人（2006）結合支持向量機（SVM）與不同的特徵篩選方法，試圖使用特徵篩選方法來提升預測準確度，使用的篩選方法分別是：過濾式特徵篩選、包裝式特徵篩選、嵌入式特徵篩選、以及調整權重的特徵篩選，結果顯示使用調整權重的特徵篩選方式，在不同的數據資料以及參數不同的支持向量機（SVM）中，都可以有效提升準確度。

第三章 研究方法

3.1 研究模型

本研究以技術指標當作輸入特徵，經由隨機森林（Random Forest）與支持向量機（Support Vector Machine , SVM）建構模型，預測 0056 元大高股息隔天收盤價的漲與跌分類，並比較兩模型的準確度。

特徵數量共有 17 個，由於避免輸入過多特徵導致過度擬合（Overfitting），因此利用變異數分析（ANOVA）進行特徵篩選，篩選出該時間點最顯著的前五項特徵進行模型訓練。此外，由於支持向量機模型（SVM）需加入懲罰係數 C 與高斯核函數頻寬γ，因此會多加入參數優化的步驟，利用 python 的 optuna 套件來找尋最適參數，並選擇最適合的核函數。為了兩種演算法比較的公平性，也在隨機森林中加入參數優化的動作。研究流程如圖 3-1 所示。

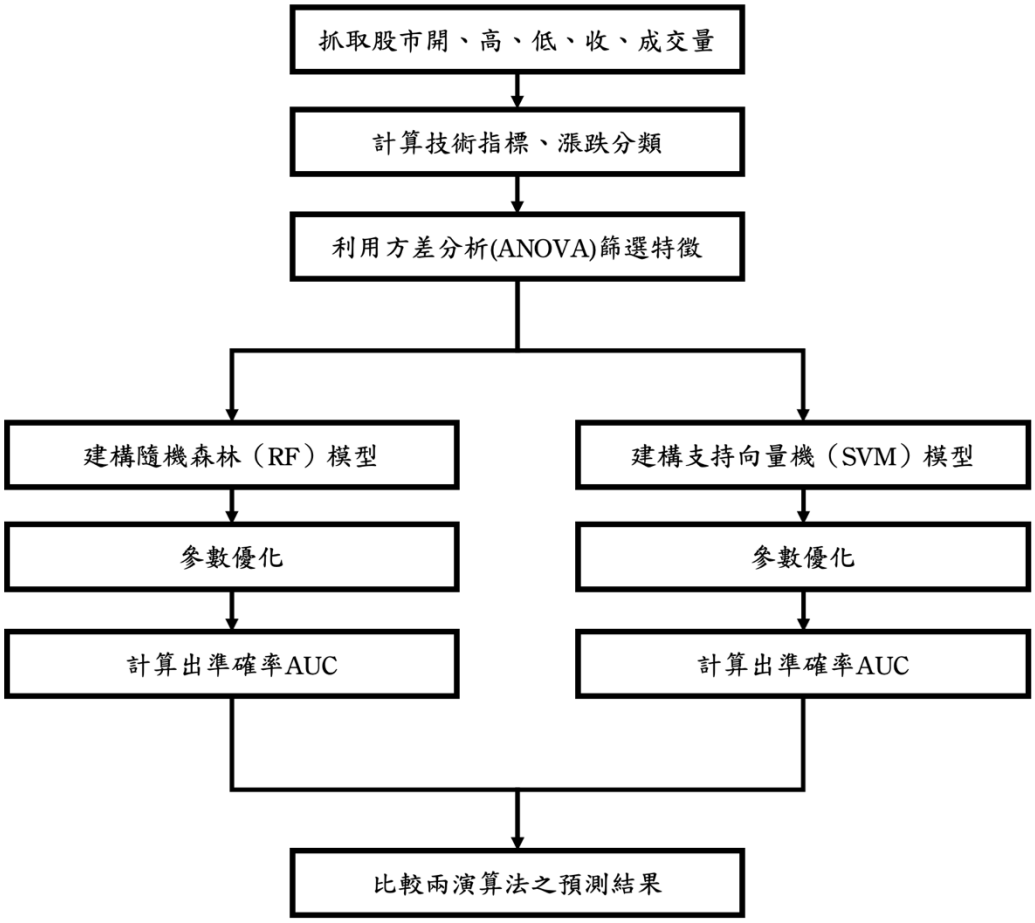

圖 3-1 模型運作流程圖

3.2 預測分類

本研究利用股價漲跌幅當作漲跌分類的依據，如果股價漲跌幅的係數為正，則歸類為漲分類，反之如果股價漲跌幅的係數為負，則歸類為跌分類。

$$\Delta C = (C_{t+1} - C_t) / C_t$$

$$Class_t = \begin{cases} 0, & \Delta C > 0 \\ 1, & \Delta C < 0 \end{cases}$$

ΔC：股價漲跌幅　C_t：t 時點收盤價　$Class_t$：t 時點分類

0：漲分類　1：跌分類

3.3 特徵資料

本研究參考文獻中所使用的技術指標，篩選出以下技術指標當作本次研究的輸入特徵。

3.3.1 布林通道帶寬指標（BandWidth）

布林通道（Bollinger Bands）是在市場中判斷價格近出場時機的指標，由均線及標準差的概念組成，通道由三條線所組成，分別為上軌（股價壓力線）、下軌（股價支撐線）、中軌（平均成本），而這三條線的計算方式如表（）所示，此外本研究設定 N = 5　M = 2。

中軌	週期 N 的簡單移動平均線（SMA）
上軌	中軌 ＋ M 倍價格標準差
下軌	中軌 － M 倍價格標準差

表 3-1 布林通道計算方法

布林通道帶寬指標（BandWidth）是由布林通道指標的提出者 John Bollinger 思考衍生出來的，是將布林通道利用一條線表示的指標，帶寬指標除了會顯示通道的寬度，同時也是通道相對於中心線移動平均線值比例。值越

低，布林通道的寬度越窄，表示呈現壓縮狀態；值越高，表示布林通道的寬度越寬，處於擴展的狀態。計算方法如下：

$$\text{帶寬指標} = \frac{\textit{布林通道上軌} - \textit{布林通道下軌}}{\textit{布林通道中軌}}$$

3.3.2 簡單移動平均線（Simple Moving Average , SMA）

簡單移動平均線（SMA）是將過去一段時間內，每天的收盤價加總，再除以週期天數，常被當作過去一段時間內，投資人買進的平均成本。本研究分別使用 5 日、10 日、20 日、60 日當作輸入特徵。計算方式如下：

$$SMA_n = \frac{C_1 + C_2 + C_3 + \cdots + C_n}{n}$$

SMA_n：n 日收盤價　C_n：第 n 日收盤價

3.3.3 指數平滑移動平均線（Exponential Moving Average , EMA）

指數平滑移動平均線（EMA）是對收盤價進行加權算術平均，給予每根 K 棒不同的權重，越近的 K 棒權重越高，越遠的 K 棒權重越低，對於當天價格的權重更加重視，可以用來判斷價格未來趨勢。

與傳統簡單移動平均線（SMA）相比，指數平滑移動平均線（EMA）對近期價格變化的反應較顯著，更能反映趨勢。本研究使用 5 日、10 日、20 日、60 日當作輸入特徵。計算方式如下：

$$\text{W} = \frac{2}{n+1}$$

$$EMA_t = C_t * W + EMA_{t-1} * (1 - W)$$

W：權重　n：均線週期　C_t：t 時點收盤價　EMA_{t-1}：前一期 EMA

3.3.4 隨機指標（Stochastic Oscillator , KD 指標）

隨機指標（Stochastic Oscillator）是由 George C. Lane 於 1950 年代所提出，主要用來呈現過去一段時間內股價的強弱趨勢。利用動能分析方法計算出 K 值與 D 值，因此又稱 KD 指標，而此處的「隨機」指的是股價在某段時間內，高低區間的波動範圍。

