**機器學習用於股票策略分析**

**第十一組**

**賴冠霖 M132040012**

**吳弘曆 M134111058**

**2024/12/17**

目錄

[摘要 - 1 -](#_Toc186498467)

[壹、研究動機 - 1 -](#_Toc186498468)

[貳、資料介紹 - 1 -](#_Toc186498469)

[參、變數介紹 - 2 -](#_Toc186498470)

[肆、資料處理 - 2 -](#_Toc186498471)

[伍、模型選擇與比較 - 5 -](#_Toc186498472)

[陸、解釋模型結果 - 9 -](#_Toc186498473)

[柒、結論 - 10 -](#_Toc186498474)

[捌、未來展望 - 10 -](#_Toc186498475)

[玖、參考資料 - 11 -](#_Toc186498476)

摘要

本研究目標使用月度指標來預測股票每月報酬率的正負方向。我們僅關注只依賴當月的特徵，以避免時間序列的問題。透過資料視覺化、相關性檢查與共線性分析、合併變數、…，篩選出能有效代表當月價格波動和交易活動的變數作為模型輸入，進行多模型比較，最終選擇準確率（accuracy）作為策略評估指標。

壹、研究動機

本研究在於探索能有效預測台積電、鴻海、聯發科股價變動的指標，從而為投資者提供短期決策的依據，並增強投資回報的穩定性。透過機器學習模型，結合股價波動率、交易量、市值等多種特徵，以預測下一期的股價走勢（即預測其價格是否會上漲或下跌）。本研究希望達到風險調整後的超額報酬增長，進一步提升投資者的決策效益。

貳、資料介紹

資料介紹：

本研究聚焦於台灣三家市值最大的上市公司：台積電（2330）、鴻海（2317）、聯發科（2454），利用其月度股票市場指標進行分析。這些數據來源於 TEJPro 台灣經濟新報，反映出個股的市場交易狀況和基本面財務數據，為短期投資決策提供支持。

股票公司簡介：

1. 台積電（2330）：

全球最大的晶圓代工公司，半導體行業的龍頭企業。

1. 鴻海（2317）：

世界領先的電子製造服務供應商，廣泛參與全球電子產品生產鏈。

1. 聯發科（2454）：

全球領先的IC設計公司，專注於消費電子芯片的開發。

數據期間：

數據以月為單位，涵蓋三家公司在不同月份的股價表現及交易情況。

台積電 : 1994/09~2024/12

鴻 海 : 1991/06~2024/12

聯發科 : 2001/07~2024/12

數據來源 ：[TEJPro 官方網站](https://tejpro.tej.com.tw)