K 值（快線）對於近期的價格趨勢較敏感，而 D 值（慢線）對於近期價格反應較遲緩，常被用來當作進出場的訊號，要計算 KD 值首先要先計算出 RSV 值，計算方式分別如下：

$$RSV_t = \frac{C_t - L_n}{H_n - L_n} * 100$$

$$K_t = RSV_t * \frac{1}{3} + K_{t-1} * \frac{2}{3}$$

$$D_t = K_t * \frac{1}{3} + D_{t-1} * \frac{2}{3}$$

n：經過的交易期間，一般設定為 9 日

C_t：當期收盤價　L_n：n 期內最低價 H_n：n 期內最高價

K_{t-1} 與 D_{t-1}：前一期的 K、D 值，第一期通常預設為 50

K 值與 D 值都會介在 0 ～ 100 之間，當 K 值大於 D 值，為上漲趨勢，而 K 值小於 D 值時，為下跌趨勢。此外，如果 K 值向上突破 D 值，稱為「黃金交叉」，為多方訊號 ; K 值往下跌破 D 值，則稱為「死亡交叉」，為空方訊號。

KD 指標也被當作超買超賣的依據：KD 值大於 80 時，稱為超買，代表股價可能較為強勢，高機率保持繼續上漲，而 KD 值小於 20 時，稱為超賣，代表股價較弱勢，高機率繼續下跌。80 與 20 為參考值，有些人利用其他數字當參考。本研究分別使用 K 值與 D 值當作輸入特徵。

3.3.5 相對強弱指標（Relative Strength Index , RSI）

相對強弱指標（RSI）是 J. Welles Wilder Jr. 在 1978 年發明的，用來衡量股價或市場在特定期間內的強弱力度，數值會介在 0～100 之間，數值越高代表上漲力道越強，越接近 0 代表下跌力道越強。另外，當數值大於 70 時，視為超買，小於 30 時則視為超賣。

通常相對強弱指標（RSI）不會單獨使用，會搭配其他的技術指標一起觀察，例如：週期短的 RSI 指標向上突破週期長的 RSI 指標時，稱為「黃金交叉」，為買進訊號；週期短的 RSI 指標向下跌破週期長的 RSI 指標時，稱為「死亡交叉」，為賣出訊號。

本研究利用 5 日 RSI 與 10 日 RSI 作為輸入特徵，計算方式如下：

Up = n 週期內收盤價上漲幅度的平均

Dn = n 週期內收盤價下跌幅度的平均

$$\text{RSI} = \frac{Up}{(Up+Dn)} * 100$$

Up：上升平均數 Dn：下跌平均數

3.3.6 指數平滑異同移動平均線（MACD）

指數平滑異同移動平均線（MACD）是 1970 年代由 Gerald Appel 所提出，透過計算指數移動平均（EMA）之間的離散程度（DIF）而來，經過雙重平滑處理，可以判斷股票的進出場時機，由於對近期價格變化反應較慢，MACD 適合判斷中長期的趨勢。計算方式如下：

DIF = EMA（短線週期）－ EMA（長線週期）

MACD = EMA（DIF , 9）

OSC = DIF － MACD

本研究設定 DIF 的短線週期為 12 日、長線週期為 26 日，MACD 為 DIF 的指數移動平均，週期設為 9 日，DIF 值與 MACD 值通常會以「線」的型態顯示

在圖表上，而 OSC 為 DIF 與 MACD 的差值，會呈現「柱」的型態。本研究使用 OSC 值當作 MACD 的衡量。

3.3.7 威廉指標（Williams %R , W%R）

威廉指標（W%R）是 1973 年由 Larry R. Williams 所提出，利用分析收盤價與價格波動幅度相對位置的關係，判斷市場是否超買或超賣，數值會介在 0～100%之間，數值越小時，表示收盤價越接近最高價，也就代表市場越處於超買中。本研究使用 5 日與 10 日當作輸入特徵，計算方式如下：

$$W\%R = \frac{H_n - C_t}{H_n - L_n} * 100\% * -1$$

n：週期　Ct：當日收盤價　H_n：週期內最高價　L_n：週期內最低價

在實務上，投資人會多乘一個(-1)，以符合指標值越大代表超買、越小則代表超賣的統一認知，並且以-80與-20當作超買超賣的界線，當數值小於-80時，表示處於超賣情況，而高於-20時，則表示處於超買情況。

本研究使用各種技術指標當作輸入特徵，預測股價漲跌趨勢，總共有 17 種特徵，各個特徵名稱如表 3-2。

表 3-2 輸入特徵一覽表

特徵 1	股價漲跌幅
特徵 2	布林通道帶寬
特徵 3	5 日 SMA
特徵 4	10 日 SMA
特徵 5	20 日 SMA
特徵 6	60 日 SMA
特徵 7	5 日 EMA
特徵 8	10 日 EMA
特徵 9	20 日 EMA

特徵 10	60 日 EMA
特徵 11	K 值
特徵 12	D 值
特徵 13	5 日 RSI
特徵 14	10 日 RSI
特徵 15	MACD
特徵 16	5 日 W%R
特徵 17	10 日 W%R

3.4 特徵篩選方法（Feature Selection）

當輸入特徵過多的時候，除了資料量太大使運算效率降低之外，也有可能會導致過擬合（Overfitting），進一步影響模型運算的性能，使預測的準確率下降，因此本研究加入了變異數分析（ANOVA）當作特徵篩選的方法，並且觀察到加入特徵篩選可以有效提升模型準確率。

本研究使用 Python 套件 scikit-learn 中的 SelectKBest 對所有輸入特徵進行挑選，SelectKBest 這個套件包含了兩個參數，分別是評估方法 score_func 與前幾名特徵 k，本研究採用 ANOVA 當作分類特徵的評估方式，因此 score_func 使用 f_classif，並找出最相關的 5 個特徵放入模型訓練，示意如下：

SelectKBest（score_func = f_classif , k = 5）

變異數分析（ANOVA）又稱 F 檢驗，是由 R.A.Fister 提出的，原本是用來檢驗各組間是否存在顯著差異，計算出來的 F 值越大，則代表各組間存在顯著差異，而應用在特徵篩選時，會先計算出各個特徵與目標值的 F 值，接著找出影響模型最多的 5 個特徵，並將這 5 個特徵放入模型中訓練。

3.5 建構模型

3.5.1 隨機森林（Random Forest）

隨機森林（Random Forest）是基於決策樹分類器的組合學習演算法，一開始是由何天琴於 1995 年提出「隨機決策森林」，之後 L. Breiman 和 Adele Cutler 在 2001 年擴展了這個演算法並提出了「隨機森林」，會稱之為「森林」是因為這個演算法包含了許多決策樹，將這些決策樹的資料合併在一起，確保能夠得到最準確的預測結果。

3.5.1.1 決策樹（Decision Tree）

決策樹（Decision Tree）屬於監督式學習，用來解決分類的問題，資料輸入從樹根開始，接著進入各個節點進行分類，節點分成根節點、內部節點以及葉節點三種。根節點是最上面的節點，而內部節點用來判斷條件，最後進入到葉節點，也就是該資料的最終分類。

決策樹的基本演算概念如下：

1. 設定資料：將資料分成兩組，分別是 Training Data（訓練資料）與 Testing Data（測試資料）。
2. 生成決策樹：使用 Training Data 建構決策樹，在每一個內部節點根據條件與篩選方法評估分支依據。
3. 剪枝：移除決策中分辨能力較差的節點來縮小決策樹的規模，降低模型複雜度，也減少過擬合（Overfitting）機率。
4. 重複以上步驟直到所有新節點都是葉節點，也就是最終結果。

以下舉例說明：

假設房屋業者對於客戶購買房屋的意願做調查，而某些特徵會影響客戶的購買意願，特徵如下：

1. 外觀：順眼、不順眼

2. 坪數：大於 30 坪、小於 30 坪

3. 裝潢：喜歡、不喜歡

4. 價錢：500 萬以下、500～1000 萬、1000 萬以上

5. 貸款：通過、不通過

外觀、坪數、裝潢、價錢、貸款為特徵名稱，資料一開始會進入根節點，並依據特徵條件判斷前進方向，進入內部節點或葉節點，如果進入內部節點會再繼續進入判斷，倘若無法再分割則進入葉節點，也就是最終結果，如此一來決策樹就完成了，模型如下圖所示。

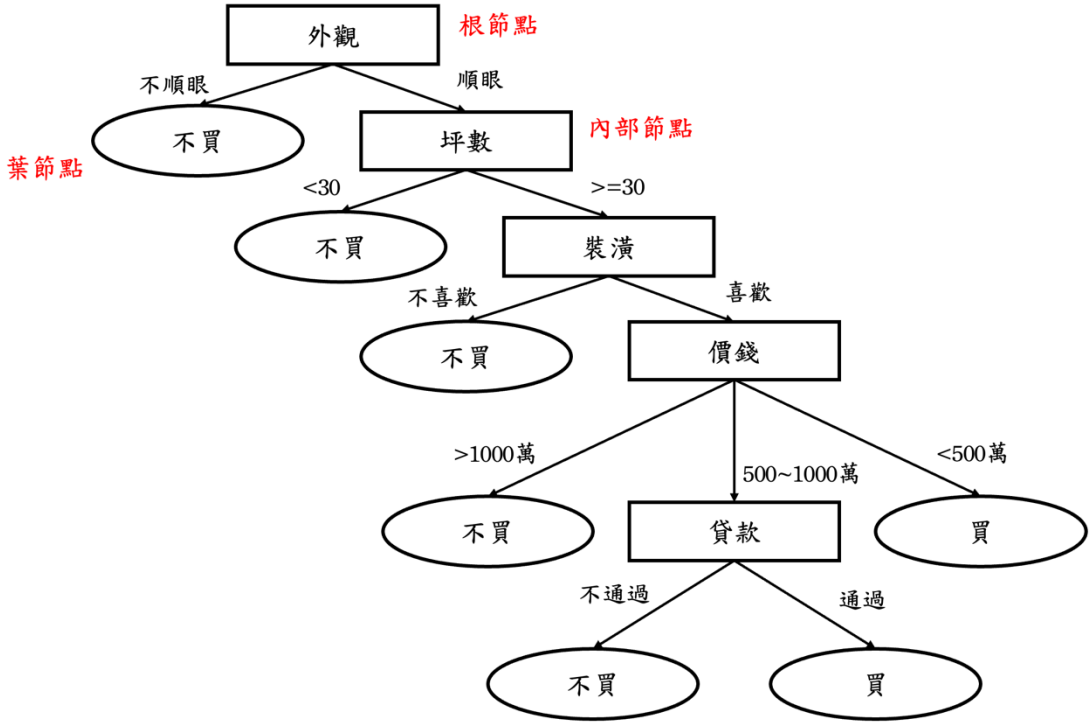

圖 3-2 決策樹模型

建構出模型後，就能依據決策樹的走向來判斷客戶的購買意願，而決策樹的內部節點並不是隨機的，而是依據屬性選擇指標做決定，屬性選擇指標主要是用來選擇某個屬性，透過該屬性將帶有分類標記的資料做分割，接下來會簡單介紹隨機森林所使用的 CART 演算法。

分類回歸樹（Classification And Regression Tree , CART）是 1980 年代由 Friedman 等人提出，是使用二分法的技術，可以根據吉尼係數（Gini）解決分

類問題，也可以使用變異縮減解決回歸問題。由於本研究採用分類方法，因此介紹 CART 分類樹。

CART 分類樹是以吉尼係數來選擇特徵，假設資料集 D 中有 K 個類別，其中P_k為在 D 中歸類為 j 類別的機率，吉尼係數通常會在 0~1 之間，越接近 0 表示結果越好，計算方式如下：

$$Gini(D) = \sum_{k=1}^{k} P_k(1 - P_k) = 1 - \sum_{j=1}^{n} P_k^2$$

利用 A 特徵分類資料集 D，分成D_1與D_2，根據條件進行二分法的吉尼係數為：

$$Gini_A(D) = \frac{|D_1|}{|D|} Gini(D_1) + \frac{|D_2|}{|D|} Gini(D_2)$$

吉尼不純度降低值（Gini Impurity Reduction Value）用來衡量分類後對於吉尼係數的改善量，數值越大表示分類越有效，計算方式如下：

$$\Delta Gini(A) = Gini(D) - Gini_A(D)$$

最後挑選吉尼不純度降低值最大，或是吉尼係數$Gini_A(D)$最小的特徵作為分類的依據。

3.5.1.2 隨機森林（Random Forest）

隨機森林屬於集成學習（Ensemble Learning）中的 Bagging 算法，Bagging 是一種隨機抽取的方法，從原始資料集中隨機抽取樣本與特徵，放入模型訓練，訓練完後再將抽取出來的樣本放回原始資料集中。

隨機森林簡單來說就是由 Bagging 與 CART 決策樹建構而成的，隨機森林可以分成兩個部分，分別是「隨機」和「森林」,「森林」指的是由許多決策樹建構成一片森林，而「隨機」指的則是隨機樣本與隨機特徵，以下整理了隨機森林的簡單建構過程。

1. 隨機樣本：從資料集中隨機抽取樣本。

2. 隨機特徵：從所有特徵中隨機抽取不定數量特徵，選擇最好的特徵當作節點。

3. 建構 CART 決策樹：重複以上步驟幾次，就形成幾棵決策樹。

4. 形成森林：將所有決策樹匯集成隨機森林，並經由簡單投票多數決判斷資料屬於哪一種分類。

3.5.2 支持向量機（Support Vector Machine , SVM）

支持向量機（SVM）於 1995 年由貝爾實驗室 Vapnik 博士團隊與 AT&T 實驗室所提出，屬於監督式學習，可以用來解決二元分類和多元分類問題，主要概念是找到一個超平面（Hyperplane），讓資料分成兩個類別，使分類的誤差最小，而超平面與兩個類別的間隔（margin）要最大，且與兩個間隔的距離相等。圖 3-3 為支持向量機（SVM）最簡單的分割概念。

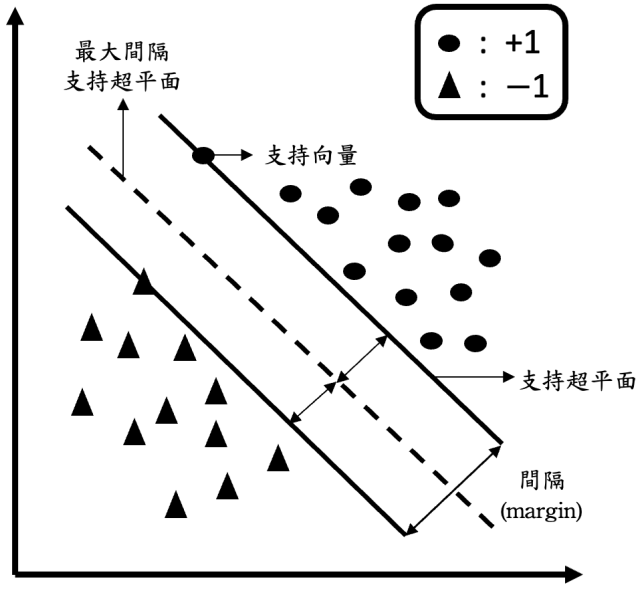

圖 3-3SVM 概念圖

支持向量機（SVM）根據不同的資料類型可以分成三種，分別是：

1. 資料線性可分割支持向量機
2. 資料線性不可分割支持向量機
3. 非線性支持向量機

資料線性可分割支持向量機指的是可以用一條線或一個平面將資料分成兩個類別，不存在混雜在一起的情況，並找出最大間隔超平面，也就是說資料分類非常明確，不會允許有一點點的誤差，示意圖如下：