參、變數介紹

以下變數用於描述每月的股票市場表現及基本財務數據：

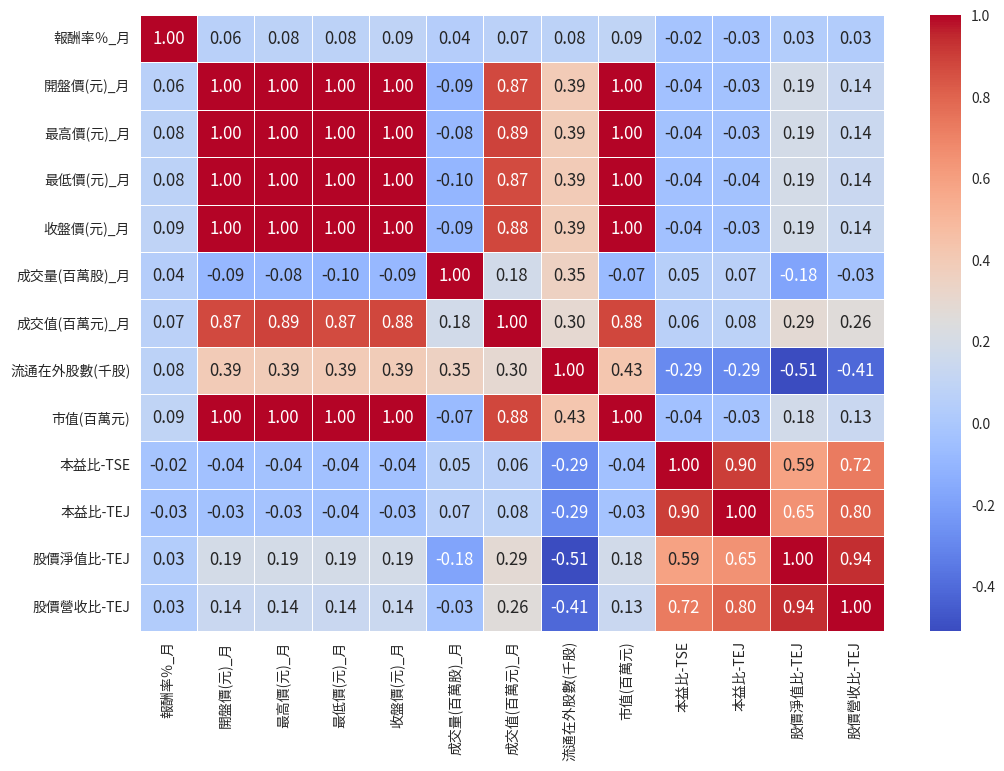
表一 變數介紹

|  |  |
| --- | --- |
| 變數 | 變數介紹 |
| 開盤價(元)\_月 | 每月第一個交易日的開市價格 |
| 最高價(元)\_月 | 該月份內的最高交易價格 |
| 最低價(元)\_月 | 該月份內的最低交易價格 |
| 收盤價(元)\_月 | 每月最後一個交易日的收市價格 |
| 成交量(百萬股)\_月 | 該月份內的股票總成交量（百萬股），反映市場交易活躍度 |
| 成交值(百萬元)\_月 | 該月份內股票總交易金額（百萬元） |
| 流通在外股數(千股) | 市場上該股的流通股數（千股），衡量股票的市場供給 |
| 市值(百萬元) | 市場價值（百萬元），等於流通在外股數乘以當月平均股價 |
| 本益比-TSE | 台灣證券交易所計算的本益比，用於衡量股價相對於每股盈餘的倍數 |
| 本益比-TEJ | TEJ提供的本益比，與TSE方法類似，為另一數據來源的參考 |
| 股價淨值比-TEJ | 股票價格相對於每股帳面價值的比率，反映財務穩定性 |
| 股價營收比-TEJ | 股票價格相對於每股營收的比率，用於比較企業的營收能力 |
| 報酬率％\_月 | 每月股票報酬率，衡量該月投資回報的百分比 |

肆、資料處理

下方圖一的相關係數矩陣為表一中提到的變數，開盤價、最高價、最低價、收盤價、市值互相的相關數為1，說明有完全正相關的問題，且很多變數都有高度線性關係的問題。

下方表二為各變數的VIF，通常VIF>10說明各變數間存在共線性的問題，由表二可以觀察出VIF>>10有共線性的問題，且少許變數亦有輕微共線性的問題。



圖一 相關係數矩陣

表二 各變數VIF

|  |  |  |
| --- | --- | --- |
|  | Feature | VIF |
| 0 | const | 39.301235 |
| 1 | 報酬率％\_月 | 1.231874 |
| 2 | 開盤價(元)\_月 | 393.660600 |
| 3 | 最高價(元)\_月 | 1071.752271 |
| 4 | 最低價(元)\_月 | 769.945683 |
| 5 | 收盤價(元)\_月 | 2072.612323 |
| 6 | 成交量(百萬股)\_月 | 2.742273 |
| 7 | 成交值(百萬元)\_月 | 16.559406 |
| 8 | 流通在外股數(千股) | 9.605910 |
| 9 | 市值(百萬元) | 1528.606917 |
| 10 | 本益比-TSE | 5.550093 |
| 11 | 本益比-TEJ | 8.967182 |
| 12 | 股價淨值比-TEJ | 17.791227 |
| 13 | 股價營收比-TEJ | 22.540411 |