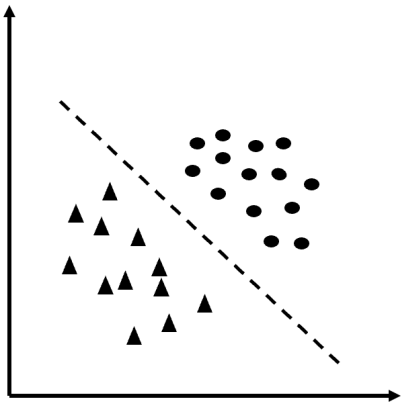

圖 3-4 資料線性可分割

但因為不是所有資料都可以如此容易的將資料分成兩類，這時候就需要資料線性不可分割支持向量機，也就是軟間隔 SVM。軟間隔（Soft Margin）的概念在 1990 年代被提出，指的是在線性可分割 SVM 中允許一些錯誤，使尋找最大間隔超平面的條件放寬一些，會引入一項懲罰因子 C 來控制錯誤的數量。懲罰因子 C 決定了模型對於錯誤分類的容忍度，C 越大，模型對於分類越嚴格，C 越小，將允許越多錯誤，容易產生過擬合（Overfitting）問題。示意圖如下：

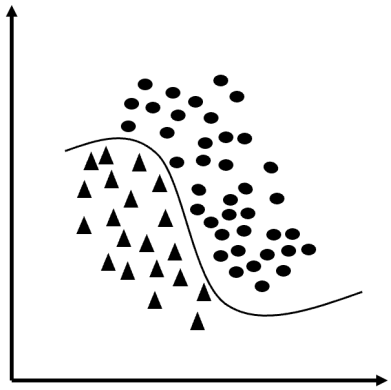

圖 3-5 資料線性不可分割

由於並非所有資料皆可以使用線性分割，因此有了非線性資料支持向量機的出現，是透過核函數進行更有效的分類，將資料映射到較高維度的特徵空

間，並在較高維度空間進行分類，不過運算量也會隨之增加，因此執行速度會較慢。以下是常見的核函數：

1. 線性核函數（Linear）
2. 多項式核函數（Polynomial）
3. 徑向基核函數（Radial Basis Function , RBF）
4. S 型核函數（Sigmoid）

圖 3-6 為非線性資料分割概念圖。

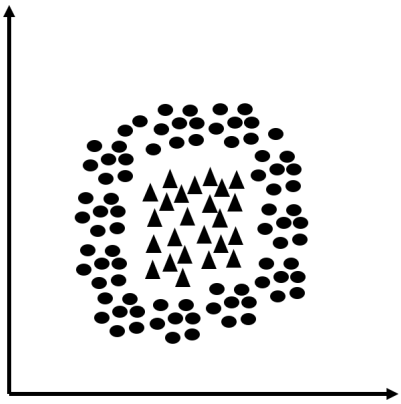

圖 3-6 非線性資料分割

本研究採用非線性支持向量機模型，為了求出最佳參數並方便比較，建構完模型後自行輸入懲罰因子的範圍與不同核函數，利用 optuna 套件來尋找最佳參數，並利用最佳參數與核函數的準確率與原本預設的參數做比較，取得最終結果。

第四章 實證結果

本章將會分成六節，第一節說明樣本資料與切割時間，第二節說明特徵篩選結果，第三節說明模型設定，第四節說明隨機森林的估計結果，第五節說明支持向量機（SVM）的估計結果，第六節為隨機森林與支持向量機（SVM）兩演算法之比較。

4.1 樣本資料與切割時間

本研究採用元大高股息（0056）的開、高、低、收日資料，使用 Talib 套件計算出每日技術指標與股價漲跌幅，資料來源為 XQ 全球贏家，資料期間為 2007 年 12 月 26 日～2022 年 11 月 30 日，總共 3683 筆資料。

切割時間分為兩個部分，第一部分是根據台灣加權指數，找尋波動較大的時間點做切割，例如：2008 年金融海嘯時期、2015 中國股災時期等，詳細切割時點如表（）。而第二部分是依據年份做切割，從 2008 年到 2022 年。

表 4-1 依事件及趨勢分割時點

事件	時點
2008 金融海嘯	2008-06-01～2008-12-31
金融海嘯後反彈	2009-01-01～2010-01-31
2011 美債危機	2011-08-01～2011-12-31
2015 中國股災	2015-04-01～2015-08-31
2018 中美貿易戰	2018-10-01～2019-01-14
2020 Covid-19 疫情	2020-01-01～2020-03-30
2020 疫情趨緩反彈	2020-03-20～2021-12-31

4.2 特徵篩選結果

為了方便比較，在進行特徵篩選時都只篩選出最有效的五個特徵，因此有些期間在進行特徵篩選前的準確度高於篩選後的，可能是因為其他高影響力的特徵並未被篩入，進而影響準確度。因此本研究之準確度將會以進行特徵篩選後之準確度為主。表 4-2 為各段期間篩選出之前五大特徵。

表 4-2 隨機森林法之輸入特徵

期間	特徵名稱
2008 金融海嘯	股價漲跌幅、布林通道帶寬、K、D、MACD
金融海嘯後反彈	SMA5、SMA10、EMA5、EMA10、EMA20
2011 美債危機	K、 D、RSI5、RSI10、W%R10
2015 中國股災	股價漲跌幅、布林通道帶寬、K、D、MACD
2018 中美貿易戰	SMA5、EMA5、EMA10、K、D
2020 Covid-19	SMA60、RSI5、RSI10、W%R5、W%R10
疫情後反彈	SMA5、SMA10、EMA5、EMA10、EMA20
2008	股價漲跌幅、布林通道帶寬、K、D、MACD
2009	SMA5、SMA10、EMA5、EMA10、EMA20
2010	K、 D、RSI5、RSI10、W%R10
2011	SMA20、SMA60、EMA20、EMA60、MACD
2012	SMA10、SMA20、EMA10、EMA20、EMA
2013	SMA10、SMA20、EMA10、EMA20、EMA60
2014	SMA5、EMA5、EMA10、EMA20、EMA60
2015	股價漲跌幅、SMA20、SMA60、EMA20、EMA60
2016	股價漲跌幅、SMA20、RSI5、RSI10、MACD
2017	布林通道帶寬、SMA5、EMA5、EMA10、EMA60
2018	K、RSI5、RSI10、WILLAR5、WILLAR10

2019	布林通道帶寬、SMA20、EMA20、MACD、WILLAR10
2020	SMA10、SMA20、SMA60、EMA20、EMA60
2021	SMA5、EMA5、K、D、MACD
2022	布林通道帶寬、SMA5、SMA10、SMA20、EMA10

以上統計出特徵挑選的數量為：股價漲跌幅 5 次、布林通道帶寬 5 次、SMA5 7 次、SMA10 6 次、SMA20 7 次、SMA60 4 次、EMA5 7 次、EMA10 8 次、EMA20 10 次、EMA60 7 次、K 8 次、D 7 次、RSI5 5 次、 RSI10 5 次、MACD 7 次、W%R5 2 次、W%R10 5 次。其中被篩選為最有效特徵最多次的為 20 日 EMA，而最少次的為五日 W%R。

4.3 模型設定

本研究除了特徵篩選方法外，加入 python 的 optuna 套件來尋找最佳參數，依據特徵篩選方法以及參數優化方法，本研究共採用 4 種模型，模型的配置分別如下：

一、 不使用特徵篩選以及參數優化

二、 不使用特徵篩選但加入參數優化

三、 加入特徵篩選但不加入參數優化

四、 加入特徵篩選及參數優化

估計後的結果都將以進行參數優化後的結果，也就是模型二和模型四為主，進行比較及分析。

4.4 隨機森林估計結果

本研究利用隨機森林演算法對各段期間進行估計，以下分別列出四種模型訓練後的準確度。

表 4-3 隨機森林估計結果–依事件分割

隨機森林				
期間	模型一	模型二	模型三	模型四
	不使用特徵篩選		加入特徵篩選	
	未尋找 最適參數	尋找 最適參數	未尋找 最適參數	尋找 最適參數
2008 金融海嘯	40%	43%	43%	50%
金融海嘯後反彈	62%	67%	65%	73%
2011 美債危機	55%	68%	41%	55%
2015 中國股災	40%	40%	62%	62%
2018 中美貿易戰	47%	60%	53%	53%
2020 Covid-19	73%	73%	64%	82%
疫情後反彈	51%	54%	51%	62%