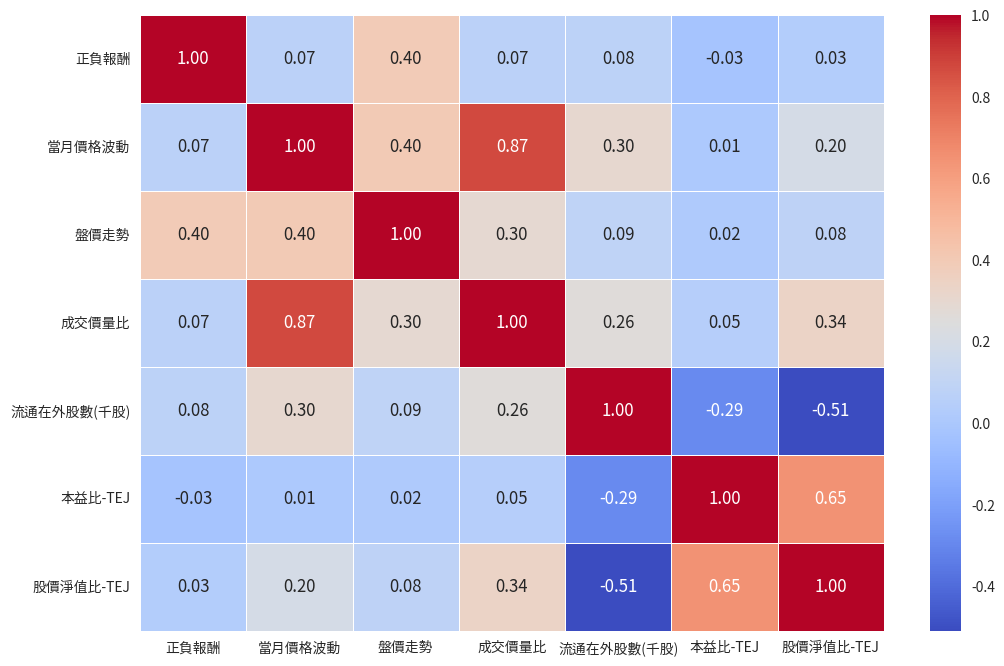
針對圖一、表二說明變數間有高度的線性關係，在下面接續說明變數的處理方式。

表三 變數處理

|  |  |  |
| --- | --- | --- |
| 變數名稱 | 合併方法 | 備註 |
| 當月價格波動 | 最高價 - 最低價 | 元/月 |
| 盤價走勢 | 收盤價 - 開盤價 | 百萬股/月 |
| 成交價量比 | 成交值 / 成交量 | 千股、百萬元 |
| 流通在外股數 | 市值=流通在外股數×股價 |  |
| 正負報酬 | 報酬率>0 : 1  報酬率<＝0 : 0 | 目標變數 |

表三為變數合併的方法，我們將最高價-最低價合併成當月價格波動、收盤價-開盤價合併成盤價走勢、成交值/成交量合併成成交價量比、市值/股價=流通在外股數、將報酬率>0設為1、報酬率<=0設為0，作為此次研究的目標變數。

最後選擇'正負報酬', '當月價格波動', '盤價走勢', '成交價量比', '流通在外股數(千股)',  '本益比-TEJ', '股價淨值比-TEJ’作為分析的變數，其中'正負報酬'為目標變數，接著查看變數處理後的相關係數矩陣及VIF。



圖二 變數處理後的相關係數矩陣

表四 變數處理後的VIF

|  |  |  |
| --- | --- | --- |
|  | Feature | VIF |
| 0 | const | 26.780355 |
| 1 | 正負報酬 | 1.227404 |
| 2 | 當月價格波動 | 4.946139 |
| 3 | 盤價走勢 | 1.456559 |
| 4 | 成交價量比 | 5.561242 |
| 5 | 流通在外股數(千股) | 2.022286 |
| 6 | 本益比-TEJ | 1.972179 |
| 7 | 股價淨值比-TEJ | 3.606143 |

經過變數合併後，由圖二發現各變數間線性相關性降低，當中與目標變數最有相關性的是盤價走勢，且在表四中VIF值皆<10，說明變數合併後解決了共線性的問題。最終我們挑選這些變數進行建模。

伍、模型選擇與比較

模型的策略實施：

定期定存(RSP)策略：

* 每月固定投入 1000 元。

制定投資策略：

* 每月固定投入 1000 元。
* 根據 test\_label\_pred 的最後一個值（0 或 1）決定買入訊號。
* 若當期的預測值為 1，則將累積的現金（包含當期定投的 1000 元）全部用於購買股票，買入價格為當期的「開盤價」。
* 若當期預測值為 0，則不進行購買操作，當期的 1000 元 將累積至後續期數，直至某一期預測值為 1 再進行投資。

＊test\_label\_pred：根據各個機器學習預測下個月漲或跌的訊號

使用模型 :

本研究應用機器學習的方法預測正負報酬，使用5種分類器，分別是"Random\_Forest""Gradient\_Boosting""Support\_Vector\_Machine""K-Nearest\_Neighbors""Logistic\_Regression"，並使用滯後期數 (lag=1~5)進行建模，將數據60/20/20分割成訓練集、驗證集、測試集，並使用上述5模型建立5個滯後期數，共25個模型進行模型比較。

準備了3股票(台積電、鴻海、聯發科) 分別建立25個模型列在下面 :

台積電2330 :