如果以隨機方式分類，準確度的期望值應為 50%，因此訓練模型時，以 50%當作模型好壞的衡量標準。表 4-3 可以看到模型二在預測 2008 金融海嘯及 2015 中國股災時，準確度分別只有 43%和 40%，預測能力並不好，但在預測 2011 美債危機及 2020 Covid-19 疫情期間時表現較佳，準確度來到 68%及 73%。而模型四在大部分時間的準確度都有超過 50%，只有在預測 2008 金融海嘯時為 50%，因此初步推測，模型四表現較模型二優秀。

表 4-4 隨機森林估計結果–依年份分割

隨機森林				
期間	模型一	模型二	模型三	模型四
	不使用特徵篩選		加入特徵篩選	
	未尋找 最適參數	尋找 最適參數	未尋找 最適參數	尋找 最適參數
2008	40%	43%	43%	50%
2009	59%	63%	69%	71%
2010	61%	69%	53%	65%
2011	48%	48%	48%	54%
2012	48%	54%	50%	50%
2013	58%	62%	38%	52%
2014	64%	64%	52%	56%
2015	51%	59%	51%	51%

2016	39%	51%	49%	53%
2017	54%	62%	56%	64%
2018	44%	52%	56%	58%
2019	59%	65%	55%	65%
2020	63%	67%	49%	53%
2021	51%	61%	55%	59%
2022	44%	56%	47%	53%

在各年份的預測中，模型二表現最好的是 2010 年，準確度為 69%，而表現最差的是 2008 年，準確度只有 43%。而模型四表現最好的是 2009 年，準確度為 71%，表現最差的為 2008 年，準確度為 50%。可以發現隨機森林在預測 2008 年股價趨勢時，表現不太好，而在其他時間點也沒有非常突出。

此外，在估計結果中比較模型一和模型三，可以發現進行特徵篩選後，準確度上升的有 11 項、降低的 7 項、持平的 4 項，共 22 項。而模型二和模型四，特徵篩選後準確度上升的有 6 項、降低的 7 項、持平的 9 項，共 22 項。我們可以發現使用特徵篩選方法來提高準確度的效果不顯著。

至於在參數優化方面，可以看到無論是模型一、模型二或是模型三、模型四的比較，在進行參數優化之後的準確度，最差就是保持不變，而其他的準確度都有上升，這表示使用 python 套件 optuna 尋找最適參數的方法是有效的。

4.5 支持向量機（ＳＶＭ）估計結果

表 4-5 支持向量機（SVM）估計結果–依事件分割

支持向量機（SVM）				
期間	模型一	模型二	模型三	模型四
	不使用特徵篩選		加入特徵篩選	
	未尋找 最適參數	尋找 最適參數	未尋找 最適參數	尋找 最適參數
2008 金融海嘯	43%	57%	50%	57%
金融海嘯後反彈	62%	71%	65%	71%
2011 美債危機	41%	77%	36%	77%
2015 中國股災	57%	57%	57%	71%

2018 中美貿易戰	47%	60%	53%	53%
2020 Covid-19	50%	80%	73%	91%
疫情後反彈	51%	62%	51%	51%

在根據事件劃分的期間中，可以發現如果同樣以 50%當作模型好壞的判斷依據，模型二和模型四的表現都非常好，尤其是在 2020 年 Covid-19 疫情的情況下，推測是因為當時的股價趨勢非常顯著的往下跌落，幾乎沒有任何反彈，使模型的預測能力較好。

表 4-6 支持向量機（SVM）估計結果—依年份分割

支持向量機（SVM）				
期間	模型一	模型二	模型三	模型四
	不使用特徵篩選		加入特徵篩選	
	未尋找最適參數	尋找最適參數	未尋找最適參數	尋找最適參數
2008	51%	51%	46%	51%
2009	55%	69%	55%	63%
2010	59%	65%	65%	65%
2011	44%	50%	46%	64%
2012	44%	62%	44%	68%
2013	50%	60%	48%	62%
2014	58%	58%	52%	58%
2015	43%	53%	51%	53%
2016	47%	63%	57%	63%
2017	58%	58%	58%	62%
2018	58%	66%	58%	60%
2019	63%	65%	61%	76%
2020	55%	59%	57%	61%
2021	53%	59%	55%	59%
2022	42%	60%	42%	60%

在各年份的預測中，模型二和模型四同樣表現的較佳，不過比起依據事件分割，根據各年份分割的模型表現較差，最高只有 76%，因此可以得知，使用支持向量機（SVM）模型，當有事件發生，股票走勢較明顯時，模型的表現較優秀，預測能力較好。

接著比較模型一和模型三，經過特徵篩選後準確度上升的有 10 個、持平的 7 個、下降的 5 個。而模型二和模型四，特徵篩選過後，準確度上升的 8 個、持平的 10 個、下降的 4 個。雖然持平的部分佔多數，不過在支持向量機（SVM）中，特徵篩選是可以提升準確度的方法。

至於進行參數優化的結果，分別觀察模型一、模型二以及模型三、模型四，可以看到所有準確度的成長幅度，最低是持平，其餘的都有顯著提升，因此得到結論，使用 python 套件 optuna 來挑選最適參數的方法可以有效提升模型訓練的準確度。