Random Forest

lag\_k = 1, valid準確率: 0.4583, test準確率: 0.3944

lag\_k = 2, valid準確率: 0.5000, test準確率: 0.4143

lag\_k = 3, valid準確率: 0.5833, test準確率: 0.4348

lag\_k = 4, valid準確率: 0.5000, test準確率: 0.3971

lag\_k = 5, valid準確率: 0.5694, test準確率: 0.5522

Gradient Boosting

lag\_k = 1, valid準確率: 0.4167, test準確率: 0.3803

lag\_k = 2, valid準確率: 0.4028, test準確率: 0.4000

lag\_k = 3, valid準確率: 0.4028, test準確率: 0.3768

lag\_k = 4, valid準確率: 0.4167, test準確率: 0.3824

lag\_k = 5, valid準確率: 0.4722, test準確率: 0.4179

Support Vector Machine

lag\_k = 1, valid準確率: 0.6111, test準確率: 0.6338

lag\_k = 2, valid準確率: 0.6111, test準確率: 0.6286

lag\_k = 3, valid準確率: 0.6250, test準確率: 0.6232

lag\_k = 4, valid準確率: 0.6250, test準確率: 0.6176

lag\_k = 5, valid準確率: 0.6111, test準確率: 0.6269

K-Nearest Neighbors

lag\_k = 1, valid準確率: 0.6111, test準確率: 0.6338

lag\_k = 2, valid準確率: 0.6111, test準確率: 0.6286

lag\_k = 3, valid準確率: 0.6250, test準確率: 0.6232

lag\_k = 4, valid準確率: 0.6250, test準確率: 0.6176

lag\_k = 5, valid準確率: 0.6111, test準確率: 0.6269

Logistic Regression

lag\_k = 1, valid準確率: 0.6111, test準確率: 0.6338

lag\_k = 2, valid準確率: 0.6111, test準確率: 0.6286

lag\_k = 3, valid準確率: 0.6250, test準確率: 0.6232

lag\_k = 4, valid準確率: 0.6250, test準確率: 0.6176

lag\_k = 5, valid準確率: 0.6111, test準確率: 0.6269

結果摘要:

Random Forest 在 lag\_k = 3 時達到最大驗證準確率: 0.5833

Gradient Boosting 在 lag\_k = 5 時達到最大驗證準確率: 0.4722

Support Vector Machine 在 lag\_k = 3 時達到最大驗證準確率: 0.6250

K-Nearest Neighbors 在 lag\_k = 3 時達到最大驗證準確率: 0.6250

Logistic Regression 在 lag\_k = 3 時達到最大驗證準確率: 0.6250

鴻海2317 :

Random Forest

lag\_k = 1, valid準確率: 0.4125, test準確率: 0.5443

lag\_k = 2, valid準確率: 0.5250, test準確率: 0.4615

lag\_k = 3, valid準確率: 0.5875, test準確率: 0.5584

lag\_k = 4, valid準確率: 0.5500, test準確率: 0.5395

lag\_k = 5, valid準確率: 0.6250, test準確率: 0.5333

Gradient Boosting

lag\_k = 1, valid準確率: 0.5000, test準確率: 0.5063

lag\_k = 2, valid準確率: 0.6250, test準確率: 0.5128

lag\_k = 3, valid準確率: 0.5500, test準確率: 0.5195

lag\_k = 4, valid準確率: 0.5125, test準確率: 0.5000

lag\_k = 5, valid準確率: 0.5375, test準確率: 0.5200

Support Vector Machine

lag\_k = 1, valid準確率: 0.3875, test準確率: 0.4557

lag\_k = 2, valid準確率: 0.4375, test準確率: 0.4231

lag\_k = 3, valid準確率: 0.3875, test準確率: 0.4545

lag\_k = 4, valid準確率: 0.4375, test準確率: 0.4211

lag\_k = 5, valid準確率: 0.4875, test準確率: 0.4000

K-Nearest Neighbors

lag\_k = 1, valid準確率: 0.3875, test準確率: 0.4557

lag\_k = 2, valid準確率: 0.4000, test準確率: 0.4359

lag\_k = 3, valid準確率: 0.3875, test準確率: 0.4545

lag\_k = 4, valid準確率: 0.4000, test準確率: 0.4474

lag\_k = 5, valid準確率: 0.5875, test準確率: 0.5600

Logistic Regression

lag\_k = 1, valid準確率: 0.4000, test準確率: 0.4557

lag\_k = 2, valid準確率: 0.3875, test準確率: 0.4359

lag\_k = 3, valid準確率: 0.4500, test準確率: 0.4416

lag\_k = 4, valid準確率: 0.4250, test準確率: 0.4342

lag\_k = 5, valid準確率: 0.4125, test準確率: 0.4267

結果摘要:

Random Forest 在 lag\_k = 5 時達到最大驗證準確率: 0.6250

Gradient Boosting 在 lag\_k = 2 時達到最大驗證準確率: 0.6250

Support Vector Machine 在 lag\_k = 5 時達到最大驗證準確率: 0.4875

K-Nearest Neighbors 在 lag\_k = 5 時達到最大驗證準確率: 0.5875

Logistic Regression 在 lag\_k = 3 時達到最大驗證準確率: 0.4500

聯發科2454 :

Random Forest

lag\_k = 1, valid準確率: 0.5536, test準確率: 0.4727

lag\_k = 2, valid準確率: 0.4821, test準確率: 0.4630

lag\_k = 3, valid準確率: 0.6071, test準確率: 0.3585

lag\_k = 4, valid準確率: 0.5357, test準確率: 0.3269

lag\_k = 5, valid準確率: 0.5357, test準確率: 0.3725

Gradient Boosting

lag\_k = 1, valid準確率: 0.4821, test準確率: 0.4909

lag\_k = 2, valid準確率: 0.5000, test準確率: 0.4074

lag\_k = 3, valid準確率: 0.5536, test準確率: 0.4151

lag\_k = 4, valid準確率: 0.4464, test準確率: 0.4038

lag\_k = 5, valid準確率: 0.3214, test準確率: 0.3529

Support Vector Machine

lag\_k = 1, valid準確率: 0.6071, test準確率: 0.6727

lag\_k = 2, valid準確率: 0.6071, test準確率: 0.6667

lag\_k = 3, valid準確率: 0.6071, test準確率: 0.6604

lag\_k = 4, valid準確率: 0.6071, test準確率: 0.6731

lag\_k = 5, valid準確率: 0.6250, test準確率: 0.6667

K-Nearest Neighbors

lag\_k = 1, valid準確率: 0.3929, test準確率: 0.3273

lag\_k = 2, valid準確率: 0.3929, test準確率: 0.3333

lag\_k = 3, valid準確率: 0.3929, test準確率: 0.3396

lag\_k = 4, valid準確率: 0.3929, test準確率: 0.3269

lag\_k = 5, valid準確率: 0.3750, test準確率: 0.3333

Logistic Regression

lag\_k = 1, valid準確率: 0.6071, test準確率: 0.6727

lag\_k = 2, valid準確率: 0.6071, test準確率: 0.6667

lag\_k = 3, valid準確率: 0.6071, test準確率: 0.6604

lag\_k = 4, valid準確率: 0.6071, test準確率: 0.6731

lag\_k = 5, valid準確率: 0.6250, test準確率: 0.6667

結果摘要:

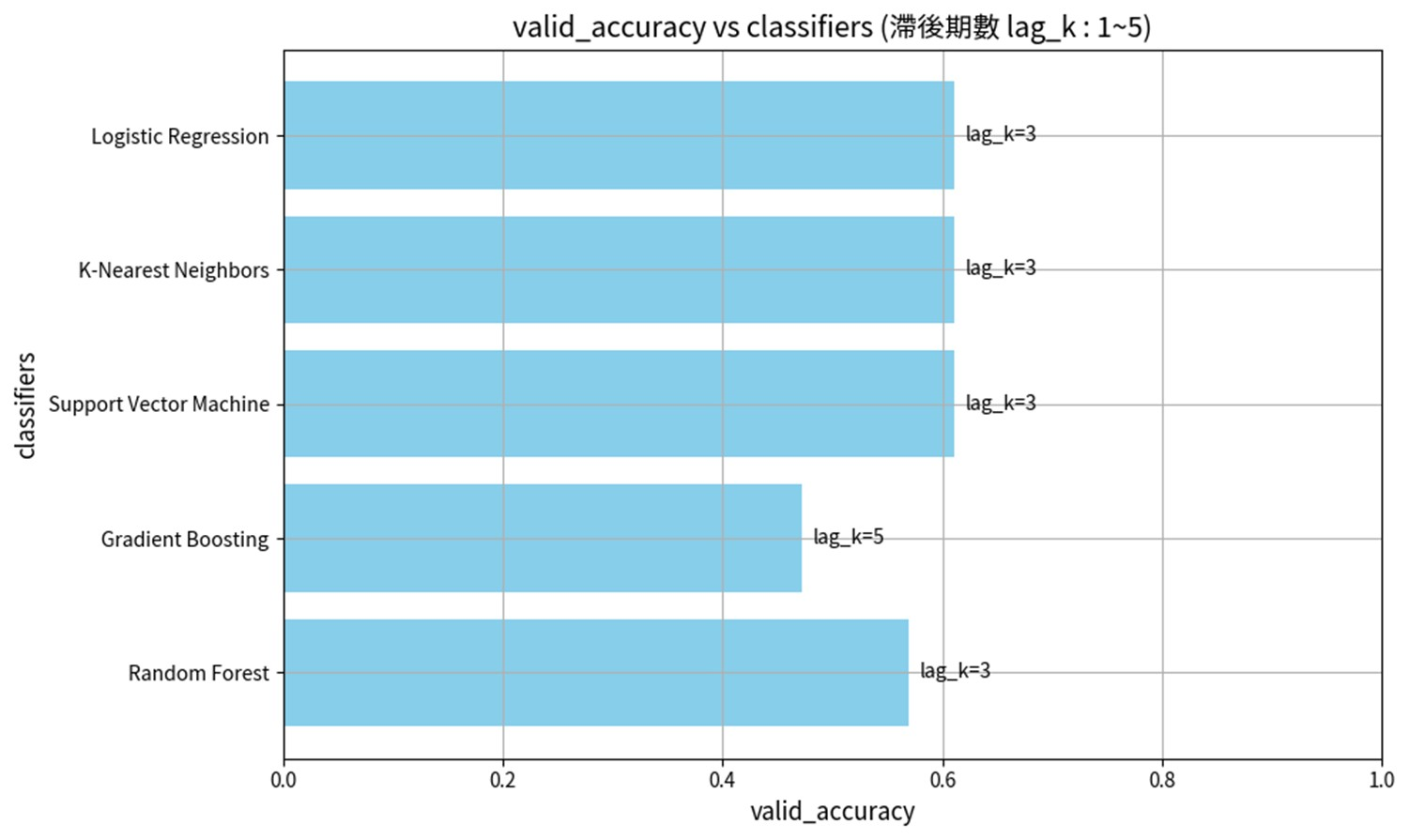
Random Forest 在 lag\_k = 3 時達到最大驗證準確率: 0.6071

Gradient Boosting 在 lag\_k = 3 時達到最大驗證準確率: 0.5536

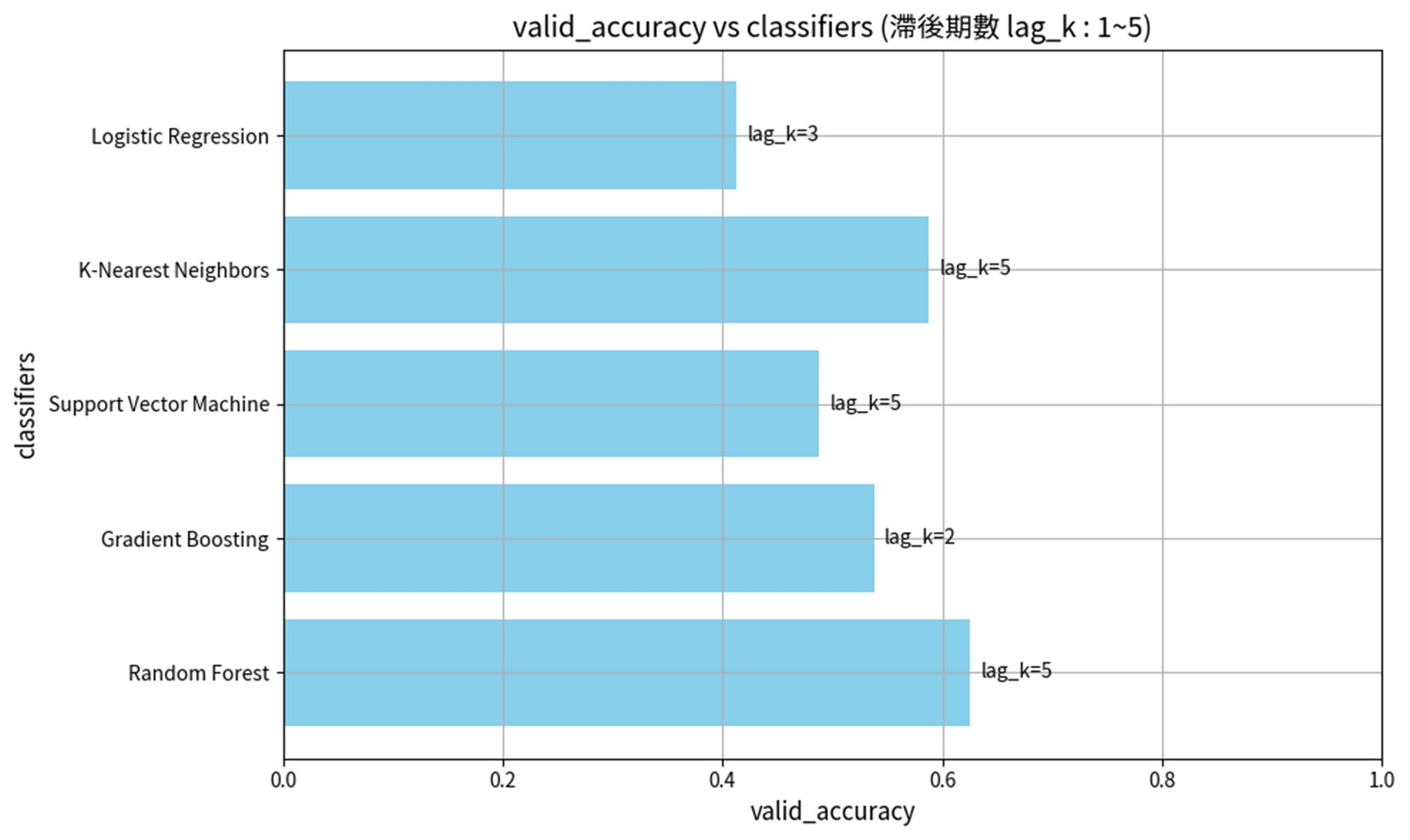
Support Vector Machine 在 lag\_k = 5 時達到最大驗證準確率: 0.6250

K-Nearest Neighbors 在 lag\_k = 1 時達到最大驗證準確率: 0.3929

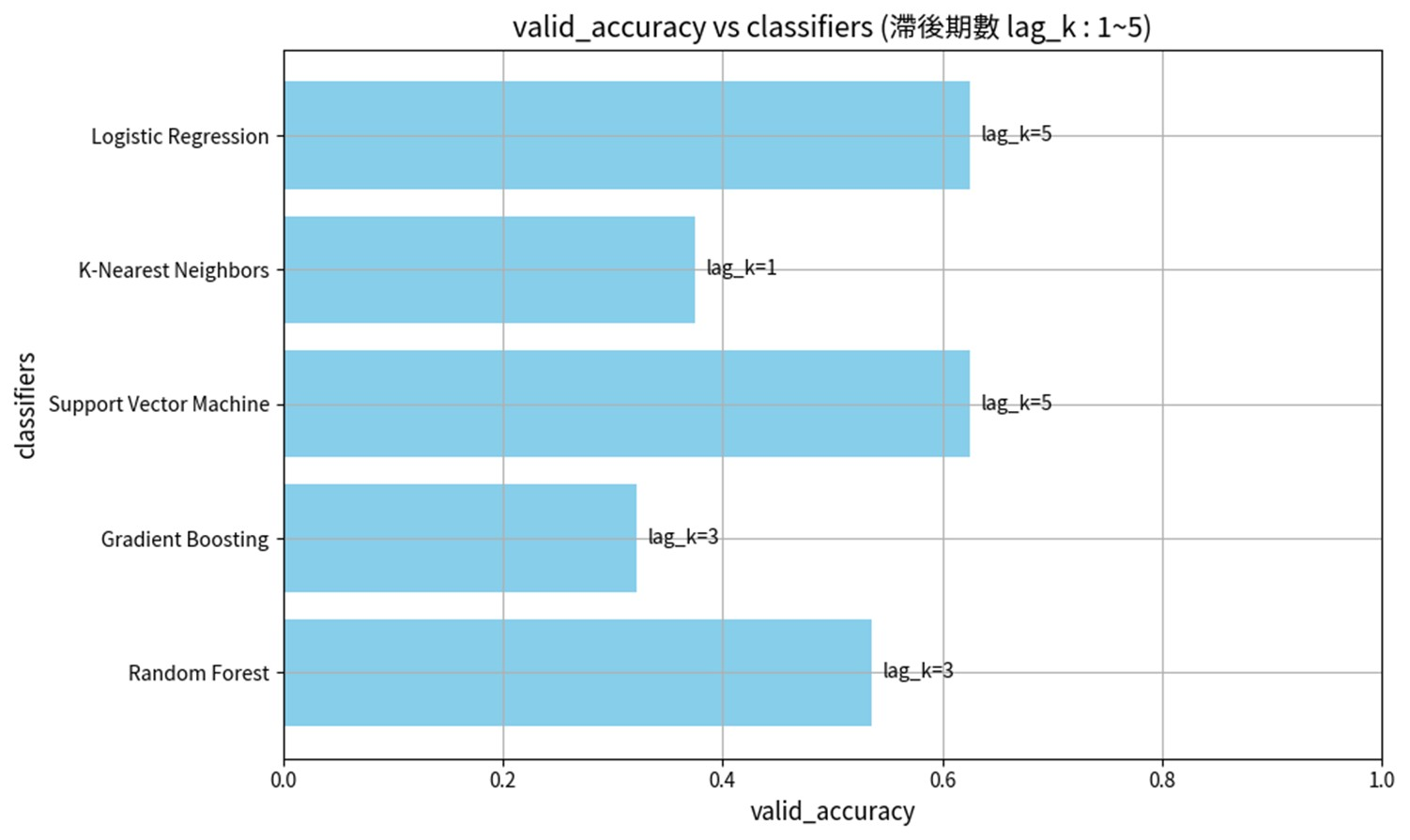
Logistic Regression 在 lag\_k = 5 時達到最大驗證準確率: 0.6250