4.6 隨機森林與支持向量機（SVM）估計結果比較

圖 4-1～圖 4-4 為隨機森林與支持向量機（SVM）在模型一到模型四，各段期間之平均準確度，也就是平均勝率，本研究以平均勝率來決定兩演算法比較之依據。

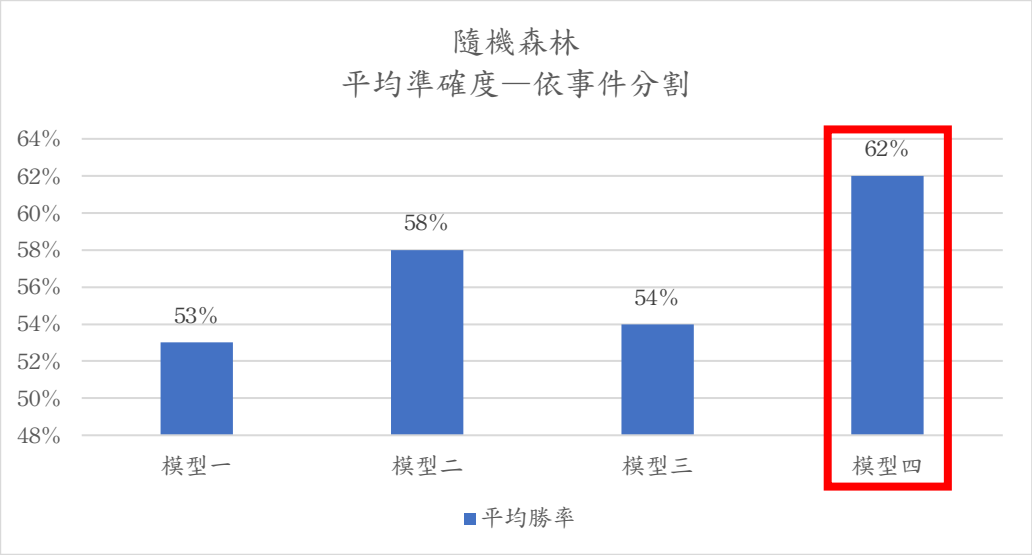

圖 4-1 隨機森林—依事件分割平均勝率

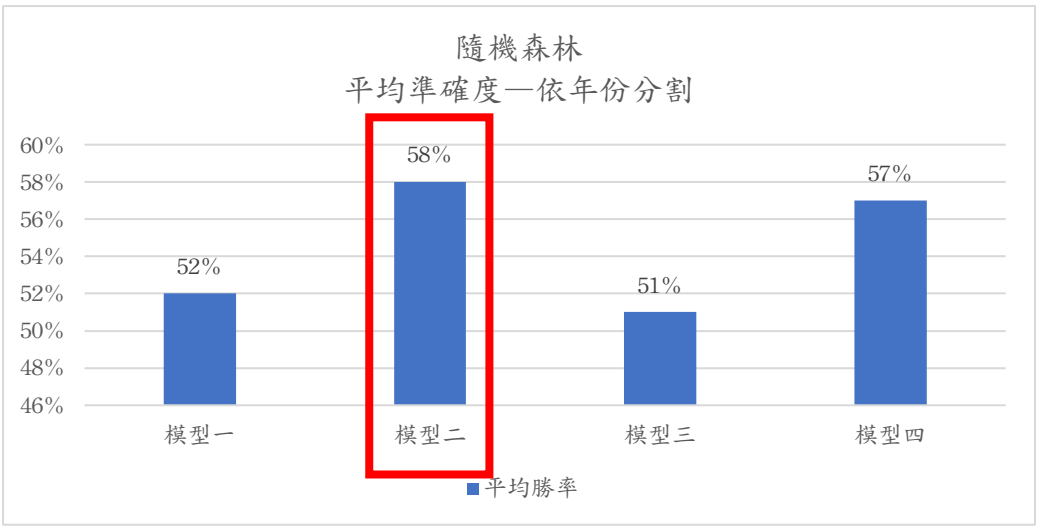

圖 4-2 隨機森林—依年份分割平均勝率

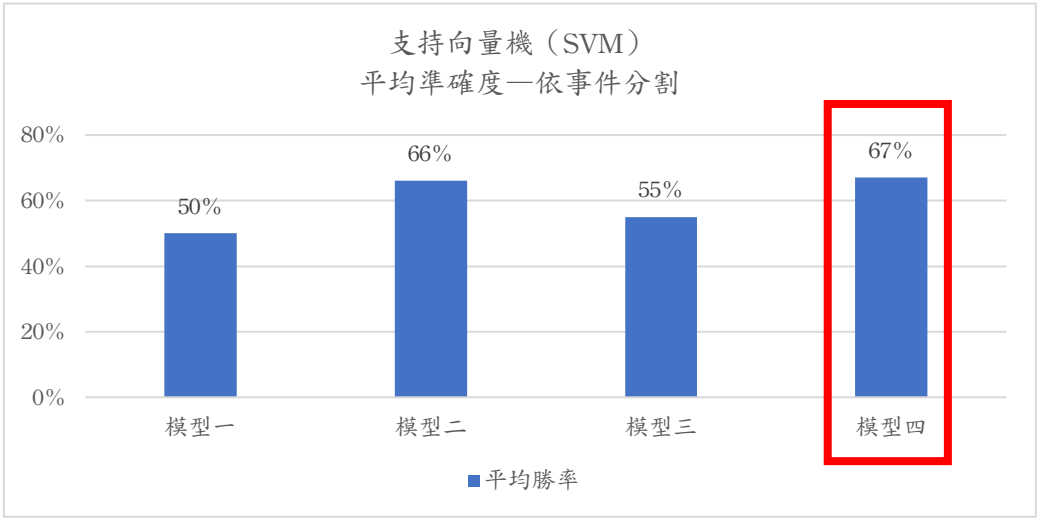

圖 4-3 支持向量機（SVM）—依事件分割平均勝率

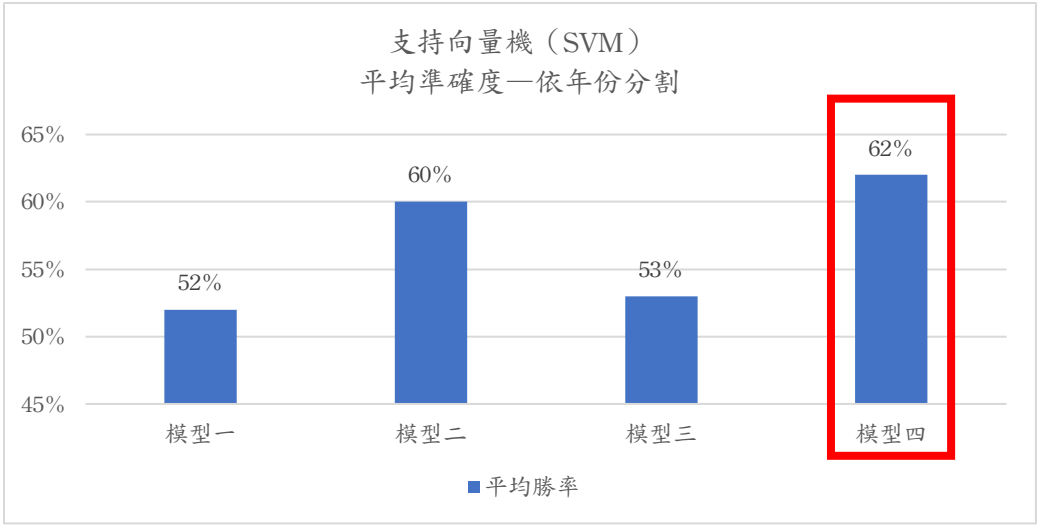

圖 4-4 支持向量機（SVM）—依年份分割平均勝率

由上圖可以看到模型四只有在使用隨機森林並依據年份分割時，平均勝率為第二名，而在其他設定的平均勝率都為最高，因此本研究使用模型四來當作兩演算法比較之依據。

本研究使用隨機森林法與支持向量機（SVM）搭配變異數分析（ANOVA）特徵篩選方法以及進行參數優化，對元大高股息（0056）進行股價趨勢預測，並依據準確度，對兩演算法進行比較。

表 4-7 兩演算法依事件分割之準確度

期間	隨機森林	支持向量機（SVM）
2008 金融海嘯	50%	57%*
金融海嘯後反彈	73%*	71%
2011 美債危機	55%	77%*
2015 中國股災	62%	71%*
2018 中美貿易戰	53%	53%
2020 Covid-19	82%	91%*
疫情後反彈	62%*	51%
平均勝率	62%	67%

註：後方之*符號代表準確度較高者。

表 4-8 兩演算法依年份分割之準確度

期間	隨機森林	支持向量機（SVM）
2008	50%	51%*
2009	71%*	63%
2010	65%	65%
2011	54%	64%*
2012	50%	68%*
2013	52%	62%*
2014	56%	58%*
2015	51%	53%*
2016	53%	63%*
2017	64%*	62%
2018	58%	60%*
2019	65%	76%*
2020	53%	61%*
2021	59%	59%

2022	53%	60%*
平均勝率	57%	62%

註：後方之*符號代表準確度較高者。

根據表 4-7 及表 4-8，可以看到兩張表的平均勝率比較，兩演算法依據事件分割的平均勝率優於依據年份分割的平均勝率，因此得知模型在進行訓練時，趨勢越明顯的情況下，預測能力較好。

另外可以看到支持向量機（SVM）預測的準確度較隨機森林好的有 15 個，而兩者相同的有 3 個，隨機森林較支持向量機（SVM）的有 2 個，再加上無論是依據事件分割還是依據年份分割，支持向量機（SVM）的平均勝率都較隨機森林表現來的好，因此，本研究得到結論，在相同的特徵篩選方法、同樣經過參數優化過程的情況下，支持向量機（SVM）的預測能力比隨機森林來的好。

第五章 結論與建議

本章分成兩個小節，第一小節為本研究根據實證結果整理之結論，第二小節為本研究建議未來研究者可以研究之方向。

5.1 結論

本研究對輸入特徵進行特徵篩選，選出最有影響力的五項特徵，接著使用隨機森林以及支持向量機（SVM）建構模型，將特徵輸入到模型中，並尋找最適參數，以預測元大高股息（0056）隔日的漲跌趨勢。最後個別將特徵篩選前後、尋找最適參數前後的準確度計算出來，並比較模型的準確度，以及驗證特徵篩選及尋找最適參數是否有效提升模型準確度。

本研究根據是否進行特徵篩選以及參數優化，共分成四種模型，研究結果表明，無論是隨機森林還是支持向量機（SVM），表現最好的都是經過特徵篩選並進行參數優化後的模型四，無論是依據事件還是年份分割，平均勝率都在55%以上。

本研究發現，想要使用特徵篩選方法來提升模型準確度，效果並不理想。不過利用 python 套件 optuna 來進行參數優化，可以有效提升模型準確度。各個模型的比較如下：