圖三 模型比較\_台積電



圖四 模型比較\_鴻海



圖五 模型比較\_聯發科

上方列出3大股票中5個模型下最優的lag期數之驗證準確率，在台積電中，LR、KNN、SVM在lag期數3時有驗證準確率最高為0.625;在鴻海中，RF在期數5時有驗證準確率最高為0.625;在聯發科中，LR、SVM在期數5時有驗證準確率最高為0.625。

陸、解釋模型結果

將先前在5個模型中最好的模型進行迴測，並與RSP進行比較，呈現在下方。

圖六 投報率表現一張含有 文字, 螢幕擷取畫面, 數字, 字型 的圖片

AI 產生的內容可能不正確。

中觀察報酬率都沒有超過RSP的報酬率，且在台積電的KNN、LR、SVM模型接說明RSP是最好的策略，其他的股票皆說明RSP是最好的測略。

柒、結論

定期定存(RSP)：在台灣前三大檔股票中總投報率最高的投資策略。

RF：各個股票當中接稍遜色於定額定存

LR：雖然在2330跟2454中表現跟定額定存一樣，但在2317中的表現卻拖垮了整體表現

SVM：跟羅吉斯回歸是一樣的問題，只是在2317中的表現更為誇張

GB、KNN：表現得差強人意了

捌、未來展望

數據優化

1. 模型優化

強化隨機森林和梯度提升機的特徵工程與參數調整，進一步優化預測準確率、對不同股票特性進行個別建模，避免單一模型拖累整體表現。

1. 引入時間序列模型

嘗試引入 ARIMA、LSTM 等時間序列模型，以更有效捕捉股價隨時間變動的趨勢。

1. 特徵選取

可進一步引入其他因子（特徵），如市場情緒指數、宏觀經濟數據等，以提升模型的預測能力。

策略優化

1. 個股選取

選取波動度大的非成長股，使模型回測的結論不單一

1. 動態資金管理

探討動態資金分配策略，根據各股票預測的報酬率，靈活調整投資比例，以提升整體報率。

1. 風險管理機制

加入止損與停利機制，避免市場極端波動對投資組合造成過大損失。

玖、參考資料

1. <https://tejpro.tej.com.tw/tejpro/NTU/?lang=zh-TW>
2. <https://rich01.com/what-is-quantitative-trading/>
3. <https://www.oanda.com/bvi-ft/lab-education/>
4. [https://www.tejwin.com/insight/【資料科學】xgboost-演算法預測報酬上/](https://www.tejwin.com/insight/%E3%80%90%E8%B3%87%E6%96%99%E7%A7%91%E5%AD%B8%E3%80%91xgboost-%E6%BC%94%E7%AE%97%E6%B3%95%E9%A0%90%E6%B8%AC%E5%A0%B1%E9%85%AC%E4%B8%8A/)
5. [https://www.tejwin.com/insight/xgboost-演算法預測報酬下/](https://www.tejwin.com/insight/xgboost-%E6%BC%94%E7%AE%97%E6%B3%95%E9%A0%90%E6%B8%AC%E5%A0%B1%E9%85%AC%E4%B8%8B/)