1. 模型一與模型二之比較：在隨機森林模型以及支持向量機（SVM）模型中，在不使用特徵篩選方法的情況下，進行參數優化都能提升準確度。
2. 模型三與模型四之比較：在隨機森林模型以及支持向量機（SVM）模型中，在使用特徵篩選方法後，進行參數優化都能提升準確度。
3. 模型一與模型三之比較：在隨機森林模型以及支持向量機（SVM）模型中，特徵篩選方法都能提升準確度。
4. 模型二與模型四之比較：在隨機森林模型中，使用特徵篩選方法來提升準確度的效果較差。但在支持向量機（SVM）模型中效果較好。

本研究使用模型四對隨機森林以及支持向量機（SVM）進行建構並分析後，對兩種演算法之結果進行比較，發現使用支持向量機（SVM）搭配特徵篩選方法以及進行參數優化後，對元大高股息（0056）之預測準確度大於隨機森林預測之結果。

5.2 後續研究建議

本研究所研究之過程與結果，在經由反思後尚有部分不足，因此提出建議，讓未來研究者可以進行後續研究。

1. 由本研究探討之分類問題改成回歸問題：將隨機森林分類器改成隨機森林回歸器，支持向量機（SVM）改成支持向量回歸（SVR），亦或是改成其餘演算法進行回歸分析，經由迴歸分析預測精準股價。
2. 加入績效實證：本研究僅預測股價漲跌趨勢，但並無將預測之結果投入真實股市進行績效回測，後續可考慮將分類結果或是上面提及的回歸結果進行績效回測。
3. 特徵數量：本研究使用之特徵數量只有 17 項，後續可將原始特徵數增加，並且使特徵篩選後之特徵數量也增加，驗證特徵篩選方法是否有效。此外，特徵不僅限於技術面，可以將籌碼面以及基本面納入。

文獻列表

中文文獻：

1. 蘇彥廷（2016）支持向量機模型在台灣加權股價指數趨勢之預測，國立中山大學財務管理學系，碩士論文
2. 洪育民（2022），以隨機森林演算法對臺灣 50 指數成分股進行報酬率預測，國立高雄科技大學金融資訊研究所，碩士論文
3. 楊駿豪（2021），基於機器學習預測股價漲跌趨勢，國立台北科技大學管理學院高階管理碩士雙聯學位學程，碩士論文
4. 陳怡諭（2021），應用機器學習方法於股價預測之研究，龍華科技大學企業管理系，碩士論文
5. 李欣隆（2021）基於機器學習與輿情分析對股價漲跌的預測-以航運股為例，國立東華大學資訊工程學系，碩士論文
6. 劉栩憬（2021）運用機器學習與深度學習技術建構股價漲跌預測模式-以台灣加權股價為例，致理科技大學企業管理系服務業經營管理碩士專班，碩士論文
7. 吳晟源（2020）以統計顯著性篩選特徵之機器學習建模研究，龍華科技大學電機工程系，碩士論文

英文文獻：

1. Ballings, M., Van den Poel, D., Hespeels, N., & Gryp, R. (2015). Evaluating multiple classifiers for stock price direction prediction. Expert Systems with Applications, 42(20), 7046-7056.
2. Gurjeet Singh（2022）. Machine Learning Models in Stock Market Prediction. International Journal of Innovative Technology and Exploring Engineering

(IJITEE) ISSN: 2278-3075 (Online), Volume-11 Issue-3.

3. Nadir Omer Fadl Elssied et al.（2013）A Novel Feature Selection Based on One-WayANOVA F-Test for E-Mail Spam Classification. Research Journal of Applied Sciences, Engineering and Technology 7(3): 625-638, 2014
4. Arowolo, M.O. et al.（2016）A Feature Selection Based on One-Way-ANOVA for Microarray Data Classification. Al-Hikmah Journal of Pure & Applied Sciences Vol.3 (2016): 30-35
5. Chen, Y. W., & Lin, C. J. (2006). Combining SVMs with various feature selection strategies. In Feature extraction (pp. 315-324). Springer Berlin Heidelberg.
6. Kara, Y., Boyacioglu, M. A., & Baykan, Ö . K. (2011). Predicting direction of stock price index movement using artificial neural networks and support vector machines: The sample of the Istanbul Stock Exchange.Expert systems with Applications, 38(5), 5311-5319.
7. Lee, M. C. (2009). Using support vector machine with a hybrid feature selection method to the stock trend prediction. Expert Systems with Applications, 36(8), 10896-10904.
8. Abraham, R., El Samad, M., Bakhach, A. M., El-Chaarani, H., Sardouk, A., El Nemar, S., & Jaber, D. (2022). Forecasting a Stock Trend Using Genetic Algorithm and Random Forest. J. Risk Financial Manag. 2022, 15(5), 188.
9. Hao Li. (2018). Stock Price Prediction Using Attention-based Multi-Input LSTM. Proceedings of Machine Learning Research 95:454-469, 2018.
10. Basak, S., Kar S., Saha S., Khaidem L., Dey S. R.(2018). North American Journal of Economics and Finance (2018), https://doi.org/10.1016/j.najef.2018.06.013.

附錄

程式碼

特徵篩選及分割資料

```
1.  from sklearn.model_selection import train_test_split
2.  from sklearn import preprocessing
3.  from sklearn.feature_selection import SelectKBest
4.  from sklearn.feature_selection import f_classif
5.  def split_data(start_time,end_time,k):
6.      X = data.loc[start_time:end_time,['布林通道帶寬','SMA5','SMA10','SMA20','SMA60','EMA5','EMA10','EMA20','EMA60','K','D','RSI5','RSI10','MACD','WILLAR5','WILLAR10']] #將技術指標當作輸入特徵
7.
8.      y_data = data.loc[start_time : end_time,['股價漲跌幅']]
9.      Y = np.where(y_data.shift(-1)> 0, 0 , 1 ) #歸類
10.
11.     #特徵篩選
12.     k_best = SelectKBest(score_func=f_classif ,k = k)
13.     X_new = k_best.fit_transform(X,np.ravel(Y))
14.
15.     selected_features = k_best.get_support(indices=True)
16.     print(selected_features)
17.
18.     X_train, X_test, y_train, y_test = train_test_split(X, Y, test_size =0.2, random_state = 0 )
19.     return X_train,X_test, y_train, y_test
```

建構 SVC 模型

```
def SupportVector(X_train, X_test, y_train, y_test):
    clf = SVC()
    model = clf.fit(X_train,np.ravel(y_train))

    y_pre = clf.predict(X_test)

    from sklearn.metrics import classification_report
    from sklearn.metrics import classification_report, confusion_matrix

    print(confusion_matrix(y_test, y_pre))
    report = classification_report(y_test,y_pre)
    print(report)

    from sklearn.metrics import accuracy_score
    accuracy = accuracy_score(y_test , y_pre)
    print("accuracy:",accuracy)

    return y_test, y_pre
```

建構隨機森林模型

```
def RandomForest(X_train,X_test,y_train,y_test):
    from sklearn.ensemble import RandomForestClassifier
    clf = RandomForestClassifier(n_estimators=2000, random_state=10, n_jobs=-1 , min_samples_leaf=5)

    model = clf.fit(X_train,np.ravel(y_train))
    y_pre = clf.predict(X_test)

    from sklearn.metrics import classification_report
    report = classification_report(y_test,y_pre)
    print(report)
    print('RandomForestClassifier')

    return y_test, y_pre
```

輸出結果

```
data = data_process(data)

X_train, X_test, y_train, y_test = split_data('2011-08-01','2011-12-31',5)
#選擇期間
y_test, y_pre = SupportVector(X_train,X_test, y_train, y_test )
import optuna
def objective(trial):

    c = trial.suggest_float('C',0,1e+5)
    gamma = trial.suggest_float('gamma',0,1e+5)
    kernel = trial.suggest_categorical('kernel',['rbf','linear','sigmoid'])

    svc_new = SVC(C = c, gamma = gamma , kernel = kernel)
    svc_new.fit(X_train ,np.ravel(y_train))
    from sklearn import metrics
    from sklearn.metrics import accuracy_score
    y_pre_new = svc_new.predict(X_test)
    auc_optuna = accuracy_score(y_test ,y_pre_new )

    return(auc_optuna)

study = optuna.create_study(direction= "maximize")
study.optimize(objective, n_trials=100)

best_trial = study.best_trial
best_accuracy = best_trial.value

best_params = study.best_params
print('準確度：',best_accuracy)
print('最佳參數',best_params)
```

程式輸出畫面

```
In [33]: data = data_process(data)

X_train,X_test,y_train,y_test = split_data(data,'2022-01-01','2022-11-30',['股價漲跌幅','布林通道帶寬','SMA5','SMA

y_test_RF, y_pre_RF = RandomForest(X_train,X_test,y_train,y_test)

auc_roc(y_test_RF, y_pre_RF)

from sklearn.ensemble import RandomForestClassifier
import optuna
def objective(trial):

    n_estimators = trial.suggest_int('n_estimators',100,2000)
    min_samples_leaf = trial.suggest_int('min_samoles_leaf',5,15)

    svc_new = RandomForestClassifier(n_estimators = n_estimators, random_state=10, n_jobs=-1 , min_samples_lea
    svc_new.fit(X_train ,np.ravel(y_train))
    from sklearn import metrics
    from sklearn.metrics import accuracy_score
    y_pre_new = svc_new.predict(X_test)
    auc_optuna = accuracy_score(y_test ,y_pre_new )

    return(auc_optuna)

study = optuna.create_study(direction= "maximize")
study.optimize(objective, n_trials=100)

best_trial = study.best_trial
best_accuracy = best_trial.value

best_params = study.best_params
print('準確度：',best_accuracy)
print('最佳參數',best_params)
```

```
[I 2023-03-08 10:40:12,696] Trial 91 finished with value: 0.5111111111111111 and parameters: {'n_estimators':
1543, 'min_samoles_leaf': 6}. Best is trial 13 with value: 0.5111111111111111.
[I 2023-03-08 10:40:15,257] Trial 92 finished with value: 0.4888888888888889 and parameters: {'n_estimators':
1801, 'min_samoles_leaf': 6}. Best is trial 13 with value: 0.5111111111111111.
[I 2023-03-08 10:40:17,282] Trial 93 finished with value: 0.4222222222222222 and parameters: {'n_estimators':
1465, 'min_samoles_leaf': 13}. Best is trial 13 with value: 0.5111111111111111.
[I 2023-03-08 10:40:19,474] Trial 94 finished with value: 0.5111111111111111 and parameters: {'n_estimators':
1568, 'min_samoles_leaf': 9}. Best is trial 13 with value: 0.5111111111111111.
[I 2023-03-08 10:40:21,248] Trial 95 finished with value: 0.5111111111111111 and parameters: {'n_estimators':
1283, 'min_samoles_leaf': 6}. Best is trial 13 with value: 0.5111111111111111.
[I 2023-03-08 10:40:23,447] Trial 96 finished with value: 0.5111111111111111 and parameters: {'n_estimators':
1504, 'min_samoles_leaf': 7}. Best is trial 13 with value: 0.5111111111111111.
[I 2023-03-08 10:40:25,531] Trial 97 finished with value: 0.4888888888888889 and parameters: {'n_estimators':
1373, 'min_samoles_leaf': 7}. Best is trial 13 with value: 0.5111111111111111.
[I 2023-03-08 10:40:27,841] Trial 98 finished with value: 0.4666666666666667 and parameters: {'n_estimators':
1645, 'min_samoles_leaf': 5}. Best is trial 13 with value: 0.5111111111111111.
[I 2023-03-08 10:40:29,018] Trial 99 finished with value: 0.5111111111111111 and parameters: {'n_estimators':
844, 'min_samoles_leaf': 6}. Best is trial 13 with value: 0.5111111111111111.

準確度： 0.5111111111111111
最佳參數 {'n_estimators': 1179, 'min_samoles_leaf': 6}
```

附圖 1 程式輸出畫面

超參數統計

支持向量機（SVM）超參數統計結果

附表 1 依事件分割—未篩選

事件未篩	C	Gamma	Kernel
2008 金融海嘯	31934.0845	55641.4029	Sigmoid
金融海嘯後反彈	18537.4855	57098.5568	Linear
2011 美債危機	63135.4612	18340.4286	Sigmoid
2015 中國股災	34957.1933	97091.7267	Sigmoid
2018 中美貿易戰	954.4491	16340.5027	Linear
2020 Covid-19 疫情	656.1132	35507.4590	Linear
2020 疫情趨緩反彈	86191.8540	29597.6675	Linear

附表 2 依事件分割—已篩選

事件已篩	C	Gamma	Kernel
2008 金融海嘯	53433.4136	67082.9524	Rbf
金融海嘯後反彈	59403.5198	59819.0180	Linear
2011 美債危機	55916.7373	52705.4934	Sigmoid
2015 中國股災	98647.3421	58368.6106	Linear
2018 中美貿易戰	2655.8397	56596.5916	Linear
2020 Covid-19 疫情	67870.2803	52512.7768	Linear
2020 疫情趨緩反彈	85616.1080	88538.2325	Linear

附表 3 依年份分割—未篩選

年份未篩	C	Gamma	Kernel
2008	3163.8893	28592.8514	Linear
2009	12787.3907	92386.4894	Linear
2010	31728.5639	13658.7618	Rbf
2011	18914.7958	75677.0494	Linear
2012	2549.3562	25082.9574	Linear
2013	57427.6125	92469.1983	Linear
2014	3048.3848	34342.2138	Rbf
2015	41740.9235	94440.4160	Linear
2016	21949.3089	5318.0423	Rbf
2017	20535.1049	11782.2777	Rbf
2018	81069.0187	5099.6787	Linear
2019	86252.1424	43591.1654	Rbf
2020	53399.5875	59074.0050	Sigmoid
2021	2826.2934	13465.5424	Sigmoid
2022	27535.6786	58742.6149	Linear

附表 4 依年份分割—已篩選

年份已篩	C	Gamma	Kernel
2008	14948.6808	63227.4340	Rbf
2009	85243.7144	41001.2054	Linear
2010	87598.4786	41472.7461	Rbf
2011	95082.2278	40747.4369	Linear
2012	61058.9362	49165.5190	Linear
2013	98838.9758	15882.1951	Linear
2014	35606.7020	58823.7233	Linear
2015	51466.5820	80772.5711	Linear
2016	47912.7524	66196.9835	Sigmoid
2017	76608.9260	21801.5653	Linear
2018	63334.9061	17943.8941	Rbf
2019	11312.4826	38846.7221	Linear
2020	4965.8741	2504.1460	Rbf
2021	65316.8576	17386.1209	Rbf
2022	71061.95418	93671.5829	Linear

隨機森林超參數統計結果

附表 5 依事件分割—未篩選

事件未篩	n_estimators	min_samples_leaf
2008 金融海嘯	1046	14
金融海嘯後反彈	387	12
2011 美債危機	423	6
2015 中國股災	1611	9
2018 中美貿易戰	1176	12
2020 Covid-19 疫情	1154	6
2020 疫情趨緩反彈	291	9

附表 6 依事件分割—已篩選

事件已篩	n_estimators	min_samples_leaf
2008 金融海嘯	921	13
金融海嘯後反彈	1543	8
2011 美債危機	168	14
2015 中國股災	1096	5
2018 中美貿易戰	1237	5
2020 Covid-19 疫情	193	9
2020 疫情趨緩反彈	1232	15

附表 7 依年份分割—未篩選

年份未篩	n_estimators	min_samples_leaf
2008	797	9
2009	1214	5
2010	1963	15
2011	566	5
2012	102	5
2013	524	6
2014	167	6
2015	830	7
2016	470	15
2017	1358	14
2018	541	11
2019	178	15
2020	105	5
2021	193	8
2022	1331	13

附表 8 依年份分割—已篩選

年份未篩	n_estimators	min_samples_leaf
2008	892	14
2009	132	8
2010	1085	14
2011	1211	7
2012	1933	5
2013	1705	14
2014	118	7
2015	1955	11
2016	1560	8
2017	1337	8
2018	1617	8
2019	989	13
2020	1113	12
2021	585	12
2022	1179	6